Adaptive Sampling of Large Deviations
Grégoire Ferré, Hugo Touchette

To cite this version:
Grégoire Ferré, Hugo Touchette. Adaptive Sampling of Large Deviations. Journal of Statistical Physics, Springer Verlag, 2018, 172 (6), pp.1525 - 1544. <10.1007/s10955-018-2108-8>. <hal-01938825>

HAL Id: hal-01938825
https://hal.archives-ouvertes.fr/hal-01938825
Submitted on 29 Nov 2018

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

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

Grégoire Ferré
CERMICS, École des Ponts ParisTech, Université Paris-Est, F-77455 Marne-la-Vallée, France

Hugo Touchette
National Institute for Theoretical Physics (NITheP), Stellenbosch 7600, South Africa and Institute of Theoretical Physics and Department of Physics, University of Stellenbosch, Stellenbosch 7600, South Africa

(Dated: July 4, 2018)

We introduce and test an algorithm that adaptively estimates large deviation functions characterizing the fluctuations of additive functionals of Markov processes in the long-time limit. These functions play an important role for predicting the probability and pathways of rare events in stochastic processes, as well as for understanding the physics of nonequilibrium systems driven in steady states by external forces and reservoirs. The algorithm uses methods from risk-sensitive and feedback control to estimate from a single trajectory a new process, called the driven process, known to be efficient for importance sampling. Its advantages compared to other simulation techniques, such as splitting or cloning, are discussed and illustrated with simple equilibrium and nonequilibrium diffusion models.

I. INTRODUCTION

We consider in this paper the problem of estimating large deviation functions characterizing the fluctuations of time-integrated functionals of Markov processes in the long-time limit. These functions have a wide range of applications in engineering and physical sciences, where they are used to predict the probability of rare events [1–3] and to understand how these events arise via transition paths or modified processes [4–6]. Large deviation theory also underlies now much of the research on nonequilibrium systems driven in steady states by non-conservative forces or boundary reservoirs [7–9]. In this context, large deviation functions play the role of nonequilibrium potentials, similar to the entropy or free energy, that characterize the steady state and fluctuations of physical quantities, such as energy or particle currents exchanged with reservoirs, as well as the presence of symmetries [10] and phase transitions in fluctuations [11–14].

Recently, many efforts have been devoted to developing numerical algorithms for estimating large deviation functions that go beyond the direct sampling of probabilities, which require prohibitively large samples. The most popular algorithms are variations of two basic approaches used in rare event simulations, namely: 1) splitting [15–17] or cloning [18–20] algorithms, which use population dynamics to estimate probabilities or generating functions that have a multiplicative structure in time, and 2) importance sampling [21–23] (including transition path sampling [24]), which modifies the process to be simulated, so as to transform rare events into typical events that can be simulated efficiently. Deterministic methods not based a priori on sampling can also be used, including spatial discretizations of various spectral and optimal control representations of large deviation functions, which work well for low-dimensional systems, in addition to action minimization methods, which can be applied in the low-noise or low-temperature limit [25–27].

In this paper, we propose an algorithm that combines spectral methods with importance sampling to efficiently estimate large deviation functions in an adaptive way. The core of the algorithm comes from recent works on learning algorithms for risk-sensitive control of Markov chains [28–31], which we adapt to continuous-time diffusion processes and to the problem of estimating large deviation functions. The algorithm works by estimating or learning “on the fly” a modified process, called the auxiliary or driven process, which corresponds to the process that is asymptotically
equivalent to the original process conditioned on the rare event of interest [6] or, alternatively, to the exponential tilting of the original process, known to be efficient for importance sampling [32]. This modified process is given by a principal eigenvalue problem related to the Feynman–Kac equation or, equivalently, by a stochastic optimal control problem [32] that we solve iteratively using stochastic approximation and feedback control methods.

The main advantage of this algorithm, compared to splitting or cloning, is that it does not require the simulation of many copies of the process considered – it runs on one long trajectory of that process, modified with a feedback-reinforcement rule, to adaptively learn the driven process, thereby reducing significantly the complexity of estimating large deviation functions. The calculation of error bars for the estimated quantities is also simplified compared to other techniques, as the algorithm is based on simple time averages and stochastic approximations [33–35]. Finally, the errors incurred by discretizing continuous-time processes and functionals can be analysed in a precise way, in principle, via Feynman–Kac semi-groups [36].

We discuss these advantages and test the algorithm in Sec. IV with simple equilibrium and nonequilibrium diffusions, after introducing the general model and notations in Sec. II and the algorithm in Sec. III. The results at this point are preliminary and are presented as a proof of concept of the algorithm. More detailed results about the time-discretization and sampling errors will be addressed in future works, together with more complex applications involving interacting particle systems and higher-dimensional diffusions.

II. FRAMEWORK

A. Model and notations

We consider an ergodic diffusion \((X_t)_{t \geq 0}\) evolving in a state space \(\mathcal{X} \subset \mathbb{R}^d\) according to the following stochastic differential equation (SDE):

\[
dX_t = b(X_t) \, dt + \sigma \, dW_t, \tag{1}
\]

where \(b : \mathcal{X} \to \mathbb{R}^d\) is a smooth function, called the drift, \(W_t\) is an \(m\)-dimensional Brownian motion, and \(\sigma\) is a \(d \times m\) matrix, assumed to be constant for simplicity.\(^1\) The generator of this diffusion reads

\[
L = b \cdot \nabla + \frac{1}{2} \nabla \cdot D \nabla, \tag{2}
\]

where \(\cdot\) denotes the scalar product and \(D = \sigma \sigma^T\), with \(T\) as the transpose, is the diffusion matrix, assumed to be positive definite. This is the generator of the evolution semi-group \(P_T\), defined by

\[
P_T \varphi(x) = \mathbb{E} \left[ \varphi(X_T) \mid X_0 = x \right], \tag{3}
\]

for all time \(T \geq 0\) and any smooth test function \(\varphi\). The dual \(L^\dagger\) of \(L\) in the space \(L^2(dx)\) of square-integrable functions with respect to the Lebesgue measure is the generator of the Fokker-Planck equation

\[
\partial_t \rho(x,t) = L^\dagger \rho(x,t), \tag{4}
\]

which gives the evolution of the probability density \(\rho(\cdot,t)\) of \(X_t\) starting from some initial density \(\rho(\cdot,0)\) for \(X_0\).

\(^1\) See [6] for a treatment of diffusions with multiplicative noise.
Our goal here is to study the fluctuations of time-integrated functionals of $X_t$, called observables, having the general form

$$A_T = \frac{1}{T} \int_0^T f(X_t) \, dt + \frac{1}{T} \int_0^T g(X_t) \circ dX_t,$$

where $f : \mathcal{X} \to \mathbb{R}$ and $g : \mathcal{X} \to \mathbb{R}^d$ are reasonably smooth functions (e.g. continuous) and $\circ$ denotes the Stratonovich product [37]. Such a functional defined over the time horizon $[0, T]$ can represent, for example, a control cost associated with the state $X_t$ and its increments [38] or a physical quantity integrated in time, such as the work performed on a particle by external forces or the heat exchanged by a particle with its environment [39].

Assuming that the process is ergodic with respect to an invariant measure $\mu(dx) = \rho^* \ast (x) dx$ with smooth density $\rho^*$, we have

$$A_T \xrightarrow{T \to \infty} \int_{\mathcal{X}} f(x) \rho^*(x) \, dx + \int_{\mathcal{X}} g(x) \cdot J^*(x) \, dx = a^*,$$

almost surely, where

$$J^*(x) = b(x) \rho^*(x) - \frac{D}{2} \nabla \rho^*(x)$$

is the stationary current field associated with $\rho^*$ [32]. The theory of large deviations [2] refines this ergodic theorem, generalized here with the additional $g$ term, by providing estimates for the rate at which the probability distribution of $A_T$ concentrates on its ergodic value $a^*$. Such estimates can be derived under general conditions (see [2]) and take, in the simplest case, the form

$$\lim_{T \to \infty} -\frac{1}{T} \log \mathbb{P}(A_T \in B) = \min_{a \in B} I(a),$$

for any Borel subset $B$ of $\mathbb{R}$, where $I : \mathbb{R} \to \mathbb{R}_+$ is a positive function such that $I(a^*) = 0$. When this limit exists, $A_T$ is said to satisfy the large deviation principle (LDP) with rate function $I$. Formally, this means that

$$\mathbb{P}(A_T \in da) = e^{-TI(a) + o(T)} \, da,$$

where $o(T)$ denotes corrections in the exponential that grow slower than linearly in $T$. Thus, we see that the rate function provides useful information about the fluctuations of $A_T$: the likelihood that $A_T = a$ decays exponentially with time for all $a \neq a^*$, since $I(a) > 0$ in this case, and converges otherwise to 1 as $T \to \infty$, since $I(a^*) = 0$. Moreover, the rate function is in general not a parabola, meaning that it describes fluctuations that are generally not Gaussian. In this sense, large deviation theory is often seen as an extension of both the ergodic theorem, which describes the concentration of $A_T$ towards its mean, and the central limit theorem, which describes the local Gaussian fluctuations of $A_T$ around its mean [2].

### B. Large deviation functions and driven process

In practice, the rate function $I(a)$ can be calculated in many different ways other than by direct sampling, which requires exponentially large samples with $T$ [21]. The most common method proceeds from the scaled cumulant generating function (SCGF), defined for $k \in \mathbb{R}$ by

$$\lambda(k) = \lim_{T \to \infty} \frac{1}{T} \log \mathbb{E} \left[ e^{kT A_T} \right].$$
By the Gärtner-Ellis Theorem [2], the Legendre–Fenchel transform of this function yields the rate function:

$$I(a) = \sup_{k \in \mathbb{R}} \{ ka - \lambda(k) \},$$

(11)

provided, essentially, that $\lambda(k)$ is differentiable; see [2] for more precise conditions.

For the SDE (1) and the additive functional (5), the SCGF is known to be given by the principal eigenvalue of the operator

$$\mathcal{L}_k = b \cdot (\nabla + kg) + \frac{1}{2} (\nabla + kg) \cdot D(\nabla + kg) + kf,$$

(12)

which is the generator of the Feynman–Kac semi-group $P^k_T$, defined by

$$P^k_T \varphi(x) = \mathbb{E} \left[ \varphi(X_T) e^{TkA_T} \bigg| X_0 = x \right],$$

(13)

for any smooth function $\varphi$; see Appendix A.2 of [6]. In the end, the rate function can thus be computed by solving the spectral problem

$$\mathcal{L}_k h_k = \lambda(k) h_k,$$

(14)

where $\lambda(k)$ is the principal eigenvalue of $\mathcal{L}_k$ and $h_k$ its corresponding eigenvector. This holds provided that this operator has reasonable spectral properties, made precise in the following assumption.

**Assumption 1.** The operator $\mathcal{L}_k$ defined in (12) acting on $L^2(\mu)$ has an isolated largest eigenvalue $\lambda(k)$. Its multiplicity is one and it is associated with a regular eigenvector $h_k \in L^2(\mu)$ such that for all $x \in \mathcal{X}$, $h_k(x) > 0$.

This assumption holds for many systems, in particular when $\mathcal{X}$ is bounded [36, 40] or when $b$ and $g$ are gradient fields with appropriate growth conditions; see [41, Sec. 2.5]. In practice, the spectral problem (14) can be solved numerically using standard projection or discretization (Galerkin) methods, which work well for low-dimensional systems [42], or more involved real space renormalization methods when dealing with higher-dimensional systems [43]. Note that, for $g = 0$, $\mathcal{L}_k$ is the usual Feynman–Kac generator with source term $kf$. Moreover, for $k = 0$, $\mathcal{L}_0 = L$ is the generator of the SDE, so that $\lambda(0) = 0$ and $h_0 = 1$, the constant unit function.

The numerical method that we propose in the next section attempts to estimate the spectral elements $\lambda(k)$ and $h_k$ in a different way using the fact that they are solutions of the family of eigenproblems

$$P^k_T h_k = e^{Tk\lambda(k)} h_k, \quad \forall T > 0,$$

(15)

which can be approximated stochastically. The method also exploits a connection between large deviations and control theory showing that $\lambda(k)$ is the ergodic limit of an optimal control cost satisfying a stochastic Hamilton–Jacobi–Bellman equation [44], and that $h_k$ determines the controlled diffusion achieving the optimal cost, which is the driven process mentioned in the introduction. These results are explained in detail in [32] (see also references therein); for the purpose of this paper, we only state them without proofs.

The controlled diffusion, denoted by $(\tilde{X}_t)_{t \geq 0}$, satisfies the SDE

$$d\tilde{X}_t = b_k(\tilde{X}_t) dt + \sigma dW_t,$$

(16)
where
\[ b_k(x) = b(x) + D[kg(x) + \nabla \log h_k(x)] \]  
(17)
is the optimal control drift defined for all \( x \in \mathcal{X} \). Under Assumption 1, this modified diffusion is ergodic with respect to a new invariant measure \( \mu_k \), whose density is
\[ \rho_k(x) = h_k(x)l_k(x), \]  
(18)
where \( l_k \) is the dual of \( h_k \) in \( L^2(dx) \) satisfying \( L^\dagger_k l_k = \lambda(k)l_k \) [6, Sec. 3.3]. These two functions are normalized such that
\[ \int_{\mathcal{X}} h_k(x)l_k(x) \, dx = 1, \quad \int_{\mathcal{X}} l_k(x) \, dx = 1. \]  
(19)

Because of the change of process, the observable \( A_T \) must concentrate on a new value, which can be shown to be given by \( \lambda'(k) \), that is,
\[ A_T \xrightarrow{T \to \infty} \lambda'(k) = a_k, \]  
(20)
amost surely with respect to \( (\tilde{X}_t)_{t \geq 0} \). Similarly, the control cost
\[ C_T = kA_T - R_T, \]  
(21)
where
\[ R_T = \frac{1}{2T} \int_0^T [b(\tilde{X}_t) - b_k(\tilde{X}_t)] : D^{-1}[b(\tilde{X}_t) - b_k(\tilde{X}_t)] \, dt, \]  
(22)
reaches in the ergodic limit the value \( \lambda(k) \), so that
\[ C_T \xrightarrow{T \to \infty} \lambda(k) \]  
(23)
amost surely under \( (\tilde{X}_t)_{t \geq 0} \) [32, Sec. 4]. Finally, it can be shown by Legendre duality that the rate function at the value \( a_k = \lambda'(k) \) is given by the ergodic limit of \( R_T \) above, leading to
\[ R_T \xrightarrow{T \to \infty} I(a_k), \]  
(24)
amost surely with respect to \( (\tilde{X}_t)_{t \geq 0} \) [32, Sec. 4]. Note that the diffusion is not modified for \( k = 0 \), so that \( a_0 = a^* \), \( \lambda'(0) = a^* \), and \( I(a^*) = 0 \).

These ergodic limits provide direct estimators of the SCGF and the rate function, based on a single trajectory of the driven process, which can be simulated for different values of the parameter \( k \in \mathbb{R} \). For the SCGF, there are in fact three possible estimators:

1. From (20): The value of \( A_T \), integrated numerically with the condition \( \lambda(0) = 0 \).
2. From (23): The value of \( C_T \).
3. The eigenvalue returned by the algorithm proposed in Sec. III.

In practice, we find that the last estimator is more stable, although the first and second are more adapted to obtain error bars.

For the rate function, we have two possible estimators:
1. The Legendre transform (11) of the SCGF, given in parametric form by

\[ I(a_k) = k a_k - \lambda(k), \]  

where \( a_k \) is either estimated from \( A_T \) or by taking the numerical derivative of \( \lambda(k) \).

2. From (24): The value of \( R_T \) obtained for the value of \( A_T \), giving the couple \((A_T, R_T)\).

In practice, we find that the first estimator based on the Legendre transform is more reliable. By comparison, the computation of \( I(a) \) based on (22) involves the optimal drift \( b_k \) and, therefore, the logarithmic derivative of \( h_k \), which is more difficult to estimate in a stable way.

In all cases, error bars can be constructed from the same trajectory by estimating, in principle, the variance of \( A_T, C_T \) and \( R_T \) using covariance techniques for Markov processes [33–35]. This is an advantage over splitting and cloning algorithms, for which the calculation of errors bars is difficult, as they involve correlated copies or “clones” of the simulated process [45–47].

In closing, it is interesting to note that the driven process can also be interpreted as the change of process in the importance sampling of the probability \( P(A_T \in B) \) that is optimal in the sense of logarithmic or asymptotic efficiency [21]. Therefore, it can be used not only to estimate the SCGF and rate function, but also to estimate the actual probability \( P(A_T \in B) \) in an efficient way [28].

The optimal change of process in this case is known to correspond to the exponential tilting of the original process [21], which is a time-dependent process in general; see Appendix D of [6]. In the ergodic limit, this process converges to a homogeneous process corresponding exactly to the driven process (16). We refer to [32] for more details about these results.

### III. ADAPTIVE ALGORITHM

We are now ready to present the algorithm for estimating the SCGF and the rate function of \( A_T \) for continuous-time diffusions. The algorithm is based, as mentioned, on prior algorithms proposed in [28–31] for Markov chains and exploits the fact that

\[ e^{-T \lambda(k)} (P_T^k \varphi) \xrightarrow{T \to \infty} h_k \int_X \varphi(x) l_k(x) dx, \]  

for any smooth test function \( \varphi \). This result follows under Assumption 1; see [6, Sec. III.B] or [36, Sec. 6.1] for a proof. The algorithm that we propose works from this limit by approximating the action of the Feynman–Kac semi-group \( P_T^k \) in a stochastic way as a time average computed over a long trajectory of the diffusion. Moreover, it continuously modifies the diffusion as we estimate \( h_k \) to construct the driven process \((\tilde{X}_t)_{t \geq 0}\), which underlies the estimators of the large deviation functions.

The algorithm is presented next. Our contribution compared to [28–31] is to consider general time-continuous processes, time-additive functionals of these processes that depend on both their state and increments, and, more importantly, to construct the driven process explicitly so as to estimate the SCGF and the rate function adaptively using the estimators introduced in the previous section.

#### A. Time discretization

The first step required in the algorithm is to discretize in time the SDE (1) and its associated Feynman–Kac semi-group (13) by transforming the Markov diffusion \((X_t)_{t \geq 0}\) into a Markov chain \((x_n)_{n \in N}\) with a small time step. Many discretization schemes can be used for the SDE; see, for
instance, [48]. Here we use the standard Euler–Maruyama scheme with constant time step $\Delta t$, given by
\[ x_{n+1} = x_n + b(x_n)\Delta t + \sigma \sqrt{\Delta t} \xi_n, \]
where $(\xi_n)_{n \geq 0}$ is a sequence of independent standard $d$-dimensional Gaussian random variables. The corresponding discretization of the evolution semi-group $P_T$ over a time step $\Delta t$ is denoted by $Q_{\Delta t}$, so that
\[ Q_{\Delta t} \varphi(x) = \mathbb{E}[\varphi(x_{n+1})|x_n = x], \]
for any test function $\varphi$ and $x \in \mathcal{X}$. We refer to [41, 48] for more information about the discretization of SDEs and their weak error analysis.

Many discretizations also exist for the Feynman–Kac semi-group $P_k^T$. Here, we use the natural scheme where the diffusion is discretized as above and the integral of $A_T$ is discretized as a Riemann sum with the left-point rule for the integral involving $f$ and the mid-point rule for the Stratonovich integral involving $g$ [36]. The action of $P_k^T$ is thus replaced by
\[ Q_{\Delta t}^k \varphi(x) = \mathbb{E} \left[ e^{k [f(x_n)\Delta t + g(\frac{x_{n+1} + x_n}{2}) - x_{n+1} - x_n]} \varphi(x_{n+1}) \bigg| x_n = x \right]. \]
For our purposes, $h_k$ will be approximated by recursive applications of $Q_{\Delta t}^k$, based on the following assumption.

**Assumption 2.** There exist a time step $\Delta t^* > 0$ and $p > 0$ such that, for $0 < \Delta t \leq \Delta t^*$,
\[ Q_{\Delta t}^k h_{k, \Delta t} = e^{\Delta t \lambda_{\Delta t}(k)} h_{k, \Delta t}, \]
where
\[ h_{k, \Delta t} = h_k + O(\Delta t^p), \quad \lambda_{\Delta t}(k) = \lambda(k) + O(\Delta t^p). \]

This assumption means that the time-discretized operator $Q_{\Delta t}^k$ admits $h_k$ as an approximate eigenvector with approximate eigenvalue $\lambda(k)$. This applies, for example, when $\mathcal{X}$ is compact. In this case, precise estimates for the errors in (31) are obtained for $g = 0$ in [36] and can be extended to $g \neq 0$.

In the following, we will drop the subscript $\Delta t$ on $h_{k, \Delta t}$ and $\lambda_{\Delta t}(k)$ to simplify the notations, and will present the algorithm essentially as if $h_k$ were an exact eigenvector of $Q_{\Delta t}^k$ with exact eigenvalue $\lambda(k)$. However, we should keep in mind that this is only approximately true due to the errors in $\Delta t$. We will comment on this in Sec. IV with specific numerical examples.

### B. Stochastic approximation and annealing

The main ingredient of the algorithm is the limit (26) of the Feynman–Kac semi-group, which shows that $h_k$ and $\lambda(k)$ can be computed by successively applying $Q_{\Delta t}^k$ to an initial guess $\varphi$, so as to obtain
\[ (Q_{\Delta t}^k)^n \varphi \sim e^{n\Delta t \lambda(k)} h_k \int_{\mathcal{X}} \varphi(x) l_k(x) \, dx \]
as \( n \to \infty \). To perform this iteration, which is a functional version of the well-known power method for matrices [49], we apply a stochastic approximation [30, 35, 50] whereby the expectation appearing in the action of \( Q_{\Delta t}^k \) is replaced by the iterates of the Markov chain:

\[
Q_{\Delta t}^k \varphi(x_n) = \mathbb{E} \left[ e^{k \left[ f(x_n) \Delta t + g \left( \frac{x_{n+1} + x_n}{2} \right) \cdot (x_{n+1} - x_n) \right]} \varphi(x_{n+1}) \, | \, x_n \right] \approx e^{k \left[ f(x_n) \Delta t + g \left( \frac{x_{n+1} + x_n}{2} \right) \cdot (x_{n+1} - x_n) \right]} \varphi(x_{n+1}),
\]

(33)

where \( x_{n+1} \) is a random variable distributed according to \( Q_{\Delta t}(x_n, \cdot) \). This approximation is known to reproduce the expectation as a statistical average in the ergodic limit \( n \to \infty \) [29, 30, 35].

In our case, we simulate not the Markov chain \( x_n \), but a modified chain, corresponding to the discretization of the driven process (16), which we express as

\[
\tilde{x}_{n+1} = \tilde{x}_n + [b(\tilde{x}_n) + F_n(\tilde{x}_n)] \Delta t + \sigma \sqrt{\Delta t} \xi_n,
\]

(34)

where

\[
F_n = D(kg + \nabla \log h_k^n)
\]

(35)
is the extra biasing force derived, according to (17), from the estimate \( h_k^n \) of \( h_k \) at time \( n \). In this case, the evolution (33) is modified by Girsanov formula [37] to

\[
Q_{\Delta t}^k \varphi(\tilde{x}_n) = \mathbb{E} \left[ e^{k \left[ f(\tilde{x}_n) \Delta t + g \left( \frac{\tilde{x}_{n+1} + \tilde{x}_n}{2} \right) \cdot (\tilde{x}_{n+1} - \tilde{x}_n) \right]} \varphi(\tilde{x}_{n+1}) R_n(\tilde{x}_n, \tilde{x}_{n+1}) \, | \, \tilde{x}_n = x \right] \approx e^{k \left[ f(\tilde{x}_n) \Delta t + g \left( \frac{\tilde{x}_{n+1} + \tilde{x}_n}{2} \right) \cdot (\tilde{x}_{n+1} - \tilde{x}_n) \right]} \varphi(\tilde{x}_{n+1}) R_n(\tilde{x}_n, \tilde{x}_{n+1}),
\]

(36)

where

\[
R_n(\tilde{x}_n, \tilde{x}_{n+1}) = \exp \left( -\frac{1}{2 \sigma^2} F_n^2(\tilde{x}_n) \Delta t - \sqrt{\frac{\Delta t}{\sigma^2}} F_n(\tilde{x}_n) \cdot \xi_n \right)
\]

(37)
is the Radon–Nikodym derivative of the transition kernel of \( x_n \) with respect to that of \( \tilde{x}_n \).

In the end, we also apply an annealing scheme, commonly used in stochastic approximations, which consists in replacing the update rule (32), defined by \( \varphi^0 = \varphi \) and \( \varphi^{n+1} = Q_{\Delta t}^k \varphi^n \), by the scheme

\[
\varphi^{n+1} = \varphi^n + a_n \left( Q_{\Delta t}^k \varphi^n - \varphi^n \right),
\]

(38)

where \( (a_n)_{n \in \mathbb{N}} \) is a decreasing sequence, often called the adaptation or learning sequence, which acts as a smoothing parameter, filtering here the noisy update of the eigenfunction. This sequence is usually chosen in such a way that

\[
\sum_{n \geq 0} a_n = \infty, \quad \sum_{n \geq 0} a_n^2 < \infty,
\]

(39)

with the understanding that \( a_n \) should not be decreased too slowly, so as to limit noise, nor too fast, so as to reach the “correct” fixed point. These conditions can be relaxed under stability assumptions or by an averaging procedure [50]; see [35] for more details.
C. Spatial projection

The iteration just described for approximating $h_k$ can be performed numerically by discretizing the state space $\mathcal{X}$ into small cells (grid discretization). For high-dimensional systems, however, it is more convenient to use a Galerkin-type approximation of the eigenvalue and eigenfunction [30], obtained by projecting the problem onto a set of basis functions $\{\phi_j\}_{j=1}^M$ with $\phi_j : \mathcal{X} \to \mathbb{R}$ [42]. Let us denote by $\mathcal{H}_M = \text{Span}\{\phi_j\}_{j=1}^M$ the space spanned by these functions. Then the $M$-dimensional eigenproblem that we need to solve is

$$Q_{\Delta t}^k h = e^{\Delta t\lambda} h, \quad h \in \mathcal{H}_M,$$

(40)

using, for notational convenience, the same symbols $h$ and $\lambda$ for the exact and the projected spectral elements. We also drop from now on the parameter $k$, which will be implicit.

The eigenfunction $h$ is expressed in that basis as

$$h(x) = \sum_{j=1}^M \alpha_j \phi_j(x) = \alpha^T \phi(x),$$

(41)

where $\alpha = [\alpha_1, \ldots, \alpha_M]^T$ and $\phi(x) = [\phi_1(x), \ldots, \phi_M(x)]^T$. Multiplying (40) by $\phi_i$ for $i \in \{1, \ldots, M\}$ and integrating over any measure $\eta$ on $\mathcal{X}$ yields

$$\sum_{j=1}^M \alpha_j \int_\mathcal{X} \phi_i(Q_{\Delta t}^k \phi_j) \, d\eta = e^{\Delta t \lambda} \sum_{j=1}^M \alpha_j \int_\mathcal{X} \phi_i \phi_j \, d\eta, \quad i \in \{1, \ldots, M\}.$$

(42)

As a result, we see that the vector of coefficients $\alpha \in \mathbb{R}^M$ is the principal solution of the eigenproblem

$$A\alpha = \Lambda B\alpha, \quad \lambda = \frac{1}{\Delta t} \log \Lambda,$$

(43)

where

$$A = \int_\mathcal{X} \phi(Q_{\Delta t}^k \phi^T) \, d\eta, \quad B = \int_\mathcal{X} \phi \phi^T \, d\eta.$$

(44)

Note that the matrix $B$ is invertible as soon as the $\{\phi_j\}_{j=1}^M$ form a linearly independent family.

D. Algorithm

We are now ready to describe all the steps of the algorithm that estimates the principal eigenvalue of $L_k$ and its corresponding eigenfunction. For a fixed $k \in \mathbb{R}$, we initiate the process at a position $x_0 \in \mathcal{X}$ and define a first approximation $h^0$ of $h$ as

$$h^0 = (\alpha^0)^T \phi,$$

(45)

where $\phi$ is the vector of basis functions and $\alpha^0$ is the initial vector of coefficients, chosen such that $h^0$ is constant. At each iteration, we then perform the following steps:

1. Draw a new position $\tilde{x}_{n+1}$ according to the Markov chain (34).

2. Compute the extra bias $F_n$ according to (35), which in the function basis takes the form

$$F_n = D \left( kg + \frac{\sum_{j=1}^M \alpha_j^0 \nabla \phi_j}{\sum_{j=1}^M \alpha_j^0 \phi_j} \right).$$

(46)
3. Compute the Girsanov weight $R_n$ according to (37).

4. Compute the matrices $A_{n+1}$ and $B_{n+1}$ using the formulae

$$A_{n+1} = \frac{1}{n+1} \sum_{m=0}^{n} e^{k \left[ f(\tilde{x}_m) \Delta t + g \left( \frac{\tilde{x}_{m+1} + \tilde{x}_m}{2} \right) (\tilde{x}_{m+1} - \tilde{x}_m) \right]} \phi(\tilde{x}_m) \phi(\tilde{x}_{m+1})^T R_m(\tilde{x}_m, \tilde{x}_{m+1}),$$

$$B_{n+1} = \frac{1}{n+1} \sum_{m=0}^{n} \phi(\tilde{x}_m) \phi(\tilde{x}_m)^T,$$

which follow by projecting (36) and (44), respectively.

5. Update the coefficient vector $\alpha^n$, giving the decomposition of the iterate $h^n$, as

$$\alpha^{n+1} = \alpha^n + a_n \left( B_{n}^{-1} A_n \frac{h^n(\tilde{x}_0)}{h^n(x_0)} - \text{Id} \right) \alpha^n,$$  

(48)

where $\text{Id}$ is the identity matrix of size $M$ [30].

6. Estimate the eigenvalue as

$$\lambda^{n+1} = \frac{n}{n+1} \lambda^n + \frac{1}{(n+1) \Delta t} \log (h^n(\tilde{x}_0)),$$  

(49)

which follows from (43).

Repeating these steps, it can be proved that the iterates $h^n$ and $\lambda^n$ converge to the solution of the spectral problem (40), following the analysis found in [28–30]; see also [35]. Moreover, because $h^n \to h$, the Markov chain $(\tilde{x}_n)_{n \in \mathbb{N}}$ samples in the long run the Euler–Maruyama discretization of the driven process (16).

### E. Remarks

The following are technical remarks worth noting about the algorithm:

1. The matrix $A_n$ can be updated at each step using

$$A_{n+1} = \frac{n}{n+1} A_n + \frac{1}{n+1} e^{k \left[ f(\tilde{x}_n) \Delta t + g \left( \frac{\tilde{x}_{n+1} + \tilde{x}_n}{2} \right) (\tilde{x}_{n+1} - \tilde{x}_n) \right]} \phi(\tilde{x}_n) \phi(\tilde{x}_{n+1})^T R_n,$$  

(50)

instead of the sum shown in (47). Similarly, the matrix $B_n$ can be updated, following [30], using the Shermann–Morrison–Woodbury formula, which leads here to

$$B_{n+1}^{-1} = \frac{n+1}{n} B_n^{-1} - \frac{n+1}{n} B_n^{-1} \phi(\tilde{x}_{n+1}) \phi(\tilde{x}_{n+1})^T B_n^{-1} R_{n+1}^{-1}.$$  

(51)

The advantage of this formula is that the computation required for updating the coefficient $\alpha^n$ scales with the number $M$ of basis vectors as $M^2$, as is common in the power method [49], whereas the typical cost of inverting the matrix $B$ scales as $M^3$. 
2. We normalize the eigenfunction at every iteration by setting $h(\tilde{x}_0) = 1$ at an arbitrary location, taken here to be the initial state $\tilde{x}_0$ [28]. This prevents the norm of the eigenfunction from diverging or decaying to zero, as is common in the power method, and provides an estimate of $\lambda(k)$ through (49). This normalization step can be based on other norms, at the expense of computing integrals.

3. The algorithm can be run with the original (unbiased) process $X_t$, but the estimation of the matrices $A$ and $B$ in this case typically suffers from large statistical errors due to the high variance of the underlying estimator. This is a known problem related to the estimation of Feynman–Kac functionals and exponential integrals in general [51–54]. Biasing the dynamics with the driven process $\tilde{X}_t$ reduces this variance in an optimal way (in the sense of asymptotic or logarithmic efficiency) by forcing the exploration of the process in important regions of the state space where the integrand of the generating function $E[e^{kTA_t}]$ is largest [32].

4. As the dynamics is biased towards the driven process $\tilde{X}_t$, the vector $\alpha_n$ of basis coefficients representing $h_k$ converges towards the solution of the eigenproblem (43), with the matrices $A$ and $B$ computed as ergodic averages under $\tilde{X}_t$, so that $\eta = \mu_k$ in (44). Even for a small number of basis functions, it would be a priori impossible to compute these matrices by numerical quadrature. This shows that the algorithm can be used to obtain good approximations of $h_k$ even for high-dimensional systems, provided that enough basis functions are used to represent the support of $\mu_k$, which is typically concentrated on a subset of $X$.

5. The algorithm is stable despite the fact that it includes the Girsanov reweighting factor, which is exponential in time. The reason for this stability, already noted in [29], is that the Girsanov weight is computed and accumulated incrementally over single time steps.

6. The learning sequence $(a_n)_{n \in \mathbb{N}}$ is chosen here in the following way:

   (a) For $0 \leq n \leq N_1$, we take $a_n = 0$, so there is no adaption at the level of $h$ and $\lambda$, although the matrices $A$ and $B$ are evolved. This “burn-in” period allows for a better initial guess of the various functions estimated through their ergodic averages.

   (b) For $N_1 < n \leq N_1 + N_{\text{iter}} - N_2$, we take $a_n = 1$, that is, the full information of the process is taken into account.

   (c) For $N_1 + N_{\text{iter}} - N_2 < n \leq N_1 + N_{\text{iter}}$, we take $a_n = C/(n - (N_1 + N_{\text{iter}} - N_2))$ with constant $C > 0$, so the process learns less with time, smoothing the noise in the long run. In the following, we choose $C = 1$.

The times $N_1$ and $N_2$ can be fixed or can be chosen dynamically according to some stopping rule.

7. In practice, we can perform independent simulations with different values of $k$ to obtain an interpolation of the SCGF over some range, say, $[k_{\text{min}}, k_{\text{max}}]$, which can then be used to obtain the rate function by Legendre transform. Alternatively, we can do a simulation in which $k$ is slowly increased from $k = 0$ in a “quasi-static” way, so as to adaptively update the biasing force over a range of values for $k$ [55] or to reach some prescribed value of the SCGF for an unknown $k$ [28].
IV. APPLICATIONS

We apply in this section the algorithm to two simple test cases involving one-dimensional diffusions. The first is the Ornstein-Uhlenbeck process, for which the large deviation functions of the area per unit time are known exactly [6], while the second is a driven diffusion on the circle, often used in physics as a model of nonequilibrium systems, including Josephson junctions perturbed by thermal noise and Brownian particles controlled by external forces [56–58]. We discuss for both the convergence and efficiency of the algorithm.

A. Ornstein-Uhlenbeck process

The first example that we consider is the mean area or mean position

\[ A_T = \frac{1}{T} \int_0^T X_t \, dt \]  

(52)

of the Ornstein-Uhlenbeck process on \( \mathbb{R} \) satisfying the SDE

\[ dX_t = -2\theta X_t \, dt + \sqrt{2} \, dW_t, \]  

(53)

where \( \theta > 0 \). In the notations of Sec. II, we thus have \( b(x) = -2\theta x \), \( \sigma = \sqrt{2} \), \( f(x) = x \) and \( g(x) = 0 \), so that

\[ L = -2\theta x \frac{d}{dx} + \frac{d^2}{dx^2}, \]  

(54)

and \( L_k = L + kx \).

For this process and observable, it can be checked [6] that the SCGF, corresponding to the dominant eigenvalue of \( L_k \), is

\[ \lambda(k) = \frac{k^2}{4\theta^2}, \]  

(55)

so that \( I(a) = \theta^2 a^2 \) from (25). This is expected, since the integral of a Gaussian process is also Gaussian. Moreover, the associated dominant eigenfunction is

\[ h_k(x) = e^{\frac{k}{\theta}x}, \]  

(56)

leading with (17) to the optimal drift

\[ b_k(x) = -2\theta x + 2(\log h_k(x))' = -2\theta x + \frac{k}{\theta}. \]  

(57)

This shows that a fluctuation of \( A_T \) is created in an optimal way by adding a constant to the drift, which “moves” the Gaussian stationary density of the Ornstein-Uhlenbeck process

\[ \rho^*(x) = \sqrt{\frac{\theta}{\pi}} e^{-\theta x^2} \]  

(58)

to

\[ \rho_k(x) = \sqrt{\frac{\theta}{\pi}} e^{-\theta (x-m_k)^2}, \quad m_k = \frac{k}{2\theta^2}, \]  

(59)
leading to

\[ A_T \xrightarrow{T \to \infty} m_k = a_k \]  

almost surely with respect to \( \tilde{X}_t \), in agreement with (20).

We compare the algorithm against these exact results using a simple mesh discretization (first-order finite elements), defined by the basis functions

\[
\phi_j(x) = \begin{cases} 
\frac{x-x_j}{\delta} + 1, & x \in [x_j - \delta, x_j], \\
\frac{x-x_j}{\delta} + 1, & x \in [x_j, x_j + \delta], \\
0, & \text{otherwise},
\end{cases}
\]

where the points \( x_j \) define the centers of each “cell” of width \( \delta > 0 \). In the following, we refer to this basis simply as the linear basis. To illustrate the flexibility of the algorithm, we also perform simulations using a Hermite polynomial basis, which forms a complete orthonormal basis in \( L^2(\rho^*) \).

We run the algorithm with \( k = 1, \theta = 1/2, x_0 = 0, T = 2 \times 10^4, T_1 = N_1 \Delta t = T_2 = N_2 \Delta t = 2 \times 10^3 \) with \( \Delta t = 5 \times 10^{-3} \), which a standard time step used in simulations relative to the basic timescale of the dynamics, corresponding here to \( 1/(2\theta) = 1 \). For the linear basis, we use \( M = 61 \) equally spaced cells with \( \delta = 0.25 \) around \( x = 0 \), whereas for the Hermite basis we use only \( M = 10 \) basis functions.

Figure 1 illustrates the results of a typical simulation, starting on the left with the evolution of the estimate of the SCGF, given by (49), as time increases. We observe a very good agreement in the long run with the exact value, which for the parameters used is equal to 1, with a faster convergence observed for the Hermite basis compared to the linear basis. As mentioned before, we can also recover the SCGF by recording the stationary value of \( A_T \), which corresponds as above to \( m_k = a_k = \lambda'(k) \), and numerically integrate the result in \( k \) from \( \lambda(0) = 0 \). The results obtained are similar to those obtained from the eigenvalue estimate (49), and are not shown for this reason.

To understand the convergence of the eigenvalue at the process level, we show in the right plot of Fig. 1 the evolution of the effective potential \( V_k(x) \) associated with the modified drift \( b_k(x) \) according to

\[ b_k(x) = -V_k'(x). \]
We show the results for both the linear basis and the Hermite basis, with the zero of the potential arbitrarily set at $V_k(0) = 0$. We see from these that the tail of the potential takes longer to be estimated correctly, as the process starts to explore values away from $m_0 = 0$. After the convergence time, there is a good agreement with the parabola

$$V_k(x) = \theta(x - m_k)^2,$$

which is the exact solution predicted by (59). The small errors are due to the finite time step $\Delta t$ used and the finite basis function set. The small “wiggles” seen in the potential obtained with the linear basis come from the fact that this basis is piecewise linear, so that $V_k(x)$ is piecewise logarithmic. The potential obtained with the Hermite basis is smoother, as expected.

Repeating the simulations for other values of $k$, we can recover $\lambda(k)$, as shown in the left plot of Fig. 2. The result is in good agreement with the exact solution (55). Estimating $\lambda'(k)$ with (20) leads to an estimated rate function, by the Legendre transform (25), which also agrees well with the exact rate function, shown in the right plot of Fig. 2. By comparison, the rate function obtained from the time average (24) of the extra biasing force (with the Hermite basis) is not as good: it lies above the exact rate function and shows a larger offset or error as $k$ increases, which does not decrease by reducing $\Delta t$ or increasing the number of basis functions.

To understand this error, we show in the left plot of Fig. 3 the evolution of the extra biasing force $F_n$ for $k = 1$, estimated at the current location of the system, which should approach the constant 2, following (57). The evolution is noisy, as can be seen, which is expected, since $F_n$ is estimated by the logarithmic derivative

$$F_n(x) = 2 \frac{(h^n(x))'}{h^n(x)},$$

computed in the function basis from (46). The derivative amplifies the Monte Carlo errors inherent in the estimate $h^n$. In addition, the denominator often takes small values away from the mean position $m_n$, which makes the estimation of the optimal force still more difficult. The right plot of Fig. 3 shows that the noise on $F_n$ is considerably filtered out by the time average underlying the estimator $R_T$ of the rate function, although a bias remains even after the convergence time of the SCGF, which leads to the offset seen in Fig. 2. The results in both cases are more noisy for the linear basis because the derivative is not continuous across the different cells.
FIG. 3. Large deviations of the mean area of the Ornstein-Uhlenbeck process. Left: Biasing force \( F_n \) at the current state \( \tilde{x}_n \) of the process at time \( n \). Right: Evolution of the time average \( R_T \), which gives the rate function according to \( (24) \).

The offset on \( R_T \) remains more or less constant by running independent simulations (error bars on 30 simulations are too small to show), and so appears to be a systematic error or bias. Many factors can account for this bias. First, the rate function estimated from the limit shown in \( (24) \) is known to be an upper bound on the true rate function \( (32) \), which is tight if and only if the modified drift estimated in the simulation is the optimal control drift \( (17) \). Here, \( F_n \) is not constant, as predicted from \( (57) \), so we expect the estimate \( R_T \) to lie above its expected value. Second, we have noticed in simulations that \( A_T \) underestimates \( a_k \) for large \( k \), which has the effect of further “pushing” the estimate of the rate function above \( I(a) \). This is most likely due again to \( F_n \) being non-constant. Finally, the time average of \( F_n \) involves the ratio of two functions, according to \( (64) \), which fluctuate in time via the updating of \( \alpha_n \) in \( (46) \). As a result, we expect this additional randomness to artificially increase the second moment of \( F_n \), leading to a further bias.

It is difficult to isolate these factors, and all, in fact, seem to play a role. In future works, it would be interesting to study the bias observed in \( R_T \) by computing, for example, the time average of \( F_n^2 \) using the final estimate of \( h_k \) rather than the time-evolved estimate \( h_n \). Different annealing sequences or averaging techniques could also be used to “filter” the time average of \( F_n^2 \) and mitigate the bias on \( R_T \). For now, the most efficient and reliable way to compute the rate function is to use the estimator based on the SCGF and its Legendre transform.

B. Periodic diffusion

We consider for the second test a diffusion on the unit circle satisfying the SDE

\[
dX_t = (-V'(X_t) + \gamma)dt + \sqrt{2}dW_t,
\]

with the periodic potential \( V(x) = \cos(2\pi x) \) and \( \gamma \in \mathbb{R} \) a constant drive. For \( \gamma \neq 0 \), the total drift cannot be expressed as the gradient of a smooth periodic function, so \( X_t \) is a nonequilibrium process violating detailed balance \( (56) \). The observable studied for this process is the winding number

\[
A_T = \frac{1}{T} \int_0^T dX_t,
\]
FIG. 4. Current large deviations of the driven periodic diffusion. Left: Evolution of the estimated SCGF with time for different time steps $\Delta t$. Right: Estimated eigenfunction $h_k$ for different time steps $\Delta t$ compared with the reference Fourier solution.

calculated with the real rather than periodic state, which can be interpreted physically as the mean velocity or current of a Brownian particle moving around the circle [59]. In the notation of Sec. II, we have $b(x) = -V'(x) + \gamma$, $\sigma = \sqrt{2}$, $f(x) = 0$, and $g(x) = 1$.

The SCGF and the rate function of this observable are not known exactly, but Galerkin approximations can easily be found by Fourier series, using the basis functions $\phi_j(x) = e^{j2\pi x}$ with $j \in \mathbb{Z}$ [59–61]. This basis is used to compute reference values for the SCGF and the rate function, devoid of $\Delta t$ errors, by projecting the spectral problem (14) in Fourier space and by ensuring that enough basis functions are used. We find in practice that $M = 41$ Fourier modes are sufficient. We use the same Fourier basis for the algorithm, also with $M = 41$ modes, in addition to $T = 2 \times 10^4$ and $T_1 = T_2 = 2 \times 10^3$ for the integration times, as in the first test.

We show in Fig. 4 the results of a typical simulation for $\gamma = 1$ and $k = 1$. The left plot in this figure shows the evolution in time of the estimated SCGF from the eigenvalue iteration (49), while the right plot shows the final estimated eigenfunction $h_k$ for different time steps $\Delta t$, compared with the reference Fourier solution. We see that the results for the eigenvalue and the eigenvector significantly depart from the reference solutions for $\Delta t = 0.05$, but converge to them as $\Delta t$ is decreased, in accordance with Assumption 2 and the theoretical results of [36]. For the same time step used for the Ornstein-Uhlenbeck process, namely, $\Delta t = 5 \times 10^{-3}$, we see no notable difference between the estimated eigenfunction and the reference values, leading to a precise estimation of the SCGF. The convergence of the eigenvalue here is much faster than for the Ornstein-Uhlenbeck process because the space explored is compact, being limited to $[0, 1]$ with periodic boundary conditions.

As for the Ornstein-Uhlenbeck process, we can repeat these simulations over a range of values for $k$ to obtain the SCGF and the rate function. The left plot of Fig. 5 shows that the SCGF is in good agreement with the Fourier solution for $\Delta t = 5 \times 10^{-3}$, and so is the rate function estimated by Legendre transform. However, as seen before, the rate function estimated with the time average $R_T$ of the biasing force shows an offset, although smaller this time, which comes from the noisy estimation of $F_n$. As before, the estimator of the rate function that should be used is the one based on the Legendre transform of the SCGF, with $a_k$ estimated by $A_T$. 
V. CONCLUSION

We have presented a new algorithm for estimating the large deviation functions of time-integrated observables of Markov processes, which characterize the likelihood of their fluctuations in the long-time limit. The algorithm draws on earlier results on stochastic control [28–31], and works by adaptively estimating the principal eigenvalue and eigenfunction of a spectral problem related to the large deviation problem. The adaptive part consists in modifying the process considered, using feedback and reinforcement learning, so as to reach the so-called driven process, which is known to be optimal for the purpose of estimating large deviation functions using importance sampling [32]. In this sense, the algorithm relates to many adaptive importance sampling methods that have been proposed recently for rare event simulations; see, for instance, [62–65]. It is also closely related to diffusion Monte Carlo methods [45, 66, 67], which attempt to estimate the ground-state wavefunction of a many-body quantum system by simulating a related stochastic process.

The proposed algorithm can be applied to diffusions, as illustrated here, but also to Markov chains in discrete or continuous time, and works for both equilibrium and nonequilibrium systems, that is, reversible and non-reversible processes, respectively. Moreover, although the test cases that we have presented are simple, they clearly show that the algorithm has the potential to improve upon other simulation methods, especially cloning methods, since it runs on a single simulation of the process and provides information about how fluctuations are created in time by constructing the driven process in a non-parametric form and with no prior information. A modification of the original cloning algorithm was proposed recently [46] to construct the driven process, but it is based on a different feedback rule that compares two time-dependent histograms, whose estimation is noisy and requires a large number of clones. The results presented here show that a single clone, evolving over a long-enough time, is sufficient. This obviously cuts the computational complexity of estimating large deviations, but also simplifies, as mentioned, the error and convergence analyses of the algorithm.

As with any new proposal, more work is needed to understand the benefits and limitations of the algorithm, to test its applicability to realistic systems, and to benchmark it against other numerical methods. Of particular importance is to derive precise error estimates associated with the space and time discretizations of the spectral elements, large deviation functions, and the driven process.
Recent results about the discretization errors in $\Delta t$ for the SCGF can be found in [36]. These discretization errors are also present in cloning algorithms when applied to diffusions, so they are not specific to the algorithm presented here. The use of Galerkin discretizations is also common to both algorithms and requires further investigations, particularly in the low-noise limit and for processes involving many interacting particles.

ACKNOWLEDGMENTS

We are grateful to Florian Angeletti for useful discussions in the initial phase of this work. G.F. is supported by the Labex Bezout. H.T. was supported by the National Research Foundation of South Africa (Grant nos 90322 and 96199) and Stellenbosch University (Project Funding for New Appointee). This research was also supported in part by the International Centre for Theoretical Sciences (ICTS) during a visit for participating in the program “Large deviation theory in statistical physics: Recent advances and future challenges” (Code: ICTS/Prog-ldt/2017/8).

* gregoire.ferre@enpc.fr
† htouchette@sun.ac.za, htouchet@alum.mit.edu


