PAPER: CLASSICAL STATISTICAL MECHANICS, EQUILIBRIUM AND NON-EQUILIBRIUM

Computing return times or return periods with rare event algorithms

Journal of Statistical Mechanics: Theory and Experiment
An IOP and SISSA journal

PAPER: Classical statistical mechanics, equilibrium and non-equilibrium

Computing return times or return periods with rare event algorithms

Thibault Lestang[1,2], Francesco Ragone[3], Charles-Edouard Bréhier[4], Corentin Herbert[1] and Freddy Bouchet[1]

[1] Univ Lyon, ENS de Lyon, Univ Claude Bernard, CNRS, Laboratoire de Physique, F-69342 Lyon, France
[2] Univ Lyon, Ecole Centrale de Lyon, Univ Claude Bernard, CNRS, Laboratoire de Mécanique des Fluides et d'Acoustique, F-69134 Ecully cedex, France
[3] Department of Earth and Environmental Sciences, University of Milano-Bicocca, Milan, Italy
[4] Univ Lyon, Université Claude Bernard Lyon 1, CNRS UMR 5208, Institut Camille Jordan, 43 blvd. du 11 novembre 1918, F-69622 Villeurbanne cedex, France

E-mail: thibault.lestang@ens-lyon.fr, francesco.ragone@unimib.it, brehier@math.univ-lyon1.fr, corentin.herbert@ens-lyon.fr and freddy.bouchet@ens-lyon.fr

Received 27 November 2017
Accepted for publication 20 March 2018
Published 25 April 2018

Online at stacks.iop.org/JSTAT/2018/043213
https://doi.org/10.1088/1742-5468/aab856

Abstract. The average time between two occurrences of the same event, referred to as its *return time* (or return period), is a useful statistical concept for practical applications. For instance insurances or public agencies may be interested by the return time of a 10 m flood of the Seine river in Paris. However, due to their scarcity, reliably estimating return times for rare events is very difficult using either observational data or direct numerical simulations. For rare events, an estimator for return times can be built from the extrema of the observable on trajectory blocks. Here, we show that this estimator can be improved to remain accurate for return times of the order of the block size. More importantly, we show that this approach can be generalised to estimate return times from numerical algorithms specifically designed to sample rare events. So far those algorithms often compute probabilities, rather than return times. The approach we propose provides a computationally extremely efficient way to estimate numerically the return times of rare events

1742-5468/18/043213+33$33.00

for a dynamical system, gaining several orders of magnitude of computational costs. We illustrate the method on two kinds of observables, instantaneous and time-averaged, using two different rare event algorithms, for a simple stochastic process, the Ornstein–Uhlenbeck process. As an example of realistic applications to complex systems, we finally discuss extreme values of the drag on an object in a turbulent flow.

Keywords: classical Monte Carlo simulations, large deviations in non-equilibrium systems, sampling algorithms, mixing, extreme value

Contents

J. Stat. Mech. (2018) 043213

1. Introduction

In many physical systems, the mean state and the typical fluctuations about this state, usually studied in statistical physics, are not the only quantities of interest. Indeed, fluctuations far away from the mean state, although they are usually very rare, can play a crucial part in the macroscopic behaviour of the system. For instance, they can drive the system to a new metastable state, possibly with radically different properties [1]. Such transitions arise in a wide variety of situations, such as Josephson junctions [2], quantum oscillators [3], turbulent flows [4], magneto-hydrodynamics dynamos [5], diffusion-controlled chemical reactions [6], protein folding [7], climate dynamics [8]. Even if the system returns to its original state after undergoing the large fluctuation, the impact of this event may be so large that it is worth being studied on its own. One may think for instance about heat waves [9] and tropical cyclones, rogue waves in the ocean [10], strong dissipative events in turbulent flows [11], shocks in financial markets [12]. Here, we are concerned with the study of such atypical fluctuations starting from the equations (deterministic or stochastic) which govern the dynamics of the system. This approach is different from and complementary to the purely statistical methods which try to extract the best possible information about the distribution of rare events from an existing timeseries, such as, for instance, extreme value statistics [13–15].

The theoretical framework which has been developed over the last decades in statistical physics to tackle this problem is that of *large deviation theory* [16–20]. Numerical methods have also been developed to efficiently sample rare events, which are not amenable to classical Monte-Carlo methods [21–23]; see [24, 25] for general references on rare event simulation. Those algorithms can be roughly divided into two main classes: those which work in state space, and evolve a population of *clones* of the system according to selection rules biased to favour the appearance of the desired rare event [26–30], and those which try to sample directly in path space the histories of the system which exhibit the phenomenon of interest [31–36]. They can be used either for stochastic processes or deterministic chaotic dynamical systems [37]. Most of those algorithms ultimately compute either one-time statistics (typically, the stationary probability distribution of the system, for which they sample efficiently the tails, or alternatively, large deviation rate functions or scale cumulant generating functions), or reactive trajectories corresponding to the transition between two metastable states.

From a modelling perspective, it is natural to assume that successive occurrences of a rare event are independent from one another [12, 38, 39]. Then, the average number of events occurring in a time interval is proportional to the length of that interval. This is the definition of a Poisson process. In this case, all the statistics are encoded in a single parameter, the rate of the Poisson process. In the following, we will assume that we are dealing with the simple case of a well identified process that can be described by a single return time or rate. This is often a sufficient framework; indeed the long time behaviour of many systems can be described phenomenologically, or exactly in some limits, as a Markov process described by a set of transition rates describing independent processes (see for instance [16] for systems driven by a weak noise). We note however that many other physical systems are not amenable to such a simple effective Markov process, for instance structural glasses or amorphous media.

J. Stat. Mech. (2018) 043213

For many practical applications, the most useful information about a rare event is the *return time*: it is the typical time between two occurrences of the same event. This is how hydrologists measure the amplitude of floods for instance [40]. As a matter of fact, one of the motivations of Gumbel, a founding father of extreme value theory, was exactly this problem [41]. Other natural hazards, such as earthquakes [42] and landslides [43], are also ranked according to their return time. Similarly, climatologists seek to determine how the frequency of given heat waves [44, 45] or cold spells [46] evolves in a changing climate [47]. Public policies rely heavily on a correct estimate of return times: for instance, in the United States, floodplains were defined in the national flood insurance program in 1968 as areas vulnerable to events with a 100 year return time. Such definitions are then used to determine insurance policies for home owners. In the industry as well, return times are the metric used by engineers to design systems withstanding a given class of events. Another property describing rare events of a time series is the average time between successive records [48]; here, because of its importance in practical applications, we focus on the return time, i.e. the average time between events of a given amplitude. Just like the extreme values of any observable, the return time of a rare event is very difficult to estimate directly from observational or numerical data, because extremely long time series are necessary.

The return time may be estimated heuristically by interpreting it as a *first-passage time*. The *first-passage time* (sometimes also called *first exit time*) is defined as the time it takes a stochastic process to reach the boundary of a given domain for the first time; the properties of this random variable have been studied extensively in statistical physics [49, 50]. Then, the return time (or return period) $r(a)$ for an event of amplitude a (return level) may be at first sight related to the inverse of the stationary probability p_s: $r(a) = \tau_c(a)/p_s(a)$, where the correlation time $\tau_c(a)$ usually depends on a but remains bounded when $p_s(a)$ goes to zero. This is true for instance for a system perturbed by a small-noise ϵ at the level of large deviations: $r(a) \underset{\epsilon\to 0}{\asymp} \mathrm{e}^{U(a)/\epsilon}$, where the quasi-potential U is defined by $p_s(a) \underset{\epsilon\to 0}{\asymp} \mathrm{e}^{-U(a)/\epsilon}$ [16]. However the return time is only roughly proportional to the inverse of the stationary probability [51]. In order to compute $\tau_c(a)$ one has to go beyond large deviation theory. For instance for gradient dynamics and for first exit time problems, exact formulas exists [52–54], valid at leading order in ϵ (we stress that different formula are obtained depending on the hypothesis made on the domain that the particle exits). Generalisations to irreversible non gradient dynamics also exist (see [55] and references therein). From these computations, it appears clearly that $\tau_c(a)$ is not simply related to $p_s(a)$ and that the return time $r(a)$ is a trajectory property, not amenable to a one point statistics like $p_s(a)$.

There is thus a need to develop rare event algorithms specifically designed for computing return times, valid also when large deviation estimates are not relevant. This is the aim of this paper. The approach developed in this work relies on the combination of two observations. First, if one assumes that the occurrence of rare events are described by a Poisson process, then return times can be related to the probability of observing extrema over pieces of trajectories, which are of duration much larger than the correlation time of the system, but typically much smaller than the computed return times. Second, several classes of rare event algorithms can be easily generalised to compute the probability of extrema over pieces of trajectories, rather than to compute single point statistics. We show that combining these two remarks enables us to build a

J. Stat. Mech. (2018) 043213

powerful tool to compute return times in an elementary way with simple and robust algorithms. As a side remark, we also discuss a new way to construct return time plots from a timeseries, which provides an important improvement for return times moderately larger than the sampling time, even when we are not using a rare event algorithm.

We illustrate the method by computing return times, first for an instantaneous observable (one-point statistics) using the adaptive multilevel splitting (AMS) algorithm [28, 56], and second for a time-averaged observable, using both the AMS algorithm and the Del Moral–Garnier algorithm [27] (or equivalently the Giardina–Kurchan algorithm [57] in a non-stationary context). The computation of return times with the AMS algorithm leads us to define a generalisation called the trajectory-adaptive multilevel splitting (TAMS) algorithm. This generalisation has several practical advantages: it computes directly return times $r(a)$ for a full range of return level a rather than a single one, and it avoids the tricky estimation of time scale on an auxiliary ensemble, and the sampling from this auxiliary ensemble. As a test, we first carry out these computations for a simple stochastic process, the Ornstein–Uhlenbeck (OU) process, for which analytical results are available and the accuracy and efficiency of the algorithm can be tested thoroughly. Then, to demonstrate the usefulness of the method in realistic applications, we briefly showcase a problem involving a complex dynamical system: extreme values of the drag on an object immersed in a turbulent flow.

The structure of this paper is as follows: in section 2, we introduce the method to compute return times from a timeseries and from rare event algorithms. We define the TAMS algorithm in section 3. We apply the method to compute return times for the instantaneous and time-averaged observables for an Ornstein–Uhlenbeck process, respectively, in section 3 (using the TAMS algorithm) and section 4 (using both the TAMS and the Giardina–Kurchan–Tailleur–Lecomte (GKTL) algorithms). Finally, we introduce the application to complex dynamical systems in section 5, before presenting our conclusions in section 6. We discuss in the conclusions the range of applicability of these algorithms.

2. Return times: definition and sampling methods

2.1. Computing return times from a timeseries

2.1.1. Definition of return times. We consider a statistically time homogeneous ergodic process (a stationary timeseries) $\{A(t)\}_{t\geqslant t_0}$. Typically, $A:\mathbb{R}^d\to\mathbb{R}$ is an observable on a system of interest, considered here as a $\mathbb{R}^d$-valued stochastic process $\left(X_t\right)_{t\geqslant t_0}$, and we should denote $A(t)=A(X_t)$. We are interested in the statistical distribution of events where the observable reaches a prescribed threshold a. The occurrence of such events is illustrated for a sample OU process, defined by $\mathrm{d}X_t=-\alpha X_t\mathrm{d}t+\sqrt{2\epsilon}\mathrm{d}W_t$, on figure 1(a). We define the return time for a given threshold a as the average time one has to wait before observing the next event with $A(t)>a$. More precisely, we define the waiting time

$$\tau(a,t)=\min\left\{\tau\geqslant t\,|A\left(\tau\right)>a\right\}-t. \tag{1}$$

As an illustration, the waiting time $\tau(a,t)$ is shown for our sample Ornstein–Uhlenbeck process on figure 1(b). Then the return time $r(a)$ for the threshold a is defined as

J. Stat. Mech. (2018) 043213

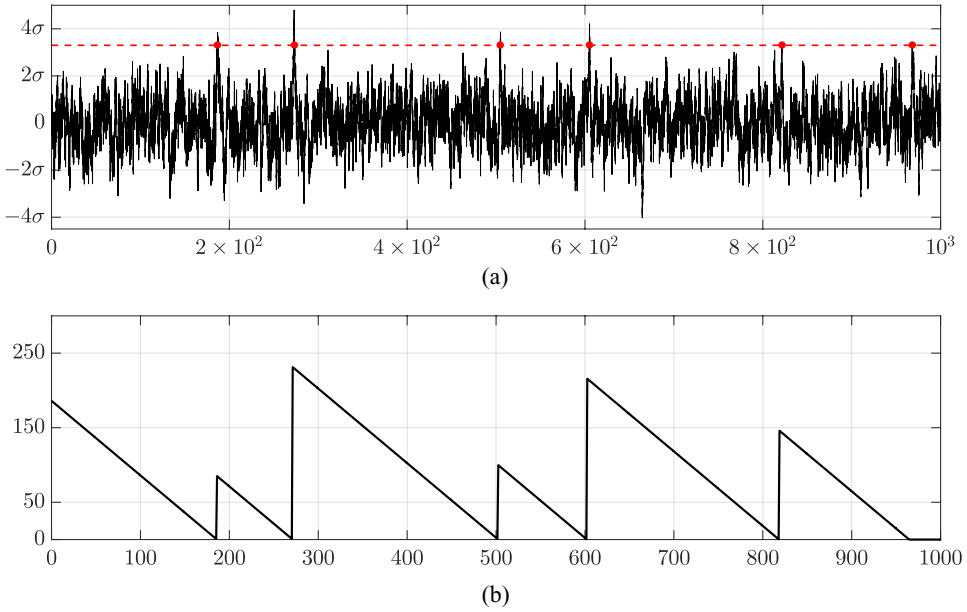

Figure 1. An example of a random process (a) and the waiting time (b) associated to events reaching a given threshold. (a) Sample timeseries (black curve), generated from an Ornstein–Uhlenbeck process (22) ($\alpha = 1$, $\epsilon = 1/2$; $\sigma = 1/\sqrt{2}$ is the standard deviation). We are interested in fluctuations which reach a prescribed threshold a (dashed red line). These events are identified by the red dots. (b) Time evolution of the waiting time $\tau(a, t)$ (see (1)) associated to the above timeseries: it is a succession of affine parts with slope -1. Note that in principle, there should be small time intervals such that $\tau(a, t) = 0$, corresponding to the duration of the event with $A(t) > a$, separating the triangles. Here, the duration of the events is too small for such intervals to be visible.

$$r(a) = \mathbb{E}\left[\tau(a, t)\right], \tag{2}$$

where $\mathbb{E}$ is the average with respect to realisations of the process X with initial condition $X_{t_0} = x_0$ (hence the notation $\mathbb{E} = \mathbb{E}_{x_0,t_0}$ in that case), or is a time average for an ergodic process. From now on, we shall omit the indices when there is no ambiguity. The return time $r(a)$ is independent of time because the process is homogeneous.

The problem we consider in this section is that of estimating $r(a)$ from a sample timeseries of duration $T_d : \{A(t)\}_{0 \leqslant t \leqslant T_d}$. The definition leads to an obvious estimator for $r(a)$, the *direct estimator* $\hat{r}_D$ defined by

$$\hat{r}_D(a) = \frac{1}{T_d} \int_0^{T_d} \tau(a, t)\, \mathrm{d}t = \frac{1}{T_d} \sum_{n=1}^{N_d} \frac{\tau_n^2}{2}, \tag{3}$$

where τ_n is the duration of the successive intervals over which $A(t) \leqslant a$, and N_d is the number of such intervals. The last identity in (3) is illustrated graphically in figure 1(b): the integral of $\tau(a, t)$ is given by computing the total area beneath the triangles.

In the limit of rare events, the return time will also be the average time between two successive independent events. However the definition (2) for the return time has the big advantage of not having to deal with the definition of independent events, which is

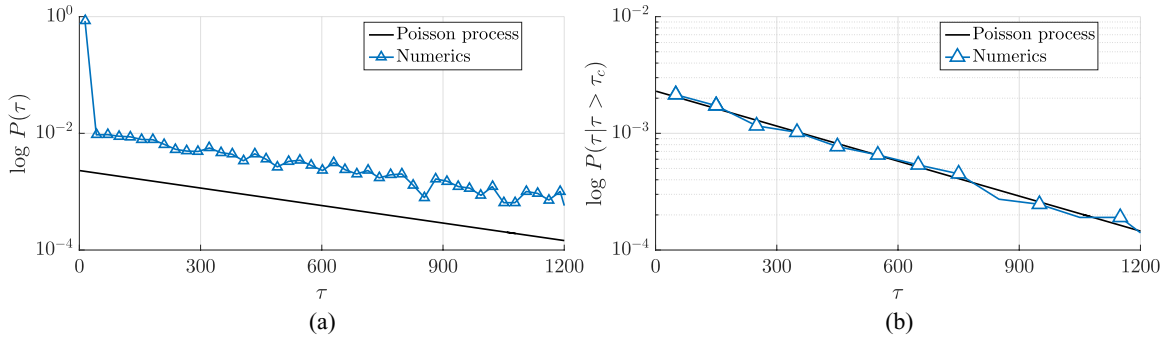

Figure 2. PDF of waiting times between two consecutive fluctuations of amplitude $a = 2.5$, estimated from a timeseries of length $T_d = 10^6$ of the Ornstein–Uhlenbeck process (22) with $\alpha = 1$ and $\epsilon = 1/2$ (blue triangles), and assuming the events follow a Poisson process with rate $1/r(a)$, $P(\tau) = \mathrm{e}^{-\tau/r(a)}/r(a)$ (black solid line), where $r(a)$ is computed from the timeseries. The correlation time of the Ornstein–Uhlenbeck process is $\tau_c = 1/\alpha = 1$. (a) Taking all intervals into account, including those corresponding to oscillations around the threshold. (b) Discarding small intervals ($\tau < \tau_c$) linked to oscillations around the threshold.

cumbersome when time correlations are not negligible. We explain this further in the following section.

2.1.2. Return times and the distribution of successive events. Estimating return times using (3) implies computing the time intervals τ_n between successive events with $A(t) > a$. When a is large enough, most of the times $A(t) < a$ and very rarely $A(t) > a$. Then we can distinguish two kinds of contributions to the time intervals τ_n. On the one hand, we have correlated events corresponding to fluctuations around the threshold value a, on a timescale of the order of the correlation time. From our point of view, these correspond to the same event, with a finite duration. On the other hand, there are successive events such as those depicted in figure 1(a), which can be considered as statistically independent events. Therefore, we expect those events to form a Poisson point process, and the corresponding time intervals τ_n should be distributed according to the distribution of time intervals of a Poisson process: $P(\tau) = \lambda \exp(-\lambda\tau)$ [12, 38, 39].

Figure 2(a) shows the probability density function (PDF) of the time interval between two occurrences of an event $A(t) > a$, drawn from a sample timeseries generated with an Ornstein–Uhlenbeck process. One can see that most of the contributions are indeed small intervals of the order of the correlation time. Discarding all the time intervals below the correlation time, one obtains the PDF displayed in figure 2(b), which coincides with the exponential distribution corresponding to a Poisson point process.

When a is large, $r(a) \gg \tau_c$ where τ_c is the correlation time of the process. Then the contribution of intervals τ_n of duration comparable to τ_c in the formula (3) becomes asymptotically negligible compared to the contribution of the time intervals $\tau_n \gg \tau_c$. Graphically, this may be seen as the fact that the sum in (3) is dominated by the contribution of very big triangles, while for small a all the triangles have roughly the same area. Then, the return time $r(a)$ coincides with the average time between two statistically independent events exceeding the value a. In other words, rare fluctuations

can be considered as independent from one another, their duration can be neglected compared to their return time, and the distribution of such events is well approximated by a Poisson process of rate $\lambda = 1/r(a)$.

Neglecting the duration of the extreme events yields $\sum_{n=1}^{N_d} \tau_n \approx T_d$ and then one can check that

$$\frac{1}{T_d}\sum_{n=1}^{N_d}\frac{\tau_n^2}{2} \approx \frac{N_d}{\sum_{n=1}^{N_d}\tau_n}\frac{1}{N_d}\sum_{n=1}^{N_d}\frac{\tau_n^2}{2} \underset{N_d\to\infty}{\to} \frac{1}{2}\frac{\mathbb{E}\left[\tau^2\right]}{\mathbb{E}\left[\tau\right]} = \frac{1}{\lambda(a)} = r(a), \tag{4}$$

where the average in this computation is taken with respect to the Poisson process interval PDF $P(\tau)$ made explicit above.

One may be tempted to use the estimator $\hat{r}'_D(a) = \frac{1}{N_d}\sum_{n=1}^{N_d}\tau_n$ instead of the estimator $\hat{r}_D$ defined by (3). For an actual Poisson process, that would just give the same result. However this estimator would be more sensitive to the effect of a finite correlation time, since the contributions from time intervals $\tau_n \approx \tau_c$ between successive events will only become negligible linearly in $\tau_c/r(a)$, as opposed to quadratically in formula (3).

From now on, we shall assume that the statistics of rare events are Poissonian. This is a reasonable approximation for many dynamical systems as long as there is a well-defined mixing time after which the initial conditions are forgotten. Of course, it would not hold for systems with long-term memory. Note that this assumption is similar to the *independent interval approximation* used in the context of persistence [50]. In the next paragraph, we use this assumption to derive new expressions that allow for accurate and efficient sampling of the return times.

2.1.3. Sampling return times for rare events. In this section we present an alternative way to compute return times, that provides an easier and more efficient way to draw return time plots for rare events than using the direct estimator (3). Let us divide the timeseries $\{A(t)\}_{0\leqslant t\leqslant T_d}$ in M blocks of duration $\Delta T \gg \tau_c$, so that $T_d = M\Delta T$, and let us define the block maximum

$$a_m = \max\left\{A(t)\,|(m-1)\Delta T \leqslant t \leqslant m\Delta T\right\}, \tag{5}$$

and $s_m(a) = 1$ if $a_m > a$ and 0 otherwise, for $1 \leqslant m \leqslant M$.

For rare events, i.e. $r(a) \gg \tau_c$, the number of events $N(t) = \sum_{m\leqslant\lceil t/\Delta T\rceil} s_m(a)$ is well approximated by a Poisson process with density $\lambda(a) = 1/r(a)$. Then, assuming $\tau_c \ll \Delta T \ll r(a)$, the probability $q_m(a)$ that a_m be larger than a is well approximated by $q_m(a) \simeq \Delta T/r(a)$. As $q_m(a)$ can be estimated by $\frac{1}{M}\sum_{m=1}^{M} s_m(a)$, an estimator of $r(a)$ is the *block maximum estimator*:

$$\hat{r}_B(a) = \frac{T_d}{\sum_{m=1}^{M} s_m(a)}. \tag{6}$$

This is the classical method for computing the return time of rare events, valid when $\Delta T \ll r(a)$ [58].

We now introduce a new, more precise estimator, also valid when $\Delta T/r(a)$ is of order one. It is obtained by using $q_m(a) = 1 - \mathrm{e}^{-\Delta T/r(a)}$. Then, a better estimator of $r(a)$ is the *modified block maximum estimator*:

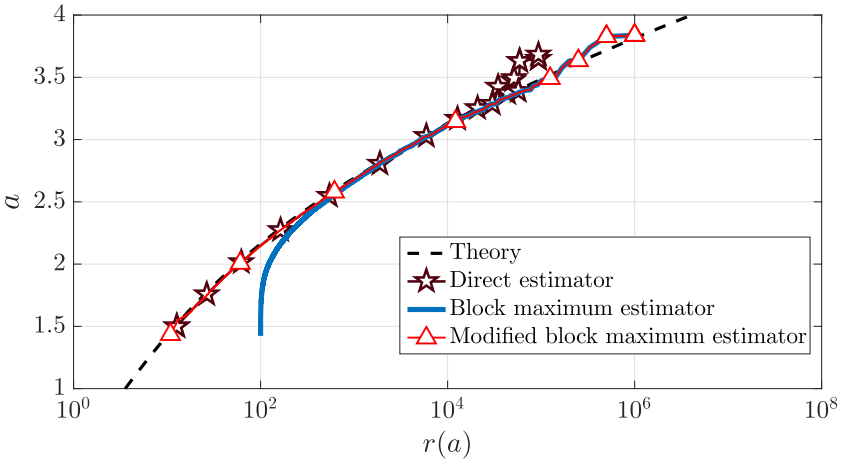

Figure 3. Return time plots for the Ornstein–Uhlenbeck process (22) with $\epsilon = 1/2$, $\alpha = 1$, estimated from a timeseries of length $T_d = 10^6$ using the direct estimator $\hat{r}_D$ (3) (pentagrams), the block maximum estimator $\hat{r}_B$ (6) ($\Delta T = 100$, solid blue line), and the enhanced block maximum estimator $\hat{r}'_B$ (7) ($\Delta T = 100$, solid red line and white triangles). These estimates are compared to the analytical solution (A.6) (dashed black line).

$$\hat{r}'_B(a) = -\frac{\Delta T}{\ln\left(1 - \frac{1}{M}\sum_{m=1}^{M} s_m(a)\right)}. \tag{7}$$

To compute these estimators in practice, we sort the sequence $\{a_m\}_{1\leqslant m\leqslant M}$ in decreasing order and denote the sorted sequence $\{\tilde{a}_m\}_{1\leqslant m\leqslant M}$ such that $\tilde{a}_1 \geqslant \tilde{a}_2 \geqslant ... \geqslant \tilde{a}_M$. Based on (6), we then associate to the threshold $\tilde{a}_m$ the return time $r(\tilde{a}_m) = M\Delta T/m$. Indeed, $\sum_{\ell=1}^{M} s_\ell(\tilde{a}_m) = m$, which means that m events with amplitude larger than $\tilde{a}_m$ have been observed over a duration $M\Delta T$. Alternatively, using the more precise estimator $\hat{r}'_B$ (7) we associate to the threshold $\tilde{a}_m$ the return time $r\left(\tilde{a}_m\right) = -\frac{\Delta T}{\log\left(1-\frac{m}{M}\right)}$. The return time plot represents $\tilde{a}_m$ as a function of $r\left(\tilde{a}_m\right)$, as illustrated for instance on figure 3. Let us stress again that formulas (6) and (7) and this method of plotting the return time are meaningful only if doing block maxima, and for ranges of parameters such that $\tau_c \ll \Delta T \ll r(a)$ for (6) or $\tau_c \ll \Delta T$ for (7).

Figure 3 illustrates the three methods for computing return times from a timeseries: from the definition (3) and the two formulas (6) and (7). The sample timeseries used in this figure is extracted from an Ornstein–Uhlenbeck process, for which the return time curve can also be computed analytically. One can see that both formulas (6) and (7) lead to the same estimate for events with $r(a) \gg \Delta T$. However, formula (6) fails to yield a correct estimate as soon as $r(a) \simeq \Delta T$.

For rare events, plotting return times using (6), as is classically done, proves itself much more convenient and efficient than the naive sampling using (3). It is important to note however, that the use of (6) is valid only after computing maxima over an interval of duration ΔT much larger than τ_c, a remark that was not considered in many previous publications. Moreover, the generalisation (7) we propose in this paper is much more accurate for events with a return time of order of ΔT. This procedure to

compute return time plots can also be generalised in combination with the use of rare event algorithms, as we shall see in the next section.

2.2. Computing return times from a rare event algorithm

In section 2.1, we defined the return time for a time-homogeneous stochastic process and explained how to efficiently compute it for rare events from a timeseries. However, a major difficulty remains as we still have to generate numerically the rare events in the timeseries, which comes at a large computational cost. In the present section, we explain how to apply the above method to the data produced by algorithms designed to sample efficiently rare events instead of direct simulations.

Rare event algorithms provide an effective ensemble of M trajectories $\{X_m(t)\}_{0\leqslant t\leqslant T_a}$ $(1 \leqslant m \leqslant M)$. Note that the length T_a of the trajectories generated by the algorithm does not necessarily coincide with the length T_d of the trajectory generated by direct sampling: in practice, as we shall see, $T_a \ll T_d$. For each of these trajectories, we compute the maximum of the observable over the time evolution $a_m = \max_{0\leqslant t\leqslant T_a}(A(X_m(t)))$. This is similar to the block maximum method described in section 2.1.3, with each trajectory playing the role of a block. There is however a major difference: unlike in the block maximum method, the different trajectories sampled by the rare event algorithm do not have identical statistical weight. To each trajectory $X_m(t)$, and thus to each maximum a_m, is associated a probability p_m computed by the algorithm. Hence, rather than just a sequence $\{a_m\}_{1\leqslant m\leqslant M}$, rare event algorithms yield a sequence $\{a_m, p_m\}_{1\leqslant m\leqslant M}$. The generalisation of the block maximum formula (7) to non-equiprobable blocks is straightforward and leads to the estimator

$$\hat{r}_A(a) = -\frac{T_a}{\ln\left(1-\sum_{m=1}^{M} p_m s_m(a)\right)}. \tag{8}$$

Of course, we could construct similarly an estimator generalising (6), but as we have seen in the previous section, the estimator (7) yields better performance.

In practice, to plot the return time curve, we sort the sequence $\{a_m, p_m\}_{1\leqslant m\leqslant M}$ in decreasing order with respect to the a_m, and denote the sorted sequence $\{\tilde{a}_m, \tilde{p}_m\}_{1\leqslant m\leqslant M}$ such that $\tilde{a}_1 \geqslant \tilde{a}_2 \geqslant ... \geqslant \tilde{a}_M$. We then associate to the threshold $\tilde{a}_m$ the return time

$$\hat{r}_A(\tilde{a}_m) = -\frac{T_a}{\ln\left(1-\sum_{\ell=1}^{m} \tilde{p}_\ell\right)}. \tag{9}$$

Indeed, the sum of the weights of the events with amplitude larger than $\tilde{a}_m$ is $\sum_{\ell=1}^{m} \tilde{p}_\ell$. Again, the return time plot represents a as a function of $r\,(a)$.

We stress that the method described here does not depend on the observable of interest, or on the details of the algorithm itself. In the remainder of the paper, we provide a *proof-of-concept* for this method, by considering two kinds of observables, sampled by two different algorithms: first, we study the return times for instantaneous observables using the AMS algorithm (section 3), then we turn to time-averaged observables using both the AMS and the GKTL algorithm (section 4). We show that the method allows to accurately compute return times at a much smaller computational cost than direct simulation. In both cases, we apply the technique to the simple

J. Stat. Mech. (2018) 043213

case of an Ornstein–Uhlenbeck process, for which the results are easily compared with direct simulation and theoretical predictions, before illustrating the potential of the method for applications in complex systems (section 5).

3. Return times sampled with the AMS algorithm

In this section, we present the computation of return times by applying the method presented in section 2.2 to a rare event algorithm known as AMS. This algorithm follows the strategy of *splitting methods* for the estimation of rare event probabilities, which dates back to the 1950s [59]. Many variants have been proposed since then. The AMS algorithm can be interpreted as simulating a system $\{x_i(t)\}$ of interacting replicas (instead of independent replicas in a crude Monte Carlo simulation), with some *selection and mutation* mechanism. We describe this mechanism in section 3.1 as a method to sample trajectory space. This contains all the necessary details for practical use of the algorithm. Then, in section 3.2 we connect the procedure to the general framework of the AMS algorithm, which enables us to directly benefit from the available mathematical results. In section 3.3, we explain what is the optimal choice of score function for our problem and we analyse its behaviour. In section 3.4, we show how the algorithm enables us to estimate return times, under the Poisson statistics assumption made above. Finally, we illustrate in section 3.5 the method by computing the return times for an Ornstein–Uhlenbeck process.

3.1. The TAMS algorithm

The classical AMS algorithm is based on the evolution of an ensemble of trajectories, based on selection-mutation rules, in order to compute rare event probabilities, and more generally committor functions. Return times can not be estimated directly from a committor function and require the estimation of trajectory statistics. The method we propose to compute return times involves the estimation of probabilities of trajectories with a fixed duration T_a. In order to deal with this, we propose a specific modification of the classical AMS algorithm, called trajectory adaptive multilevel splitting.

While the classical AMS algorithm requires to specify only a real-valued *score function* ξ—also called a *reaction coordinate* in many works, due to connections with molecular dynamics simulations, see [56], and also [60, section 4.3]—the TAMS requires in general a time dependent score function, see section 3.3 for the optimal choice.

We consider a continuous time Markov model able to generate trajectories. It can be either a stochastic process, for instance a diffusion, or a chaotic deterministic dynamical system. Let us now describe the algorithmic procedure.

We start by simulating N independent trajectories, denoted $\{x_n^{(0)}(t)\}_{1\leqslant n\leqslant N}$, for a fixed duration T_a. To each of these trajectories, we associate a weight $w_0 = 1$. Then, at iteration $j \geqslant 1$, we evaluate the performance of all replicas $\{x_n^{(j-1)}(t)\}_{1\leqslant n\leqslant N}$ at iteration $j-1$, measured by the maximum of the score function ξ over the whole trajectory:

J. Stat. Mech. (2018) 043213

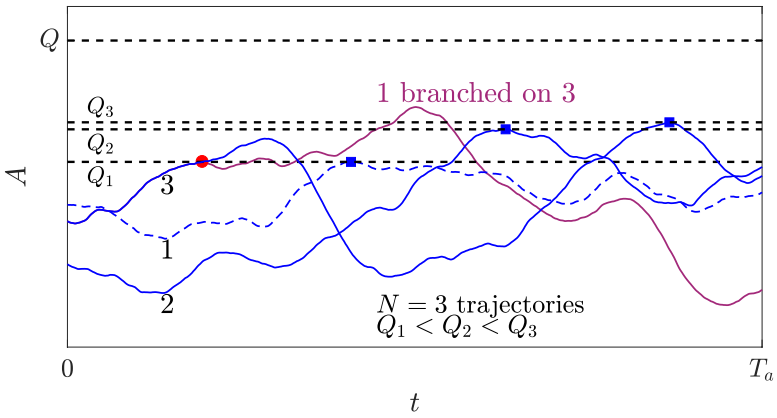

Figure 4. Illustration of one selection-mutation step in the AMS algorithm for the computation of the probability that an observable $A : \mathbb{R}^d \to \mathbb{R}$ reaches values larger than Q over a trajectory of duration T_a.

$$\mathcal{Q}_n^{(j)} = \sup_{0 \leqslant t \leqslant T_a} \xi(t, x_n^{(j-1)}(t)). \tag{10}$$

We select the trajectories corresponding to the lowest $\mathcal{Q}_n^{(j)}$: let us denote $\mathcal{Q}_j^\star = \min_{1 \leqslant n \leqslant N} \mathcal{Q}_n^{(j)}$ and $n_{j,1}^\star, \ldots, n_{j,\ell_j}^\star$ the indices such that:

$$\mathcal{Q}_{n_{j,1}^\star}^{(j)} = \cdots = \mathcal{Q}_{n_{j,\ell_j}^\star}^{(j)} = \mathcal{Q}_j^\star. \tag{11}$$

One might expect intuitively that $\ell_j = 1$. This is not necessarily the case, as explained in [60]: because of the discretization of the dynamical equations in the numerical model, two or more trajectories may yield the same level $\mathcal{Q}_n^{(j)}$.

We then proceed to the mutation step. For each trajectory $x_{n_{j,\ell}^\star}^{(j-1)}$ $(1 \leqslant \ell \leqslant \ell_j)$, we choose a trajectory $x_{n_\ell}^{(j-1)}$ $(n_\ell \neq n_{j,1}, \ldots n_{j,\ell_j})$ randomly among the $N - \ell_j$ remaining trajectories, and define the time $t_{j,\ell}$ defined as the smallest time t such that $\xi(t, x_{n_\ell}^{(j-1)}(t)) > \mathcal{Q}_j^\star$. Finally, we define the new replica $x_{n_{j,\ell}^\star}^{(j)}$ by copying the trajectory $x_{n_\ell}^{(j-1)}$ from t_0 to $t_{j,\ell}$, and simulating the rest of the trajectory, from $t_{j,\ell}$ to T_a. For a Markov process, for instance a diffusion, a new realisation of the noise is used in order to simulate the new trajectory from t_j to T_a. For a chaotic deterministic system, a small amplitude noise is added to the initial condition at time t_j. The other trajectories are not modified: $x_n^{(j)} = x_n^{(j-1)}$ for $n \neq n_{j,1}^\star, \ldots, n_{j,\ell}^\star$. The selection-mutation process is illustrated on figure 4. We associate to the trajectories $x_n^{(j)}$ forming the ensemble at step j the weight w_j given by [28, 56, 60]:

$$w_j = \prod_{i=1}^{j} \left(1 - \frac{\ell_i}{N}\right) = \left(1 - \frac{\ell_j}{N}\right) w_{j-1}. \tag{12}$$

Note that we could mutate more replicas at each step by selecting an arbitrary number of levels $\mathcal{Q}_n^{(j)}$, instead of just the minimum $\mathcal{Q}_j^\star$ as described above. The particular case described above is sometimes referred to as the *last particle method* [61].

The selection-mutation process is iterated J times (two possible definitions of J are given below). The number of resampled trajectories is given by $\tilde{J} = \sum_{j=1}^{J} \ell_j$. Note that

$\tilde{J} \geqslant J$, but the two need not necessarily coincide. In the end, the algorithm generates $M = N + \tilde{J}$ trajectories, given explicitly by the set $\{x_n^{(0)}\}_{1\leqslant n\leqslant N} \cup \{x_{n^\star_{j,\ell}}^{(j)}\}_{1\leqslant \ell\leqslant \ell_j, 1\leqslant j\leqslant J}$, or equivalently, the set $\{x_n^{(J)}\}_{1\leqslant n\leqslant N} \cup \{x_{n^\star_{j,\ell}}^{(j-1)}\}_{1\leqslant \ell\leqslant \ell_j, 1\leqslant j\leqslant J}$. Each trajectory has an associated weight, given by the iteration until which it was a member of the ensemble: w_J for the final trajectories $\{x_n^{(J)}\}_{1\leqslant n\leqslant N}$, and w_{j-1} for the trajectories $\{x_{n^\star_{j,\ell}}^{(j-1)}\}_{1\leqslant \ell\leqslant \ell_j, 1\leqslant j\leqslant J}$ mutated at iteration $1 \leqslant j \leqslant J$. Let us relabel these trajectories and their associated weights as $\{(x_m, w_m)\}_{1\leqslant m\leqslant M}$. Normalising the weights with $W = \sum_{m=1}^{M} w_m$, we obtain the probabilities $p_m = w_m/W$ associated with the trajectories.

Note that instead of just one realisation of the algorithm, one may carry out K independent realisations, thus yielding $M = \sum_{k=1}^{K}(N_k + \tilde{J}_k)$ trajectories with the associated weights, where N_k and $\tilde{J}_k$ denote the number of initial trajectories and resampled trajectories for realisation k, respectively. The probabilities for the trajectories are computed as above.

For any observable $O[x(t)]$, we can define an estimator based on our sampling of trajectory space:

$$\hat{O}_M = \sum_{m=1}^{M} p_m O[x_m(t)]. \tag{13}$$

For practical applications, we shall be interested in two particular cases:

- Instantaneous observable: $O[X,t] = A(X(t))$, for some time-independent observable $A : \mathbb{R}^d \to \mathbb{R}$.
- Time-averaged observable: $O[X,t] = \frac{1}{T}\int_{t-T}^{t} A(X(s))\mathrm{d}s$ for some time-independent observable $A : \mathbb{R}^d \to \mathbb{R}$ and prescribed width T for the averaging window. Note that this is a case where the time-dependent observable O is defined on a different interval than the original process X, here $[T, T_a]$.

The number of iterations J can either be a prescribed integer (in that case the stopping criterion for the algorithm is simply $j = J$), or a random number such that all the trajectories in the ensemble reach a threshold level $\mathcal{Q}$ (the stopping criterion is then $\mathcal{Q}_n^{(j)} > \mathcal{Q}$ for all $1 \leqslant n \leqslant N$). The latter case is more common in existing AMS implementations, however both cases are covered by the general framework developed in [60], and give consistent results. We further discuss these two possible choices in section 3.4.

Let us now estimate the computational cost of an AMS run. The number of trajectories generated by an AMS run is $M = N + \tilde{J}$, as pointed out above. Each resampled trajectory is not simulated over the whole duration T_a, but over $\tau < T_a$, with τ a random number depending on the branching point. We thus define $\gamma \in [0,1]$ so that $\mathbb{E}[\tau] = \gamma T_a$ is the average duration of the resampled part of a mutated trajectory. Performing K identical and independent realisations of the AMS algorithm, the average computational cost associated with a given experiment is then approximately

$$\mathcal{C} = K \times (N + \gamma J) T_a. \tag{14}$$

J. Stat. Mech. (2018) 043213

3.2. Connection with the AMS algorithm for time-dependent observables

In this section, we describe the connection between the TAMS algorithm and the classical AMS algorithm. The aim is to deduce the mathematical properties of the TAMS algorithm from the known ones for the AMS algorithm. For instance we will conclude that the optimal score function is the committor function (17). This section can be skipped by the reader interested in the algorithm only, without trying to understand the mathematical aspects.

The AMS algorithm has originally been designed [28] to efficiently and accurately estimate probabilities of rare events of the type $\mathbb{P}_{x_0,t_0}(\tau_{\mathcal{B}} < \tau_{\mathcal{A}}) \in (0,1)$: the probability that a Markov process $\left(X_t\right)_{t\geqslant t_0}$, initialised with $X_{t_0} = x_0$, hits a set $\mathcal{B}$ before hitting a set $\mathcal{A}$ (with $\mathcal{A}\cap\mathcal{B} = \emptyset$), where $\tau_{\mathcal{C}} = \inf\{t > t_0; X_t \in \mathcal{C}\}$ is the hitting time of a set $\mathcal{C}$. In this section, we show how the problem of estimating the maximum value of a time-dependent observable over a trajectory (which later will be used to estimate return times) falls within the scope of the AMS algorithm. This enables us to benefit directly from the theoretical properties of the AMS algorithm. Some recent mathematical results about the algorithm are reviewed in appendix B. This review is not exhaustive, see for instance [60] and references therein.

We consider a $\mathbb{R}^d$-valued Markov process $\left(X_t\right)_{t\in[0,T_a]}$, with continuous trajectories, for some fixed final time T_a, and a time-dependent observable $O[X,t]$: this is a (time-dependent) functional of the process X, taking value in $\mathbb{R}$. It may be defined for times belonging to a subset of $[0,T_a]$, but for simplicity we shall still denote T_a the final time. The aim is to estimate the probability that the observable reaches a threshold a at some point of the trajectory, i.e.

$$q(a) = \mathbb{P}_{x_0,0}\left[\max_{0\leqslant t\leqslant T_a} O[X,t] > a\right]; \tag{15}$$

(the notation $\mathbb{P}_{x_0,t_0}$ means the probability over realisations of the Markov process with initial condition $X_{t_0} = x_0$). The AMS algorithm provides an estimator $\hat{q}(a)$ for this quantity. Indeed, the event $\{\max_{0\leqslant t\leqslant T_a} O[X,t] > a\}$ can be identified with the event $\{\tau_{\mathcal{B}} < \tau_{\mathcal{A}}\}$ for an auxiliary Markov process Y_t, with an appropriate definition of the sets $\mathcal{A}$ and $\mathcal{B}$, as follows:

$$Y_t = (t, O[X,t]) \in [0,T_a]\times\mathbb{R}, \qquad \mathcal{A} = \{(T_a,z);\ z\leqslant a\}, \qquad \mathcal{B} = \{(t,z);\ t\in[0,T_a], z > a\}. \tag{16}$$

Note that Y is not necessarily a time-homogeneous process. In section 3.1, we have described the TAMS algorithm that gives a procedure to sample the process Y to provide a good estimate of $q(a)$, based on a score function ξ, which measures the distance between $\mathcal{A}$ and $\mathcal{B}$ (in many implementations of the AMS, $\xi(\partial\mathcal{A}) = 0$ and $\xi(\partial\mathcal{B}) = 1$). We describe the corresponding estimator $\hat{q}(a)$, and the related estimator for return times, in section 3.4.

It follows from the above paragraph that the convergence properties of the TAMS algorithm are a direct consequence of the known results for the AMS algorithm (see appendix B). Let us, however, explain in a heuristic way the validity of the algorithm. We refer to [28, 60] for rigorous mathematical arguments.

J. Stat. Mech. (2018) 043213

The algorithm iterates a selection-mutation mechanism on a system of clones. At the selection step, (typically) one clone is removed from the system. To keep a constant number of clones, one new replica needs to be sampled. Statistical consistency is ensured by the introduction of the weights, and appropriate rules for their update. Observe in particular that the sum of the weights of the $N-1$ selected clones, before update, is equal to the sum, after update, of the N clones. At the mutation step, the new clone is sampled by branching one of the selected clones, at the current level. The Markov property of the dynamics is used to sample the end of the trajectory, after crossing the current level. This ensures that, after the mutation step, the N clones observed after the first crossing of the current level, are (conditionally) independent and identically distributed. Observe that they also have the same weight. Eventually, at the last iteration, all the N clones reach the (rare) event of interest, by construction. The weights (or equivalently the random number of iterations) are used to estimate the probability of this event. First, consider non-adaptive versions of the multilevel splitting algorithm, where the levels and the number of iterations are fixed, as originally developed in [59]. This consists in a decomposition of the probability of the rare event, as a product of conditional probabilities. The weights are then products of standard estimators of these conditional probabilities. In the adaptive versions, initially developed in [28], the levels are computed on-the-fly as empirical quantiles: the minima of scores among N clones. The factor $1-1/N$ can be interpreted as the associated conditional probability, hence the validity of the approach—but the analysis in the adaptive case is more complex.

3.3. The optimal score function

This section is a theoretical discussion of the properties of the optimal score function; it may be skipped by readers who are only interested in the application of the TAMS algorithm for computing return times.

As explained in appendix B, the statistical properties, and in particular the variance of the AMS estimator $\hat{q}(a)$, depend on the choice of the score function ξ. The variance is minimal for a particular choice of the score function, sometimes referred to as the *committor*. In a very generic manner, for the AMS algorithm, it is given by $\bar{\xi} = \mathbb{P}[\tau_{\mathcal{B}} < \tau_{\mathcal{A}}]$. In the specific case of the TAMS algorithm, the optimal score function takes the form:

$$\bar{\xi}(t, x; T_a, a) = \mathbb{P}_{x,t}\left[\max_{t \leqslant s \leqslant T_a} O[X, s] > a\right], \tag{17}$$

for all $(t, x) \in [0, T_a] \times \mathbb{R}^d$, where we denote $\mathbb{P}_{x,t}$ the probability over the process initialised at position x at time t, and the threshold a and trajectory duration T_a are fixed parameters. Note that the optimal score function depends both on time and space. Of course, we cannot use this score function in practice, because it is exactly what we are trying to compute. Indeed, as mentioned above, the algorithm ultimately provides an estimate of the probability $q(a) = \bar{\xi}(0, x_0; T_a, a)$. Nevertheless, a crucial point to implement the AMS algorithm is to choose a score function that provides a good approximation of the committor. In practical applications, constructing the score function will often be based on heuristic considerations, but it may also be useful to have theoretical results about the optimal score function.

J. Stat. Mech. (2018) 043213

Here, we want to explain the qualitative properties of the time-dependent committor (17) specific to the TAMS algorithm. For simplicity, we shall only discuss the case of an instantaneous observable: $O[X,t] = A(X_t)$. Moreover, for the precision of the discussion, we assume that the stochastic process X solves the stochastic differential equation $\mathrm{d}X_t = b(X_t)\mathrm{d}t + \sqrt{2\epsilon}\mathrm{d}W_t$, where b is a vector field with a single fixed-point $x_\star$. We further assume that the basin of attraction of $x_\star$ is the full phase space. With this hypothesis, the invariant measure of the diffusion is concentrated close to the attractor $x_\star$ when $\epsilon \ll 1$. Let us assume that the set $\mathcal{C} = \{x \mid A(x) \leqslant 0\}$ is a neighbourhood of $x_\star$ on which most of the invariant measure mass is concentrated. We call $\mathcal{C}$ the attractor. The target set $\mathcal{D} = \{x \mid A(x) \geqslant a\}$ is similarly defined. The hitting times for the sets $\mathcal{C}$ and $\mathcal{D}$ are the random variables given by $\tau_\star = \inf\{t > 0 \mid A(X_t) \leqslant 0\}$ and $\tau_a = \inf\{t > 0 \mid A(X_t) \geqslant a\}$, respectively, where the process is started from a point x at time $t = 0$, such that $0 \leqslant A(x) \leqslant a$. We finally define the static committor $\xi_0(x,a) \equiv \mathbb{P}_{x,0}[\tau_a < \tau_\star]$. The aim of the following discussion is to explain the relation between the time-dependent committor (17) and the static committor $\xi_0(x,a)$.

On the one hand, the time-dependent committor $\bar{\xi}$ satisfies a backward Fokker–Planck equation

$$\frac{\partial \bar{\xi}}{\partial t} = -L[\bar{\xi}], \quad \text{with } L = b_i \frac{\partial}{\partial x_i} + \epsilon \frac{\partial^2}{\partial x_i^2}, \tag{18}$$

in the domain $A^{-1}([0,a]) \subset \mathbb{R}^d$ with boundary condition $\bar{\xi}(t,x;T_a,a) = 1$ for $x \in \partial\mathcal{D}$, and final condition $\bar{\xi}(T_a,x;T_a,a) = 0$. This follows directly from the backward Fokker–Planck equation for the transition probability $P(y,s|x,t)$, and the fact that, with an absorbing boundary condition on $\partial\mathcal{D}$, $\bar{\xi}(t,x;T_a,a) = 1 - \int \mathrm{d}y P(y,T_a|x,t)$. Note that when $T_a - t \gg r(a)$, $\bar{\xi}(t,x;T_a,a) \approx 1$ everywhere ($\bar{\xi}$ converges to 1). On the other hand, $\xi_0(x,a)$ satisfies $L[\xi_0] = 0$, but with different boundary conditions: $\xi_0(x,a) = 1$ if $x \in \partial\mathcal{D}$ and $\xi_0(x,a) = 0$ if $x \in \partial\mathcal{C}$. In the next paragraph, we argue that when $T_a - t$ is much smaller than $r(a)$, the time-dependent committor $\bar{\xi}(t,x;T_a,a)$ given by (17) is well approximated by the static committor $\xi_0(x,a)$, except in two boundary layers: a spatial one of size ϵ for x close to the attractor, and a temporal one of size τ_c for t close to T_a.

Using the notations of section 3.2, the events $\{\tau_{\mathcal{B}} < \tau_{\mathcal{A}}\}$ can be decomposed into the disjoint union of events for which the observable reaches the threshold a before or after hitting 0. The typical time for X to reach $\mathcal{C}$ is the correlation time τ_c. If we assume that $T_a - t \gg \tau_c$, we have the approximation $\bar{\xi}(t,x;T_a,a) \simeq \xi_0(x,a) + [1 - \xi_0(x,a)]\bar{\xi}(t,x_\star;T_a,a)$ (we have used here the approximations $\bar{\xi}(\tau_\star,y;T_a,a) \simeq \bar{\xi}(\tau_\star,x_\star;T_a,a)$ for any $y \in \partial\mathcal{C}$, and $\bar{\xi}(\tau_\star,x_\star;T_a,a) \simeq \bar{\xi}(t,x_\star;T_a,a)$). Moreover, when $T_a - t \ll r(a)$, the Poisson approximation $\bar{\xi}(t,x_\star;T_a,a) \simeq (T_a - t)/r(a)$ holds. To sum up, in the limit $\tau_c \ll T_a - t \ll r(a)$,

$$\bar{\xi}(t,x;T_a,a) \simeq \xi_0(x,a) + \frac{T_a - t}{r(a)}[1 - \xi_0(x,a)]. \tag{19}$$

Let us now introduce the quasipotential V. We note that $\xi_0(x,a) \underset{\epsilon\to 0}{\asymp} \exp(-(\inf_{y\in A^{-1}(\{a\})} V(y) - V(x))/\epsilon)$, while $r(a) \underset{\epsilon\to 0}{\asymp} \exp((\inf_{y\in A^{-1}(\{a\})} V(y))/\epsilon)$. We can

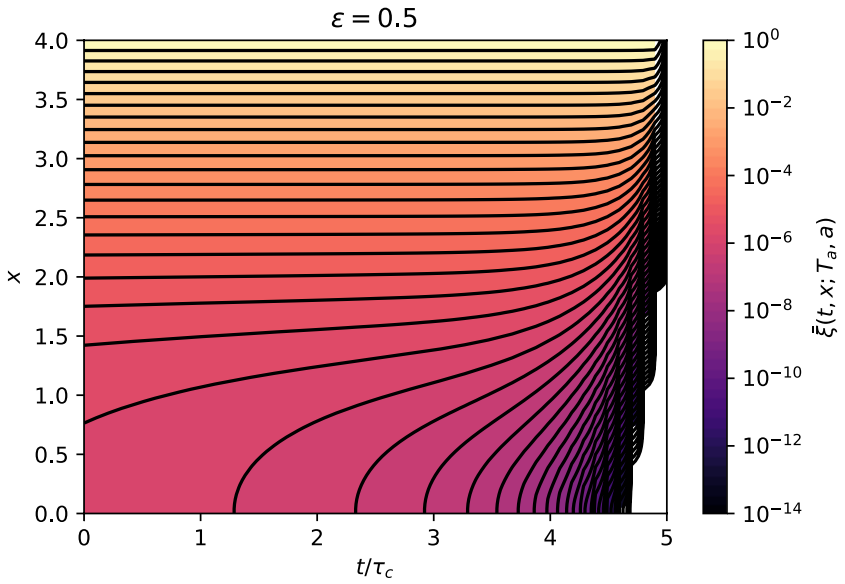

Figure 5. Contour lines of the time-dependent committor $\bar{\xi}(t, x; T_a, a)$ for the Ornstein–Uhlenbeck process (with $\alpha = 1, \epsilon = 1/2$; in particular $\tau_c = 1$), obtained by solving numerically the backward Fokker–Planck equation (18), with $a = 4, T_a = 5$.

thus conclude that $\xi_0(x, a)$ dominates this expression for all x except in a region of size ϵ around the attractor $x_\star$.

As a conclusion, when $T_a - t$ is much smaller than $r(a)$, the time-dependent committor $\bar{\xi}(t, x; T_a, a)$ (17) is well approximated by the static committor $\xi_0(x, a)$, except in two boundary layers: a spatial one of size ϵ for x close to the attractor, and a temporal one of size τ_c for t close to T_a. This is illustrated in figure 5, representing the committor $\bar{\xi}(t, x; T_a, a)$ for the Ornstein–Uhlenbeck process (with $\alpha = 1, \epsilon = 1/2$), obtained by solving numerically the backward Fokker–Planck equation (18), with $a = 4, T_a = 5$.

3.4. Computing return times

As explained in section 3.1, the algorithm generates an ensemble of M trajectories $x_m(t)$ with associated probability p_m. It follows directly from (13) that an estimator of $q(a)$ is:

$$\hat{q}_M(a) = \sum_{m=1}^{M} p_m s_m(a), \tag{20}$$

where $a_m = \max_{0 \leqslant t \leqslant T_a} A(x_m(t))$ is the maximum value for the observable over the trajectory m, p_m the associated probability (see section 3.1), and $s_m(a) = 1$ if $a_m > a$, 0 otherwise (2.1.3).

As explained in section 2.1.3, the return time is related to $q(a)$ by the hypothesis that these events are Poissonian, and we obtain the estimator for the return time $\hat{r}_M(a) = \frac{-T_a}{\ln(1-\hat{q}_M(a))}$ given by (8) (alternatively, we could use $\hat{r}_M(a) = \frac{T_a}{\hat{q}_M(a)}$). In essence, to draw return time plots, it suffices to sort the set $\{(a_m, p_m)\}_{1 \leqslant m \leqslant M}$ according to the a_m and use (9), as described in section 2.2. Note that in practice, with the particular choice of score function $\xi(t, x) = A(x)$, storing the levels $\mathcal{Q}_n^{(j)}$ for the killed trajectories directly provides the corresponding values a_m.

J. Stat. Mech. (2018) 043213

By definition, the estimators $\hat{q}_M(a)$ and $\hat{r}_M(a)$ are random variables. In appendix B, we describe their statistical properties, and how to interpret them in terms of consistency and efficiency of the AMS algorithm. In particular, we show that $\hat{q}_M(a)$ is an unbiased estimator of $q(a)$, study the variance, and show the existence of a central limit theorem.

In section 3.1, we proposed two choices for the number of iterations in the algorithm. First, we described the algorithm with a fixed number of iterations J. Alternatively, as is often seen in the AMS literature, one may decide to iterate the algorithm until all trajectories reach set $\mathcal{B}$. Then J is a random number. In that case, the threshold a which defines the set $\mathcal{B}$ becomes the control parameter for the stopping criterion. Under those circumstances, the estimator $\hat{q}_M$ can be expressed as

$$\hat{q}_M(a) = \prod_{j=1}^{J} \left(1 - \frac{\ell_j}{N}\right). \tag{21}$$

This formula remains valid in the case where the number of iterations J is prescribed: it suffices to define the set $\mathcal{B}$ *a posteriori*, by choosing $a = \min_{1\leqslant n\leqslant N} a_n^{(J)}$ the minimum value of the a_m among the final trajectories. The formula could also be used to compute $\hat{q}_M(b)$ with $b < a$, simply by changing the number of iterations required to meet the stopping criterion. In practice, the easiest approach is to use the expression given in (20).

In the above, we have defined the AMS estimators $\hat{q}_M$ and $\hat{r}_M$ based only on the number of trajectories generated by the algorithm. In fact, the N initial trajectories and the $\tilde{J}$ resampled trajectories (generated during the J iterations) are qualitatively different. In practice, the user does not choose the parameter M directly, but rather the number of ensemble members N on the one hand, and either the threshold a or the number of iterations J on the other hand. As explained in appendix B, the number of initial trajectories N governs the convergence of the estimators. Another practical constraint on the choice of N is the problem of *extinction*: for some systems, if N is too small, all the members of the ensemble become identical after a number of iterations. The other parameter (the threshold a or the number of iterations J) selects the type of events we are interested in. Indeed, from (21), we obtain an approximate relation between the number of resampled trajectories $\tilde{J}$ and the target return times: we write $\ln \hat{q}_M(a) = \sum_{j=1}^{J} \ln\left(1 - \frac{\ell_j}{N}\right)$. For large N, this leads to $\ln \hat{q}_M(a) \approx -\sum_{j=1}^{J} \ell_j/N \approx -\tilde{J}/N$. Targeting rare events with probability $10^{-\beta}$, i.e. return times of order $10^{\beta} T_a$, $\tilde{J}$ is then $\mathcal{O}(N\beta)$. This indicates how to choose the number of iterations J in practice. In particular, for rare events, we should often be in the regime $J = N\beta$.

To sum up, to compute return time plots $r(a)$, one may either fix the target amplitude a, and run the algorithm for a random number of iterations, until the observable reaches a for all the trajectories (i.e. until all the trajectories reach set $\mathcal{B}$), or fix the target return time $r(a)$, and iterate the algorithm a fixed number of times by choosing $J = N \ln(r(a)/T_a)$. In the former case, the prescribed amplitude a needs not correspond to the largest event for which we should estimate the return time, but it will approximately be the case as soon as $N \ll J$, i.e. if a is large enough for fixed N. Similarly, in the latter case, the largest return time computed by the algorithm will approximately be equal to the prescribed target return time when $N \ll J$.

J. Stat. Mech. (2018) 043213

Please note that this method computes the probability to exceed a threshold a, by averaging over trajectories or over K algorithm realisations the sampled value of $q(a)$. This gives an unbiased estimator of $q(a)$, as explained in appendix B. The standard deviation of this estimator is of order $1/\sqrt{KN}$. When computing $r(a)$ through the nonlinear relation $\hat{r}(a) = -T_a/\ln(1-\hat{q}(a))$, we thus obtain an estimator of $r(a)$ with a bias of order of $1/(KN)$ and a standard deviation of order $1/\sqrt{KN}$. If however we had made averages over return times among algorithm realisations, then the estimator for each realisation would have been biased with a bias of order $1/N$ (see appendix B), and the final estimator after K realisations would still be biased with a bias of order $1/N$.

3.5. Return times for the Ornstein–Uhlenbeck process from the TAMS algorithm

We consider the Ornstein–Ulhenbeck process X_t defined as

$$\mathrm{d}X_t = -\alpha X_t \mathrm{d}t + \sqrt{2\epsilon}\mathrm{d}W_t \tag{22}$$

with $\alpha = 1$ and $\epsilon = 1/2$. The correlation time is $\tau_c = 1$ and the variance is $\sigma^2 = 1/2$. We now illustrate the use of the TAMS algorithm for computing the return times $r(a)$ for the variable X_t being larger than a threshold a. This amounts to choose the observable as $A(x) = x$. We use the TAMS algorithm described in section 3.1 with a score function $\xi(x,t) = x$. This choice of score function is motivated by the fact that the optimal score function is nearly independent of time, except on a small boundary layer, as explained in section 3.3, and that in dimension 1, the level set of x will be the same as the level set of the static committor function.

The algorithm relies on three numerical parameters : the length of the generated trajectories T_a, the maximum threshold value a_{max} and the number of replicas N. As explained in appendix B, the relative error depends on N. Additionally, one has to choose $T_a \ll \tau_c$, as explained in section 2.1.3. We see empirically that a good trade-off between this requirement and computational burden is to choose trajectories of length T_a equal to a few correlation times.

Figure 6 shows the return time plot computed using $N = 100$ replicas, $T_a = 5\tau_c$ and $a = 7\sigma$, using the TAMS in conjunction with the methodology described in section 3.4. For comparison, figure 6 also features the theoretical value, estimated by computing the mean first-passage time (see appendix A), and the estimate obtained from a direct sampling with the same computational cost as the TAMS run. We see that return times are very well recovered by the TAMS algorithm. Furthermore, figure 6 clearly illustrates the computational gain from the TAMS algorithm. Indeed, for the same computational cost as direct sampling, the use of the TAMS algorithm gives access to return times for much rarer events: we can now accurately compute return times on the order of 10^{13}, about seven orders of magnitude larger than direct sampling.

4. Return times sampled with the Giardina–Kurchan–Tailleur–Lecomte algorithm

In this section, we illustrate the computation of return times using the method described in section 2.2 for a *time-averaged* observable. Even though it could be done using the TAMS algorithm presented in section 3.2, we instead illustrate the use of a different

J. Stat. Mech. (2018) 043213

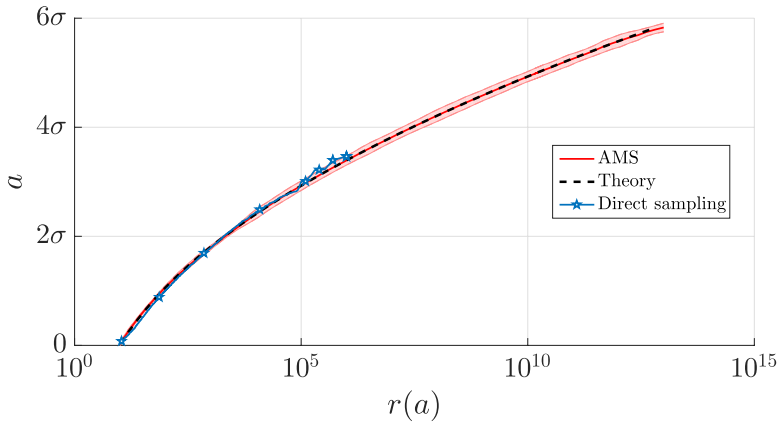

Figure 6. Return time plot for a random variable following an Ornstein–Uhlenbeck process (22) with $\alpha = 1$ and $\epsilon = 1/2$ ($\sigma = 1/\sqrt{2}$ is the standard deviation). The solid red line represents the estimate obtained using the TAMS with $N = 100$ replica, $T_a = 5\tau_c$ and $a = 7\sigma$. The total number of trajectories (both initial and resampled) is $M \approx 2 \times 10^3$ so that the total computational cost is $\mathcal{O}(10^6\tau_c)$. It is compared to the modified block maximum estimator $\hat{r}'_B$ applied to a sample timeseries of length $T_d = 10^6\tau_c$ (blue stars) and to the analytical result (A.6). The shaded area represents the confidence interval on the estimation of the fluctuation amplitude a, for a fixed value for the return time $r(a)$. It is computed as the empirical mean over the interpolated return time curves originating from the independent realisations of the algorithm.

rare-event algorithm, specifically designed to compute large deviations of time-averaged dynamical observables: the GKTL algorithm [29, 57, 62].

4.1. The algorithm

The underlying idea of the GKTL algorithm is to perform a biased sampling of trajectory space. It relies on the simulation of a population of trajectories, which, unlike direct Monte-Carlo methods, interact dynamically: at regular time intervals, some members of the ensemble are killed and some are cloned according to a weight which depends on the history of the replica. The weights are chosen such that, after several iterations of the algorithm, generated trajectories are distributed according to a probability distribution that is tilted in order to favour trajectories with large values of a chosen time averaged observable. This sort of algorithm has first been proposed by [57] and has been used to study rare events in both stochastic [57, 63–65] and deterministic systems [57, 62]. The idea of sampling quantities of interest from a distribution biased in a controlled way is very general; it is referred to as *importance sampling*, and was used in many different contexts (see e.g. [66, 67] and the general references [24, 25]).

More precisely, we perform simulations of an ensemble of N trajectories $\{X_n(t)\}$ (with $n = 1, 2, ..., N$) starting from random initial conditions. Like in section 3, the total integration time of the trajectories is denoted T_a. We consider an observable of interest $A(X(t))$ and a resampling time τ. At times $t_i = i\tau$ (with $i = 1, 2, ..., T_a/\tau$) we assign to each trajectory n a weight W_n^i defined as

$$W_n^i = \frac{\mathrm{e}^{k\int_{t_{i-1}}^{t_i} A(X_n(t))\mathrm{d}t}}{R_i} \quad \text{with} \quad R_i = \frac{1}{N}\sum_{n=1}^{N} \mathrm{e}^{k\int_{t_{i-1}}^{t_i} A(X_n(t))\mathrm{d}t}. \tag{23}$$

For each trajectory X_n, a random number of copies of the trajectory are generated, on average proportional to the weight W_n^i and such that the total number of trajectories produced at each event is equal to N. The parameter k is chosen by the user in order to control the strength of the selection and thus to target a class of extreme events of interest. The larger the value of k, the more trajectories with large values of the time average observable will survive the selection.

As mentioned above, the GKTL algorithm performs importance sampling in the space of trajectories, which is relevant for out-of-equilibrium systems. Let us denote formally $\mathbb{P}_0\left(\{X(t)\}_{0\leqslant t\leqslant T_a} = \{x(t)\}_{0\leqslant t\leqslant T_a}\right)$ the probability to observe a trajectory $\{x(t)\}_{0\leqslant t\leqslant T_a}$ in the model, and $\mathbb{P}_k\left(\{X(t)\}_{0\leqslant t\leqslant T_a} = \{x(t)\}_{0\leqslant t\leqslant T_a}\right)$ the probability to observe the same trajectory with the algorithm. By construction of the algorithm through the weights (23), we have

$$\mathbb{P}_k\left(\{X(t)\}_{0\leqslant t\leqslant T_a} = \{x(t)\}_{0\leqslant t\leqslant T_a}\right) \underset{N\to\infty}{\sim} \frac{\mathrm{e}^{k\int_0^{T_a} A(x(t))\mathrm{d}t}}{Z(k,T_a)} \mathbb{P}_0\left(\{X(t)\}_{0\leqslant t\leqslant T_a} = \{x(t)\}_{0\leqslant t\leqslant T_a}\right) \tag{24}$$

where the normalisation factor is given by $Z(k,T_a) = \mathbb{E}_0\left[\mathrm{e}^{k\int_0^{T_a} A(X(t))\mathrm{d}t}\right]$, denoting by $\mathbb{E}_0$ the expectation value with respect to $\mathbb{P}_0$, and $\underset{N\to\infty}{\sim}$ means that this is true only asymptotically for large N. The typical error is of order $1/\sqrt{N}$ when evaluating averages over observables. Equation (24) is obtained by assuming the mean field approximation

$$R_1 = \frac{1}{N}\sum_{n=1}^{N} \mathrm{e}^{k\int_0^{t_1} A(X_n(t))\mathrm{d}t} \underset{N\to\infty}{\sim} Z(k,t_1) = \mathbb{E}_0\left[\mathrm{e}^{k\int_0^{t_1} A(X(t))\mathrm{d}t}\right], \tag{25}$$

which, by induction, and using a formula similar to (25) at each step of the induction, leads to [29, 57]:

$$\prod_{i=1}^{T_a/\tau} R_i \underset{N\to\infty}{\sim} Z(k,T_a) = \mathbb{E}_0\left[\mathrm{e}^{k\int_0^{T_a} A(X(t))\mathrm{d}t}\right]. \tag{26}$$

The validity of the mean field approximation and the fact that the typical relative error due to this approximation is of order $1/\sqrt{N}$ has been proven [68, 69] to be true for a family of rare event algorithms including the one adopted in this paper.

Formula (24) is valid only for times T_a that are integer multiples of the resampling time τ. The killed trajectories have to be discarded from the statistics. Starting from the final N trajectories at time T_a, one goes backwards in time through the selection events attaching to each piece of trajectory its ancestor. In this way one obtains an effective ensemble of N trajectories from time 0 to time T_a, distributed according to $\mathbb{P}_k$. All trajectories reconstructed in this way are real solutions of the model: we have not modified the dynamics, but only sampled trajectories according to the distribution $\mathbb{P}_k$ rather than according to the distribution $\mathbb{P}_0$.

The GKTL algorithm was initially designed to compute large deviation rate functions [57]. Indeed, using $\lambda(k,T_a) = \frac{1}{T_a}\ln Z(k,T_a)$, the *scaled cumulant generating function*

J. Stat. Mech. (2018) 043213

[19] $\lambda(k) = \lim_{T_a \to +\infty} \lambda(k, T_a)$ can easily be estimated from the algorithm. From there, the large deviation rate function $I(a)$, such that $\mathbb{P}_0\left[\int_0^{T_a} A(X(t))\mathrm{d}t = T_a a\right] \asymp \mathrm{e}^{-T_a I(a)}$, is recovered by the Legendre–Fenchel transform $I(a) = \sup_k(ka - \lambda(k))$ [19]. In fact, the algorithm can be used to compute the statistical properties with respect to the distribution $\mathbb{P}_0$ of any observable, from the distribution $\mathbb{P}_k$. This is done using the backward reconstructed trajectories and inverting formula (24). If, for example, one wants to estimate the expectation value of an observable $O\left(\{X(t)\}_{0\leqslant t\leqslant T_a}\right)$, an estimator is given by

$$\mathbb{E}_0\left[O\left(\{X(t)\}_{0\leqslant t\leqslant T_a}\right)\right] \underset{N\to\infty}{\sim} \frac{1}{N}\sum_{n=1}^{N} O\left(\{X_n(t)\}_{0\leqslant t\leqslant T_a}\right) \mathrm{e}^{-k\int_0^{T_a} A(X_n(t))\mathrm{d}t} \mathrm{e}^{T_a \lambda(k,T_a)}, \tag{27}$$

where the X_n are the N backward reconstructed trajectories. Empirical estimators of quantities related to rare (for $\mathbb{P}_0$) events of the kind of (27) (thus using data distributed according to $\mathbb{P}_k$) have a dramatically lower statistical error, due to the larger number of relevant rare events present in the effective ensemble. In particular, one can use the reconstructed trajectories to compute return times using the method described in section 2.2. Of course, the above formula will not perform well for quantities which are rare for the biased statistics, and we should carefully construct the effective ensemble depending on the class of observables O we are trying to estimate.

4.2. Return times for the time-averaged Ornstein–Uhlenbeck process from the GKTL algorithm

We consider the time averaged position

$$\overline{X}_T(t) = \frac{1}{T}\int_{t-T}^{t} x(s)\mathrm{d}s, \;\; t \in [T, T_a] \tag{28}$$

where the position x follows an Ornstein–Uhlenbeck process (22) between times 0 and T_a. We call σ_T^2 the variance of $\overline{X}_T$ and $\tau_{c,T}$ the correlation time. In this section we illustrate the application of the GKTL algorithm to the computation of the return times $r(a)$ for $\bar{X}_T$ being larger then a. We make use of the GKTL algorithm with $T_a > T$, computing the time-averaged position $\bar{X}_T(t)$ for $T \leqslant t \leqslant T_a$ as a moving average.

Similarly to the case of the TAMS (see section 3.5), the application of the GKTL algorithm depends on three numerical parameters: the number of trajectories N, the length of the trajectories T_a and the bias parameter k. The number of trajectories N governs the relative error, as explained in section 4.1, and one should use T_a so that $T_a - T \gg \tau_{c,T}$, as explained in section 2.1.3. Finally, as for the strength of the selection k, its relation with the amplitude of the generated fluctuations is not known beforehand, and one has to set its value empirically[5].

In figure 7, we show the return times $r(a)$ for $\overline{X}_T$, with $T = 10\tau_c$, computed from the GKTL algorithm described in section 4.1, following the methodology described in section 2.2. In order to validate the computation, the estimate obtained from the algorithm is compared to the direct sampling method (7). For rare events ($r(a) \gg \tau_{c,T}$),

[5] When the duration of the average is long enough so that a *large deviation regime* is attained, the relation between the value of k and the typical amplitude of the fluctuations generated by the algorithm is known from the Gartner–Ellis theorem. See [19] for further details.

J. Stat. Mech. (2018) 043213

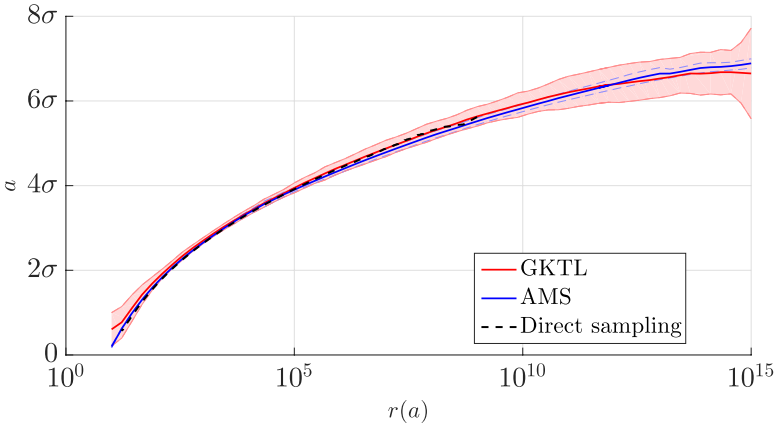

Figure 7. Return time plot for the time-averaged Ornstein–Uhlenbeck process $\overline{X}_T$ (28) with $\alpha = 1$ and $\epsilon = 1/2$ ($\sigma = 1/\sqrt{2}$ is the standard deviation), estimated from the GKTL algorithm (solid red line) and AMS algorithm (solid blue line). The GKTL algorithm was used with $N = 500$ replica, $T_a = 20\tau_c$ and $k = 0.9$. It was repeated $K = 100$ times. The TAMS algorithm was used with $N = 100$ replicas, $T_a = 50$ and $a = 6.5\sigma_T$. It was repeated $K = 10$ times. Finally, the dashed black line represents the result of a direct sampling over a timeseries of length $T_d = 10^9$. Parameters of both the GKTL and AMS algorithms were chosen so that 100 realisations of the algorithms amount to a computational cost of $\mathcal{O}(10^6\tau_c)$. The cost of the direct sampling is $10^9\tau_c$. The shaded area represents the confidence interval on the estimation of the fluctuation amplitude a, for a fixed value for the return time $r(a)$. It is computed as the empirical mean over the 100 interpolated return time curves originating from the 100 independent realisations of the algorithm.

the results from the GKTL algorithm agree well with direct sampling. Furthermore, the comparison of the computational costs for the two different methods shows the efficiency of the algorithm. Indeed, for direct sampling, the length of the sample trajectory, $10^9\tau_c$ in the case of figure 7, naturally sets an upper bound on the return times one is able to compute. By contrast, the total cost of the GKTL estimate is $10^6\tau_c$ and one can see in figure 7 that it allows to reach return times larger by many orders of magnitude. Figure 8 shows an estimate of the PDF for $\bar{X}_T$ along the trajectories generated using the GKTL algorithm. Even though importance sampling is performed for the observable $\bar{X}_{T_a}$, the observable averaged over the whole trajectory of length T_a, it better samples the tail of the PDF for $\bar{X}_T$, resulting in better estimation of the corresponding return times.

Figure 7 also shows the return time for $\bar{X}_T(t)$ being larger than a computed using the TAMS algorithm (see section 3.1). We use as a score function the time-averaged observable itself $\xi(t) = \bar{X}_T(t)$, for $T \leqslant t \leqslant T_a$. The selection is then done according to the maximum value of $\bar{X}_T(t)$ for each trajectory for $T \leqslant t \leqslant T_a$. More precisely, following the notations of section 3.1, for iteration j we denote $\mathcal{Q}_j^\star$ the lowest maximum of $\bar{X}_T$ over the trajectories in the set $\{x_n^{(j)}(t)\}_{0\leqslant t\leqslant T_a, 1\leqslant n\leqslant N}$. Following the TAMS algorithm described in section 3.1, the l_j new replica are defined by copying the trajectories $x_{n_\ell}^{(j-1)}$ from 0 to the smallest time t such that $\bar{X}_{T,n_\ell}^{(j-1)}(t) > \mathcal{Q}_j^\star$ and simulating the rest of the trajectory from this time to T_a.

J. Stat. Mech. (2018) 043213

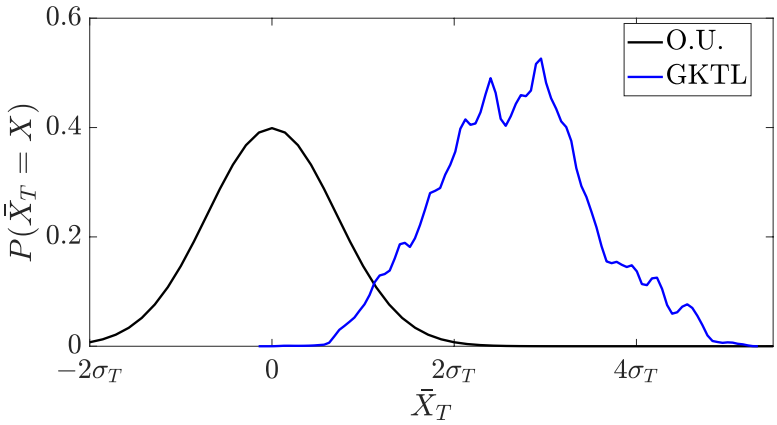

Figure 8. PDF of the time-averaged observable $\bar{X}_T$, with $T = 10\tau_c$, for the Ornstein–Uhlenbeck process with $\alpha = 1$ and $\epsilon = 1/2$ ($\sigma = 1/\sqrt{2}$ is the standard deviation): computed from a direct simulation of length $T_d = 10^6$ (black curve), and based on the trajectories generated by the GKTL algorithm with 500 replicas, $T_a = 20\tau_c$ and $k = 0.9$ (blue curve).

The agreement between the two estimates illustrates that the method to compute return times from rare event algorithms proposed in section 2.2 can be applied to any rare event algorithm suitable for the type of observable under study. Here, while the AMS algorithm allows for computing return times for both the instantaneous and time-averaged observables, the GKTL algorithm is not suited for instantaneous observables.

5. Application: extreme drag force on an object immersed in a turbulent flow

A key issue with rare event algorithms is to understand if they are actually useful to compute rare events and their probabilities for actual complex dynamical systems. The AMS algorithm has shown to be very efficient for partial differential equations with noise [30]. In this section, we give a brief illustration that more complex dynamics can be studied. We illustrate the computation of return times using rare event algorithms for a turbulent flow. The possible limitations of rare event algorithms are further discussed in the conclusion.

Unlike simple low-dimensional models, such as the Ornstein–Uhlenbeck process, numerical simulations of turbulent flows of interest for physicists and/or engineers require tremendous computational efforts. As a consequence, direct sampling of rare events based on a long time series is simply unthinkable for such systems. A common practice in the engineering community is to generate synthetic turbulent flows, without resolving explicitly the small scales, to study numerically the physical phenomena of interest [70, 71]. However, the main difficulty is to capture synthetically the correct long-range (spatio-temporal) correlations of turbulence and such approaches can not capture the essential effects of coherent structures. We show here that rare event methods such as the GKTL and the AMS algorithms can be used in order to study extremes in turbulent flows without having to rely on such modelling.

J. Stat. Mech. (2018) 043213

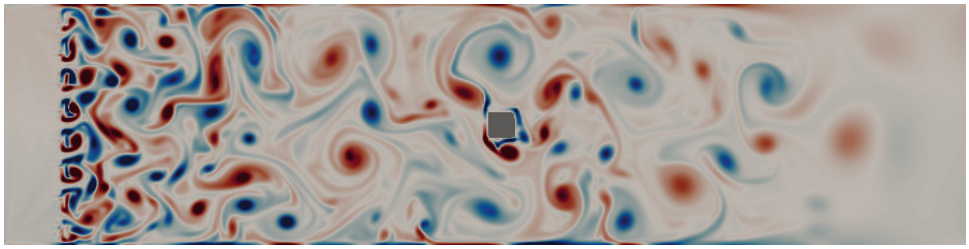

Figure 9. Snapshot of a typical vorticity field of the flow under study. A steady parabolic velocity profile is imposed at the inlet. Turbulence is then generated by a grid. We used the GKTL algorithm the compute the return times of the average drag over the square here marked by the grey area.

The example we consider is the sampling of extreme fluctuations of the mechanical stresses caused by a turbulent flow on an immersed object. Being able to compute flow trajectories associated to such extremes is of great interest both for fundamental issues and applied problems, such as reliability assessment for industrial structures. More specifically, we focus here on the averaged drag $F_T(t) = \frac{1}{T}\int_t^{t+T} f_d(t')\mathrm{d}t'$, which corresponds to the averaged sum of the efforts from the flow, projected along the flow direction. The length of the averaging window depends on the nature of the application. For instance, it could be related to the typical response time of a material, in order to average out high frequency excitation that has a minor impact on the deformation of the structure. Note that the choice of the observable is arbitrary and one could choose to study other related physical quantities, such as the lift or torque.

In order to provide a *proof-of-concept* for such rare events approaches for turbulent flows, we compute the return time for extreme values of the drag in a simple academic flow. The setup we consider, illustrated in figure 9, is that of a two-dimensional channel flow, with a square obstacle immersed in the middle of the domain. Turbulence is generated upstream by means of a grid. This flow is simple enough so that long time series can be obtained in a reasonable amount of computational time, allowing for the computation of reference return times. In practice, we carry out a direct numerical simulation using the lattice Boltzmann method [72], which offers low implementation effort for performances comparable to other methods for such simple geometries and boundary conditions. The application of the GKTL and AMS algorithms to deterministic dynamics requires that some randomness is artificially introduced in the dynamics so that copies originated from the same parent follow different paths. This can be achieved by randomly perturbing the restart state at branching points.

Figure 10 illustrates the computation of the return times for the drag averaged over 5 correlation times using the GKTL algorithm. It shows that the use of the algorithm makes accessible the computation of rare events at a much lower computational cost than direct sampling. More precisely, the algorithm was applied using $N = 128$ replicas simulated over 10 correlation times. The return time curve presented in figure 10 is based on the data from $K = 10$ repetitions of the algorithm, leading to an overall computational cost of, roughly, 10^4 correlation times. From a direct sampling of similar computational cost, the rarest accessible event has a return time close to the computational cost itself, in this case is 10^4. Figure 10 shows that the use of the GKTL algorithm allows for the computation of return times of much rarer events. The reference

J. Stat. Mech. (2018) 043213

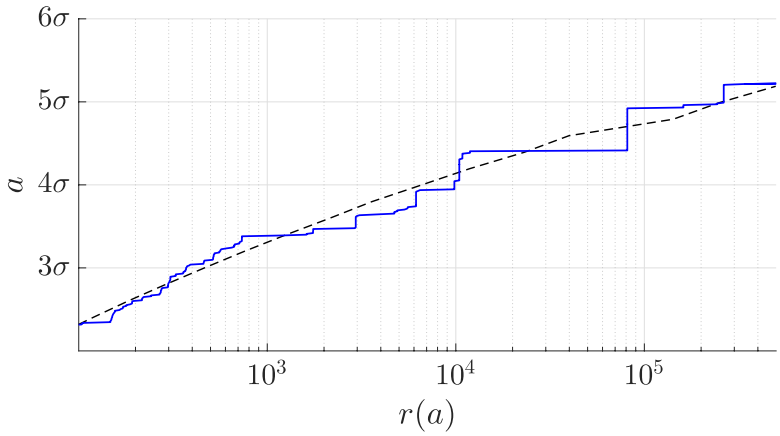

Figure 10. Illustration of the computation of return times for the averaged drag over the square obstacle pictured in figure 9. The averaging window is 5 correlation times. The dashed black line represents the reference return times computed from a timeseries spanning 10^6 correlation times, using (7). The solid blue line represents the return times obtained using the GKTL algorithm.

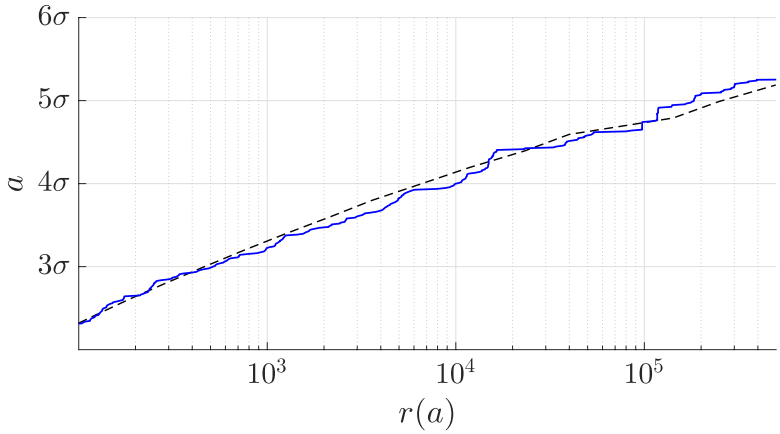

Figure 11. Illustration of the computation of return times for the averaged drag over the square obstacle pictured in figure 9, using 50 repetitions of the GKTL algorithm. The parameters are the same as in figure 10. This figure illustrates the reduction in the occurrence of plateaus for the return time curve obtained using the GKTL algorithm. The dashed black line represents the reference return times. The solid blue line represents the return times obtained using the GKTL algorithm.

curve was computed from a time series spanning 10^6 correlation times. For events having a return time close to $5 \cdot 10^5$ correlation times, the computational cost of estimating the return times using the GKTL algorithm is 50 times lower than direct sampling.

The occurrence of plateaus in figure 10 is due to the increasing multiplicity of trajectories as the amplitude a increases. Indeed, because of the selection procedure involved in the GKTL algorithm, a subset of trajectories can share the same ancestor. Henceforth, they are likely to differ only by a small time-interval at the end of their whole duration. In such cases, it is common that the maximum over the trajectory is attained in earlier times. As a consequence, this subset of trajectories will contribute

J. Stat. Mech. (2018) 043213

the same value to the set of maxima from which return times are computed. This effect is accentuated in the present case of a deterministic system, as it takes some time for copies to separate after being perturbed at a branching point. A straightforward way of mitigating the occurrence of such plateaus is to increase the number of trajectories or/and the number of repetitions of the algorithm. As an illustration, figure 11 shows the return time plot obtained using 50 repetitions instead of 10 in figure 10.

6. Conclusion

In this paper, we have considered the question of estimating the return time of rare events in dynamical systems. We have compared several estimators, using both usual timeseries (generated with direct numerical simulations) and rare event algorithms, by generalising the approach relating the return times to the extrema over trajectory blocks. This approach relies on the fact that rare events behave, to a good approximation, like a Poisson process: this allows for the derivation of a simple formula (see (6)) for estimating the return times based on block maxima. We slightly improved this formula (see (7)), and further showed that it was possible, provided only minor modifications, to evaluate it with data produced by rare event algorithms. Indeed, while the traditional block maximum method consists in dividing a given trajectory in blocks with arbitrary length (larger than the correlation time of the system, and smaller than the return time one seeks to estimate), there is a class of rare event algorithms which yields precisely an ensemble of trajectories exhibiting the rare event more often than direct simulation, together with the probability of observing each member of the ensemble. Hence, we have generalised the block maximum formula to non-equiprobable trajectory blocks; this allowed us to use directly rare event algorithms, such as the AMS and the GKTL algorithm, to estimate return times for rare events. Using the Ornstein–Uhlenbeck process as an illustration, we showed that the method is easy to use and accurately computes return times in a computationally efficient manner. Indeed, compared to direct sampling, combining the generalised block maximum approach to rare event algorithms allowed for computing return times many orders of magnitude larger, at fixed computational cost. This method does not depend on the dynamics of the system or on the type of observable, as long as a suitable rare event algorithm is selected. As an illustration, we computed return time plots for both instantaneous and time-average observables for the Ornstein–Uhlenbeck process, using the AMS and the GKTL algorithms. This approach paves the way to numerical computation of return times in complex dynamical systems. To showcase the potential of the method, we discussed briefly an application of practical interest: extreme values of the drag force on an object immersed in turbulent flows. Another example of application given very recently is the study of heat waves [73].

A key issue with rare event algorithms is to understand if they are actually useful to compute rare events and their probabilities for actual complex dynamical systems. Many of the proposed approaches fail to pass such a test, either because the algorithm is too complex to be used for complex dynamical systems, or the algorithm is restricted to specific systems (equilibrium or reversible dynamics, diffusions with small noises),

J. Stat. Mech. (2018) 043213

or the algorithm simply fails. A key issue with many potentially successful rare event algorithms, for instance the AMS algorithm and the GKTL algorithm among others, is that their success depends much on the quality of the rule used for selecting trajectories. For instance the AMS or the TAMS algorithm rely on a score function, and the GKTL use as a selection rule the increment of a the time average which one aims at computing. Whenever one uses a good score function, those algorithms are extremely useful and show tremendous sampling improvements [30]. For the AMS algorithm, the choice of a good score function often relies on a good rough qualitative understanding by the user of the effective dynamics that leads to the rare events. Then the AMS algorithm leads to excellent quantitative results, even with complex dynamical systems (see for instance [30]). Several examples have illustrated than those algorithm may fail to lead to improvement in other cases, see for instance [74]. Faced with such difficulties, one may either use an empirical approach, or try to improve the algorithms in order to cure potential problems, as we explain now.

The empirical approach consists in identifying *a priori* the conditions for success of the algorithms and identify relevant dynamical phenomena that fulfil these conditions. For the AMS algorithm this amounts to understanding sufficiently well the dynamics, in order to be able to define a macroscopic variable that will describe well the dynamics leading to the extremes, and to propose a related score function. The algorithm may also be used to test some hypothesis on such macroscopic variables, and learn about the dynamics. The GKTL algorithm is usually successful in conditions when the sampling of time averages is dominated by a persistent macroscopic state.

Several authors have proposed new algorithms to cure some of the problems. A class of algorithms seek at changing the dynamics such that the computation will be more efficient (see for instance [75] for diffusions with small noise, or [74] in relation with the GKTL algorithm and references therein). Those methods are limited to diffusions, as they require to relate the statistics of paths for different dynamics, for instance through the Girsanov formula. They can involve recursive learning of an optimal dynamics and be very successful for dynamics with a few degrees of freedom [74]. Another class of algorithms, milestoning (see [76]), is aimed at computing a reduced description of the original dynamics, that can afterwards permits to efficiently compute dynamical quantities, for instance first passage times (see [77] and references therein).

Acknowledgments

The research leading to these results has received funding from the European Research Council under the European Union's seventh Framework Program (FP7/2007-2013 Grant Agreement No. 616811). This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 753021. We would like to thank Emmanuel Lévêque for insightful discussions, and Joran Rolland and two anonymous referees for their comments which helped improving the manuscript. Simulations have been performed on the local HPC facilities at Ecole Normale Supérieure de Lyon (PSMN) and Ecole centrale de Lyon (PMCS2I). These HPC facilities are supported by the Auvergne-Rhône-Alpes region (GRANT CPRT07-13 CIRA) and the national Equip@Meso grant (ANR-10-EQPX-29-01).

J. Stat. Mech. (2018) 043213

Appendix A. Mean first-passage time for the Ornstein–Uhlenbeck process

Throughout the paper, we consider as an example the Ornstein–Uhlenbeck process:

$$\mathrm{d}X_t = b(X_t)\mathrm{d}t + \sqrt{2\epsilon}\mathrm{d}W_t, \tag{A.1}$$

where W_t is the standard Wiener process and the drift term is linear: $b(x) = -\alpha x$. We write the corresponding Fokker–Planck equation for the probability density $P(x,t)$ of the random variable X_t:

$$\frac{\partial P}{\partial t} = LP, \text{ with } LP = -\frac{\partial[b(x)P(x,t)]}{\partial x} + \epsilon\frac{\partial P(x,t)}{\partial x^2}. \tag{A.2}$$

The stationary probability density is $P_s(x) = \sqrt{\frac{\alpha}{2\pi\epsilon}}\mathrm{e}^{-\frac{\alpha x^2}{2\epsilon}}$: $LP_s = 0$. We shall denote the standard deviation by $\sigma = \sqrt{\epsilon/\alpha}$.

For a threshold value a much larger than the standard deviation ($a \gg \sqrt{\epsilon/\alpha}$), the return time $r(a)$ should be well approximated by the mean first-passage time $\mathbb{E}[\tau_a]$, where $\tau_a = \min\{t \geqslant 0 | X_t \geqslant a\}$. Computing the mean first-passage time for such a simple stochastic process is a classical textbook problem (see for instance [53, section 5.5]): we consider the transition probability $P(x',t|x,0)$, which also satisfies the Fokker–Planck equation, with initial condition $P(x',0|x,0) = \delta(x'-x)$.

We now introduce the quantity $G(x,t) = \int_{-\infty}^{a}\mathrm{d}x' P(x',t|x,0)$, with the initial condition $G(x,0) = \chi_{]-\infty,a[}(x)$, where χ is the indicator function, and with absorbing boundary conditions at a, $G(a,t) = 0$. Using the backwards Kolmogorov equation for the transition probability: $\partial_t P(x',0|x,t) = L^\dagger P(x',0|x,t)$, and using time-homogeneity $P(x',t|x,0) = P(x',0|x,-t)$, we see that the evolution of G is also governed by $\partial_t G = L^\dagger G$. $G(x,t)$ is the probability that a particle initially at position x has not reached a after time t. In other words, it is the probability, conditioned on the initial condition x, that $\tau_a > t$. The moments of the first-passage time follow directly:

$$\mathbb{E}_x[\tau_a^n] = -\int_0^{+\infty} t^n \partial_t G(x,t)\mathrm{d}t = n\int_0^{+\infty} t^{n-1}G(x,t)\mathrm{d}t. \tag{A.3}$$

From there, a recursion relation can be obtained for the moments of τ_a:

$$\mathbb{E}_x[\tau_a^n] = -\frac{1}{n+1}\left[b(x)\frac{\partial}{\partial x} + \epsilon\frac{\partial^2}{\partial x^2}\right]\mathbb{E}_x[\tau_a^{n+1}]. \tag{A.4}$$

In particular, with $\mathbb{E}_x[\tau_a^0] = 1$, we obtain an exact formula for $\mathbb{E}_x[\tau_a]$:

$$\mathbb{E}_x[\tau_a] = \frac{1}{\epsilon}\int_x^a \mathrm{d}y \mathrm{e}^{\frac{\alpha y^2}{2\epsilon}}\int_{-\infty}^{y}\mathrm{d}z \mathrm{e}^{-\frac{\alpha z^2}{2\epsilon}} = \frac{\pi}{\alpha}\left\{\mathrm{erfi}\left(\sqrt{\frac{\alpha}{2\epsilon}}a\right) - \mathrm{erfi}\left(\sqrt{\frac{\alpha}{2\epsilon}}x\right)\right\} - \frac{\sqrt{\pi}}{\alpha}\int_{\sqrt{\frac{\alpha}{2\epsilon}}x}^{\sqrt{\frac{\alpha}{2\epsilon}}a}\mathrm{d}u \mathrm{e}^{u^2}\mathrm{erfc}(u), \tag{A.5}$$

when $x < a$, and 0 otherwise, where erfc and erfi are the complementary and imaginary error functions, respectively [78]. It is straightforward to obtain the mean first passage time conditioned on the stationary measure:

$$\mathbb{E}_s[\tau_a] = \int_{-\infty}^{+\infty}\mathrm{d}x P_s(x)\mathbb{E}_x[\tau_a] = \sqrt{\frac{\alpha}{2\pi\epsilon^3}}\int_{-\infty}^{a}\mathrm{d}y \mathrm{e}^{\frac{\alpha y^2}{2\epsilon}}\left(\int_{-\infty}^{y}\mathrm{d}z \mathrm{e}^{-\frac{\alpha z^2}{2\epsilon}}\right)^2. \tag{A.6}$$

The above formula provides the theoretical prediction against which numerical estimates of return times for the Ornstein–Uhlenbeck process are compared in the paper.

Appendix B. Statistical properties of AMS estimators

The standard way of analysing the efficiency of an estimator $\hat{\theta}_N$ (or rather, a family of estimators indexed by a parameter N, e.g. a sample size) of a quantity θ is to consider the mean-square error:

$$\mathrm{MSE}(N) = \mathbb{E}|\hat{\theta}_N - \theta|^2 = \big(\mathbb{E}[\hat{\theta}_N] - \theta\big)^2 + \mathrm{Var}\big(\hat{\theta}_N\big), \tag{B.1}$$

which is decomposed into the contributions of the bias $b(N) = \mathbb{E}[\hat{\theta}_N] - \theta$ (which represents the systematic, or model, error) and of the variance $\mathrm{Var}\big(\hat{\theta}_N\big)$ (which represents the statistical error). For some error tolerance $\epsilon > 0$, the cost of the simulation is the expected cost of one realisation of the algorithm using a parameter N such that $\mathrm{MSE}(N) \leqslant \epsilon^2$: finding optimal N requires a bias-variance trade-off. The precision of the estimation is improved by controlling the bias, and the fluctuations by controlling the variance.

We now consider the AMS estimator $\hat{q}_N(a)$, defined in section 3.4. Note that we index this estimator with the number of initial trajectories N: as we shall see, this is the parameter which controls the statistical properties, and not the total number of sampled trajectories $M = N + \tilde{J}$. One of the main properties of the AMS algorithm is the following unbiasedness result, see [60] for more general statements, and discussion on the influence of the time discretization of the Markov dynamics.

Theorem 1. *For every N, for every score function ξ, $\hat{q}_N$ is an unbiased estimator of q:*

$$\mathbb{E}[\hat{q}_N] = q. \tag{B.2}$$

Thus only the statistical error $\mathrm{Var}(\hat{q}_N)$ depends on the choice of N, and, more importantly, on the score function ξ; see [60, 79] for extensive numerical simulations concerning the role of the score function. In practice, it is recommended in [60] that one computes empirical averages $\overline{q}_{N,K} = \frac{1}{K}\sum_{k=1}^{K} \hat{q}_N \mathrm{e}^{(k)}$ over K independent realisations of the algorithm, with large K: the associated mean-square error is $\mathrm{MSE}(N,K) = \frac{\mathrm{Var}(\hat{q}_N)}{K}$. Moreover, repeating the experience with different choices of score functions is a way to validate the results, checking the overlap of confidence intervals.

In addition, it has been proved, in different contexts, see [80, 81], that $\hat{q}_N$ is a consistent estimator of q: the convergence $\hat{q}_N \underset{N\to\infty}{\to} q$ holds true, in probability. More precisely, it is proved in [81], that the estimator $\hat{q}_N$ satisfies a central limit theorem,

$$\sqrt{N}\big(\hat{q}_N - q\big) \underset{N\to\infty}{\to} \mathcal{N}(0, \sigma^2(\xi, q)), \tag{B.3}$$

with an asymptotic variance $\sigma^2(\xi,q) \in [-q^2 \ln q, 2q(1-q)]$. The minimal variance $-q^2 \ln q$ is obtained when choosing

$$\xi(y) = \overline{\xi}(y) \equiv \mathbb{P}_y(\tau_{\mathcal{B}} < \tau_{\mathcal{A}}). \tag{B.4}$$

In practice, the optimal score function $\overline{\xi}$, also referred to as the *committor*, is of course not known; note that the estimated probability satisfies $q = \xi(y_0)$. Below we will discuss

J. Stat. Mech. (2018) 043213

more precisely the statistical properties of the estimators $\hat{q}_N$ and $\hat{r}_N$ when choosing $\overline{\xi}$ as the score function.

Note that $\sigma^2(\xi, q) \leqslant 2q(1-q) = 2\mathrm{Var}(\mathcal{P})$, where $\mathcal{P}$ is a Bernoulli random variable with mean q. This ensures that in terms of variance the AMS algorithm performs better than or similarly to the crude Monte Carlo method, in the rare event regime $q \to 0$; moreover, the AMS algorithm with optimal score function outperforms the crude Monte-Carlo method (please note that this is the variance normalised by N, where N is the number of initial trajectories, and not by M, where $M = N + \tilde{J}$ is the total number of computed trajectories).

Note that $\mathbb{E}[\hat{r}_N] \neq r$, and thus $\hat{r}_N$ is not an unbiased estimator of r. However, a central limit theorem still holds true: since $\hat{r}_N = \phi(\hat{q}_N)$ and $r = \phi(q)$ for some function ϕ, such that $\phi'(q) \neq 0$, the δ-method [82] implies

$$\sqrt{N}\big(\hat{r}_N - r\big) \underset{N\to\infty}{\to} \mathcal{N}\big(0, \sigma^2(\xi, q)(\phi'(q))^2\big), \tag{B.5}$$

where $q^2\phi'(q) \underset{q\to 0}{\to} T_a$, with T_a the size of the window. The estimators for the return time r correspond to the choices $\phi(q) = -T_a/\ln(1-q)$ or $\phi(q) = T_a/q$.

For an arbitrary choice of the score function ξ, it is not possible in general to obtain precise results concerning the bias for the return time $\hat{r}_N$, and the asymptotic variance. However, when the optimal score function $\overline{\xi}(y) = \mathbb{P}_y(\tau_{\mathcal{B}} < \tau_{\mathcal{A}})$ is used, elementary arguments are sufficient to analyse the statistical properties of estimators $\hat{q}_N$ and $\hat{r}_N = \frac{1}{\hat{q}_N}$ (with $T_a = 1$). The key property [61, 80, 83], is that, when using the optimal score function, the number of iterations J follows a Poisson distribution, with parameter $-N\ln q$. This situation is referred to as the idealised case in the mathematical literature. Since $\hat{q}_N = \big(1 - \frac{1}{N}\big)^J$, proving the following results is straightforward: first, concerning the bias,

$$\mathbb{E}[\hat{q}_N] = q, \qquad \mathbb{E}\left[\frac{1}{\hat{q}_N}\right] - \frac{1}{q} \underset{N\to\infty}{\sim} \frac{-\ln q}{qN}. \tag{B.6}$$

Second, concerning the asymptotic variance,

$$\mathrm{Var}(\hat{q}_N) = q^2\big(q^{-\frac{1}{N}} - 1\big) \underset{N\to\infty}{\sim} \frac{-q^2\ln q}{N}, \qquad \mathrm{Var}\left(\frac{1}{\hat{q}_N}\right) \underset{N\to\infty}{\sim} \frac{-\ln q}{Nq^2}. \tag{B.7}$$

Note that relative bias and variance are both of size $\frac{-\ln q}{N}$. The derivation of the central limit theorem [84], and large deviations results [85] is also straightforward in the idealised case.

References

[1] Kramers H A 1940 *Physica* **7** 284
[2] Kurkijärvi J 1972 *Phys. Rev.* B **6** 832
[3] Dykman M 2012 *Fluctuating Nonlinear Oscillators: from Nanomechanics to Quantum Superconducting Circuits* ed M Dykman (Oxford: Oxford University Press)
[4] Bouchet F and Simonnet E 2009 *Phys. Rev. Lett.* **102** 094504
[5] Berhanu M *et al* 2007 *Europhys. Lett.* **77** 59001
[6] Calef D F and Deutch J M 1983 *Ann. Rev. Phys. Chem.* **34** 493

J. Stat. Mech. (2018) 043213

[7] Noé F, Schütte C and Vanden-Eijnden E 2009 *Proc. Natl Acad. Sci. USA* **106** 19011
[8] Paillard D 1998 *Nature* **391** 378
[9] Robine J M, Cheung S L K, Le Roy S, Van Oyen H, Griffiths C, Michel J P and Herrmann F R 2008 *C. R. Biol.* **331** 171
[10] Dysthe K, Krogstad H E and Muller P 2008 *Ann. Rev. Fluid Mech.* **40** 287
[11] Yeung P K, Zhai X M and Sreenivasan K R 2015 *Proc. Natl Acad. Sci. USA* **112** 12633
[12] Embrechts P, Klüppelberg C and Mikosch T 2013 *Modelling Extremal Events: For Insurance and Finance* (*Stochastic Modelling and Applied Probability* vol 33) (Berlin: Springer)
[13] Ghil M *et al* 2011 *Nonlinear Process. Geophys.* **18** 295
[14] Fortin J Y and Clusel M 2015 *J. Phys. A: Math. Theor.* **48** 1
[15] Lucarini V, Freitas A C M, Faranda D, Freitas J M, Holland M, Kuna T, Nicol M, Todd M and Vaienti S 2016 *Extremes and Recurrence in Dynamical Systems* (New York: Wiley)
[16] Freidlin M I and Wentzell A D 1998 *Random Perturbations of Dynamical Systems* 2nd edn (New York: Springer)
[17] Ellis R S 1985 *Entropy, Large Deviations, and Statistical Mechanics* (New York: Springer)
[18] Den Hollander F 2008 *Large Deviations* vol 14 (Providence, RI: American Mathematical Society)
[19] Touchette H 2009 *Phys. Rep.* **478** 1
[20] Vulpiani A, Cecconi F, Cencini M, Puglisi A and Vergni D 2014 *Large Deviations in Physics, The Legacy of the Law of Large Numbers* (Berlin: Springer)
[21] Asmussen S and Glynn P 2007 *Stochastic Simulation: Algorithms and Analysis* (*Stochastic Modelling and Applied Probability* vol 57) (New York: Springer) p 14, 476
[22] Landau D P and Binder K 2015 *A Guide to Monte Carlo Simulations in Statistical Physics* 4th edn (Cambridge: Cambridge University Press) p 17, 519
[23] Liu J S 2008 *Monte Carlo Strategies in Scientific Computing* (*Springer Series in Statistics*) (New York: Springer) p 16, 343
[24] Bucklew J 2004 *Introduction to Rare Event Simulation* (*Springer Series in Statistics*) (New York: Springer) p 12, 260
[25] Rubino G and Tuffin B 2009 *Rare Event Simulation Using Monte Carlo Methods* (Chichester: Wiley) pp 1–13
[26] Grassberger P 2002 *Comput. Phys. Commun.* **147** 64
[27] Del Moral P and Garnier J 2005 *Ann. Appl. Probab.* **15** 2496
[28] Cérou F and Guyader A 2007 *Stoch. Anal. Appl.* **25** 417
[29] Giardina C, Kurchan J, Lecomte V and Tailleur J 2011 *J. Stat. Phys.* **145** 787
[30] Rolland J, Bouchet F and Simonnet E 2016 *J. Stat. Phys.* **162** 277
[31] Dellago C, Bolhuis P and Geissler P L 2002 *Adv. Chem. Phys.* **53** 291–318
[32] Weinan E, Ren W and Vanden-Eijnden E 2002 *Phys. Rev.* B **66** 052301
[33] Weinan E, Ren W and Vanden-Eijnden E 2004 *Commun. Pure Appl. Math.* **52** 637
[34] Laurie J and Bouchet F 2015 *New J. Phys.* **17** 015009
[35] Grafke T, Grauer R and Schäfer T 2015 *J. Phys. A: Math. Theor.* **48** 1
[36] Grigorio L S, Bouchet F, Pereira R M and Chevillard L 2017 *J. Phys. A: Math. Theor.* **50** 055501
[37] Wouters J and Bouchet F 2016 *J. Phys. A: Math. Theor.* **49** 374002
[38] Leadbetter M R 1983 *Extremes and Related Properties of Random Sequences and Processes* (*Springer Series in Statistics*) (New York: Springer)
[39] Doucet A, De Freitas N and Gordon N 2001 *Sequential Monte Carlo Methods in Practice* (New York: Springer)
[40] Sveinsson O, Salas J D and Boes C D 2002 *J. Hydrol. Eng.* **7** 49
[41] Gumbel E J 1941 *Ann. Math. Stat.* **12** 163
[42] Corral A 2005 *Phys. Rev.* E **71** 017101
[43] Peres D J and Cancelliere A 2016 *J. Hydrol.* **541** 256
[44] Meehl G A and Tebaldi C 2004 *Science* **305** 994
[45] Rahmstorf S and Coumou D 2011 *Proc. Natl Acad. Sci. USA* **108** 17905
[46] Cattiaux J, Vautard R, Cassou C, Yiou P, Masson-Delmotte V and Codron F 2010 *Geophys. Res. Lett.* **37** L20704
[47] Shepherd T G 2016 *Curr. Clim. Change Rep.* **2** 28
[48] Godrèche C, Majumdar S N and Schehr G 2017 *J. Phys. A: Math. Theor.* **50** 333001
[49] Redner S 2001 *A Guide to First-Passage Processes* (Cambridge: Cambridge University Press)
[50] Bray A J, Majumdar S N and Schehr G 2013 *Adv. Phys.* **62** 225
[51] Nicolis C and Nicolis S 2007 *Europhys. Lett.* **80** 40003
[52] Langer J S 1969 *Ann. Phys.* **54** 258

J. Stat. Mech. (2018) 043213

[53] Gardiner C W 2009 *Handbook of Stochastic Methods for Physics, Chemistry, and the Natural Sciences* 4th edn (Berlin: Springer)
[54] Risken H 1989 *The Fokker–Planck Equation* 2nd edn (Berlin: Springer)
[55] Bouchet F and Reygner J 2016 *Ann. Henri Poincaré* 17 3499
[56] Cérou F, Guyader A, Lelièvre T and Pommier D 2011 *J. Chem. Phys.* **134** 054108
[57] Giardina C, Kurchan J and Peliti L 2006 *Phys. Rev. Lett.* **96** 120603
[58] Otto F, Massey N, van Oldenborgh G, Jones R and Allen M 2012 *Geophys. Res. Lett.* **39** L04702
[59] Kahn H and Harris T E 1951 *Natl Bur. Stand.* **12** 27
[60] Bréhier C E, Gazeau M, Goudenège L, Lelièvre T and Rousset M 2016 *Ann. Appl. Probab.* **26** 3559
[61] Simonnet E 2016 *Stat. Comput.* **26** 211
[62] Tailleur J and Kurchan J 2007 *Nat. Phys.* **3** 203
[63] Lecomte V and Tailleur J 2007 *J. Stat. Mech.* P03004
[64] Garrahan J P, Jack R L, Lecomte V, Pitard E, van Duijvendijk K and van Wijland F 2007 *Phys. Rev. Lett.* **98** 195702
[65] Hurtado P I and Garrido P L 2009 *J. Stat. Mech.* P02032
[66] Berg B A and Neuhaus T 1992 *Phys. Rev. Lett.* **68** 9
[67] Hartmann A K 2002 *Phys. Rev.* E **65** 056102
[68] Moral P D 2004 *Feynman–Kac Formulae: Genealogical and Interacting Particle Systems with Applications* (New York: Springer)
[69] Del Moral P 2013 *Mean Field Simulation for Monte Carlo Integration* (*Monographs on Statistics and Applied Probability* vol 126) (Boca Raton, FL: CRC Press) p xlvii+578
[70] Spalart P R 2000 *Int. J. Heat Fluid Flow* **21** 252
[71] Moin P 2002 *Int. J. Heat Fluid Flow* **23** 710
[72] Chen S and Doolen G D 1998 *Ann. Rev. Fluid Mech.* **30** 329
[73] Ragone F, Wouters J and Bouchet F 2017 *Proc. Natl Acad. Sci. USA* 201712645
[74] Nemoto T, Bouchet F, Jack R L and Lecomte V 2016 *Phys. Rev.* E **93** 062123
[75] Vanden-Eijnden E and Weare J 2012 *Commun. Pure Appl. Math.* **65** 1770
[76] Faradjian A K and Elber R 2004 *J. Chem. Phys.* **120** 10880
[77] Schütte C, Noé F, Lu J, Sarich M and Vanden-Eijnden E 2011 *J. Chem. Phys.* **134** 05B609
[78] Abramowitz M and Stegun I 1965 *Handbook of Mathematical Functions* (New York: Dover)
[79] Rolland J and Simonnet E 2015 *J. Comput. Phys.* **283** 541
[80] Bréhier C E, Lelièvre T and Rousset M 2015 *ESAIM Probab. Stat.* **19** 361
[81] Cerou F, Delyon B, Guyader A and Rousset M 2016 A central limit theorem for Fleming–Viot particle systems with soft killing (arXiv:1611.00515)
[82] van der Vaart A W 1998 *Asymptotic Statistics* (*Cambridge Series in Statistical and Probabilistic Mathematics* vol 3) (Cambridge: Cambridge University Press) p 16, 443
[83] Guyader A, Hengartner N and Matzner-Løber E 2011 *Appl. Math. Optim.* **64** 171
[84] Bréhier C E, Goudenège L and Tudela L 2016 *Monte Carlo and Quasi-Monte Carlo Methods* (*Springer Proceedings in Mathematics and Statistics* vol 163) (Berlin: Springer) pp 245–60
[85] Bréhier C E 2015 *ALEA Latin Am. J. Probab. Math. Stat.* **12** 717