ECOLE POLYTECHNIQUE

CENTRE DE MATHÉMATIQUES APPLIQUÉES
UMR CNRS 7641

91128 PALAISEAU CEDEX (FRANCE). Tél: 01 69 33 41 50. Fax: 01 69 33 30 11

http://www.cmap.polytechnique.fr/

Geometric convergence for functionals of Markov processes using sequential control variates.

Emmanuel Gobet and Sylvain Maire

R.I. N^0 535 *May 2004*

GEOMETRIC CONVERGENCE FOR FUNCTIONALS OF MARKOV PROCESSES USING SEQUENTIAL CONTROL VARIATES.

EMMANUEL GOBET AND SYLVAIN MAIRE

ABSTRACT. Using a sequential control variates algorithm, we compute Monte Carlo approximations of solutions of linear partial differential equations connected to linear Markov processes by the Feynman-Kac formula. It includes diffusion processes with or without absorbing/reflecting boundary and jump processes. We prove that the bias and the variance decrease geometrically with the number of steps of our algorithm. Numerical examples show the efficiency of the method on elliptic and parabolic problems.

1. INTRODUCTION

We are concerned with the numerical evaluation of $\mathbf{E}(\Psi(X_s : s \geq t)|X_t = x)$, where $(X_t)_t$ is a Markov process (with linear dynamics) and where Ψ belongs to a class of functionals related to Feynman-Kac representations. These issues arise for example in physics in the computations of the solution of diffusion equations (see [CDL$^+$89]), or in finance in the pricing of European options (see [DG95] and references therein). Monte Carlo methods are usually used to evaluate these expectations for high-dimensional problems or when the functionals are complex. They give a rather poor approximation because of a slow convergence as $\sigma/\sqrt{M}$, M being the number of simulations and σ^2 the relative variance. A better accuracy can nevertheless be reached by using relevant variance-reduction tools like for instance the control variates method or importance sampling [Hal70][New94]. One of the most performing tools is the sequential Monte Carlo approach which consists in using iteratively these variance-reduction ideas [Hal62][Hal70][Boo89]. Using respectively importance sampling and control variates, this approach has been recently developed in [BCP00] for Markov chains and in [Mai03] for the numerical integration of multivariate smooth functions. We have introduced in [GM03] a sequential Monte Carlo method to solve the Poisson equation with Dirichlet boundary conditions over square domains. This method was based on Feynman-Kac computations of pointwise solutions combined with a global approximation on Tchebychef polynomials [BM97]. Pointwise solutions were computed using walk-on-spheres (WOS) simulations of stopped Brownian motion, which induces a simulation error due to the absorption layer thickness. We have nevertheless observed a geometric reduction of both the simulation error and the variance with the number of steps of the algorithm. The global error was comparable to standard deteministic spectral

Date: First version: July 11, 2003. Current version: May 27, 2004.
1991 *Mathematics Subject Classification.* Primary 65Cxx, 65Bxx; Secondary 65M70.
Key words and phrases. Sequential Monte Carlo, Feynman-Kac formula, variance reduction.

methods [BM97] while avoiding the resolution of a linear system. Our goal here is twofold:

- to extend the scope of the approach to general Markov processes connected to linear elliptic and parabolic Dirichlet problems;
- to analyze mathematically the phenomenon of geometric convergence for both the bias and the variance.

This will be achieved for general discretization schemes for the stochastic processes and also for general global approximations of the solution (not only using Tchebychef polynomials). We also emphasize two major improvements compared to [BCP00] where analogous geometric convergences are proved for Markov chains. First we incorporate in our analysis the influence of the discretization error on the underlying process. Second we allow the global solution not to be in the right approximation space.

In Section 2, we make a complete study of the algorithm on elliptic problems with general boundary conditions. At each step of the algorithm, the Monte Carlo computation of $\mathbf{E}(\Psi(X_s : s \geq 0)|X_0 = x)$ at some points x in the domain is required. Then, we build a global approximation using the values at these points. This approximation is used as a control variate at the next step and so on. If the discretization step is small enough, we first prove that the error on the mean value of the global solution reduces geometrically up to a limit directly linked to the approximation error of the exact solution. If furthermore the number of drawings at each step is large enough, we also prove that the variance of the solution reduces geometrically. The proofs of convergence mainly rely on independence properties of the different simulations, on the connection with a linear partial differential equation (PDE) and on the linearity of the functionals w.r.t. the data. This means that the algorithm can be used for Brownian stochastic differential equations (SDEs) with or without absorbing/reflecting boundary, or for Lévy-driven SDEs. The last two sections describe the practical implementation of the algorithm. We first make a discussion on the discretization schemes and on the approximation problems. We then give numerical examples on elliptic and parabolic problems after having precisely studied the speed of convergence of the algorithm on the relative approximation bases. The numerical results confirm the efficiency of the method and the phenomenon of geometric convergence on both the bias and the variance.

2. Statement of the problem

2.1. **Elliptic problems.** Before giving a general formulation, we prefer listing relevant examples. The Markov process underlying to our study is denoted by $X(x) = [X_t(x)]_{t\geq 0}$ and its initial value x belongs to a domain $D \subset \Re^d$. The functionals $\Psi(X_t(x) : t \geq 0) := \Psi(f, g, X(x))$ are related to Feynman-Kac formulas and represented by two continuous functions f and g, respectively defined on $\bar{D}$ and its boundary ∂D. We especially consider

Ex. 1: Brownian SDEs [RY94]: $X_t = x + \int_0^t b(X_s)ds + \int_0^t \sigma(X_s)dW_s$, where W is a Brownian motion. Set $Z_t = e^{-\int_0^t c(X_r)dr}$ and $\tau = \inf\{t \geq 0 : X_t \notin D\}$: we can take $\Psi(f, g, X(x)) = g(X_\tau)Z_\tau - \int_0^\tau f(X_s)Z_s ds$ (provided that $\tau < +\infty$ *a.s.*). If $D = \Re^d$, we may consider $\Psi(f, g, X(x)) = -\int_0^\infty f(X_s)Z_s ds$ ($g \equiv 0$) (see [Fre85]).

Ex. 2: SDEs with reflection on ∂D in the non-tangential direction γ [Fre85]: $X_t = x + \int_0^t b(X_r)dr + \int_0^t \sigma(X_r)dW_r + \int_0^t \gamma(X_r)d\Lambda_r$, where X_t takes values in $\bar{D}$ and where the so-called local time $(\Lambda_t)_t$ is increasing only when X_t is on ∂D. Set $Z_s = e^{-\int_0^s c(X_r)dr - \int_0^s \lambda(X_r)d\Lambda_r}$: we can take $\Psi(f,g,X(x)) = -\int_0^\infty f(X_s)Z_s ds - \int_0^\infty g(X_s)Z_s d\Lambda_s$.

Ex. 3: Lévy-driven SDEs (see [BL84, GM92]): $X_t = x + \int_0^t b(X_r)dr + \int_0^t \sigma(X_r)dW_r + \int_0^t \int_{\Re^p} \Upsilon(X_r, z)\mu(dr, dz)$, where μ is a martingale normalized Poisson measure on $\Re^p$, with Lévy measure $m(dz)$. We can take $\Psi(f,g,X(x)) = -\int_0^\tau f(X_s)Z_s ds$ $(g \equiv 0)$ where τ and Z are defined as in the Brownian case. Note that reflections could be included [BL84]. Similar situations also occur with transport equations [CDL+89].

The processes above are well defined under, for example, Lipschitz assumptions on their coefficients. For the first two cases, the relative infinitesimal generator is given by $L_0\phi = \sum_{i=1}^d b_i(x)\partial_{x_i}\phi + \frac{1}{2}\sum_{i=1}^d [\sigma\sigma^*(x)]_{i,j}\partial^2_{x_i,x_j}\phi$. For the last one, jumps are taken into account by an extra integral kernel. Thus, the associated infinitesimal generator is defined by $L_0\phi(y) + \int_{\Re^p} \left[\phi(y+z) - \phi(y) - z\cdot\nabla\phi(y)\mathbf{1}_{|z|\le 1}\right] M(y,dz)$ (here the measure $M(y,\cdot)$ is defined by $M(y,A) = m\{z : \Upsilon(y,z) \in A\}$). The key point is that these operators are linear. The goal of this section is to describe how to evaluate efficiently the quantity

$$u(x) = \mathbf{E}\big(\Psi(f,g,X(x))\big). \tag{2.1}$$

We assume

(H1) The process X, the domain D and the data (f,g) are such that $\Psi(f,g,X(x))$ is a linear map w.r.t. the data (f,g) and that $\mathbf{V}\mathrm{ar}(\Psi(f,g,X(x))) < +\infty$.

This assumption is natural in view of the previous examples. Usually, it imposes restrictions on the domain, on the sign of c arising in Z... See Proposition 4.1 for explicit conditions about absorbed Brownian SDEs. We now assume that u solves an elliptic PDE with appropriate boundary conditions, using the connection between Markov processes and PDEs.

(H2) The process X, the domain D and the data (f,g) are such that u is a classic[1] solution of

$$\begin{cases} \mathcal{A}u = f & \text{in } D, \\ \mathcal{B}u = g & \text{on } \partial D, \end{cases} \tag{2.2}$$

where $\mathcal{A}$ and $\mathcal{B}$ are second order linear operators.

The domain may be bounded or unbounded in some cases, the diffusion processes may be elliptic or hypo-elliptic ... We refer the reader to [Fri75, BL84, Fre85] for details and references. For the first example, we have $\mathcal{A}u = L_0u - cu$ and $\mathcal{B}u = u$ (Dirichlet boundary condition), for the second example $\mathcal{B}u = \nabla_x u.\gamma - \lambda u$ (Neumann boundary condition)... Second order operators $\mathcal{B}$ arise with Ventcel's boundary conditions corresponding to processes having a diffusion part on ∂D (see [Cat92]). These different boundary conditions can also be mixed.

[1]The regularity of u depends on the type of the boundary condition.

2.2. **Parabolic problems.** The extension to problems with a terminal time T is straightforward. Formally, it is achieved by considering the time-space process $(t, X_t)_t$ in the domain $]0,T[\times D$. Then the operator $\mathcal{A}$ has to replaced by $\partial_t + \mathcal{A}$. In that case, we can take $D = \Re^d$ (this is the so-called Cauchy problem). The coefficients of X and the domain D may also be time-dependent [BL84, Lie96]. The reader can check that the following algorithm and its convergence proof are derived in the same way.

3. Study of the algorithm

3.1. **Description.** We now describe our algorithm, which computes iterative approximations $(u_n)_{n\geq 0}$ of the global solution u. These approximations rely on the computations of $\mathbf{E}[\Psi(\tilde{f},\tilde{g},X(x))]$ (for data $\tilde{f}$ and $\tilde{g}$ possibly different from f and g) at some points $(x_i)_{1\leq i\leq N}$.

Initialization. We begin with $u_0 \equiv 0$.

Iteration n, step 1. Assume that an approximated solution u_{n-1} of class $C^2(\bar{D})$ is built at stage $n-1$ and that the representation $u_{n-1}(x) = \mathbf{E}(\Psi[\mathcal{A}u_{n-1}, \mathcal{B}u_{n-1}, X(x)])$ holds (which simply means that u_{n-1} solves (2.2) with $f = \mathcal{A}u_{n-1}$ and $g = \mathcal{B}u_{n-1}$). The idea is to compute a correction $y_n = u - u_{n-1}$ on this approximation. Using (2.1), we have

$$y_n(x) = \mathbf{E}(\Psi[f - \mathcal{A}u_{n-1}, g - \mathcal{B}u_{n-1}, X(x)]). \tag{3.1}$$

In the above equation, the expectation is relative to the law of X and not to the law of u_{n-1} which can be random. We intend to compute a Monte Carlo approximation of $y_n(x_i)$. For this, we replace the simulations of the random variable $\Psi[f - \mathcal{A}u_{n-1}, g - \mathcal{B}u_{n-1}, X(x_i)]$ by $\Psi[f - \mathcal{A}u_{n-1}, g - \mathcal{B}u_{n-1}, X^{\Delta}(x_i)]$ using a suitable discretization procedure $X^{\Delta}(x_i)$ for the stochastic process $X(x_i)$. For the moment, we prefer keeping quite abstract notations concerning the discretization scheme, since mild assumptions are required (see assumption **(H4)** below). We just mention that Δ usually represents the discretization parameter which tends to 0 (for instance, for the WOS procedure [Sab91], Δ is the space step; for the Euler procedure [CPS98, Gob01], Δ is the time step). Consequently, $y_n(x_i)$ is approximated by

$$\bar{y}_n(x_i) = \frac{1}{M}\sum_{m=1}^{M} \Psi[f - \mathcal{A}u_{n-1}, g - \mathcal{B}u_{n-1}, X^{\Delta,n,m}(x_i)] \tag{3.2}$$

using M independent simulations of the paths $X^{\Delta,n,m}(x_i)$. They are also generated independently of everything else. In fact, the independence of simulations at different points is not crucial to ensure the convergence of the algorithm. Nevertheless, we think that dependent drawings slow down the convergence of the method and are less adapted to parallel computations.

Iteration n, step 2. In order to build a global approximation y_n based on the values $(\bar{y}_n(x_i))_i$, we use a linear approximation [CHQZ88, BM97]. The linear approximation of a function $w(\cdot)$ at some points $(x_j)_j$ can always be written

$$\mathcal{P}w(x) = \sum_{j=1}^{N} w(x_j)\mathcal{C}_j(x) \tag{3.3}$$

for some functions $(\mathcal{C}_j)_j$. In addition we assume a stability property:

$$\mathcal{P}[\mathcal{P}w] = \mathcal{P}w \quad \text{for any function } w. \tag{3.4}$$

If we use an interpolation, the functions $(\mathcal{C}_j)_j$ simply verify $\mathcal{C}_j(x_i) = \delta_{i,j}$. This is for example the case of interpolation in dimension 1 on Lagrange polynomials $L_j(x) = \frac{\prod_{i\neq j}(x-x_j)}{\prod_{i\neq j}(x_i-x_j)}$. This approximation can also come from a problem of fitting an approximation model $\sum_{k=1}^K \alpha_k\varphi_k$ on some basis functions $(\varphi_k)_k$ to the values $(w(x_i))_i$. This leads to a discrete least-square problem [Bjö96], using the norm associated to the discrete inner product $< u, v >_\mu = \sum_{j=1}^N \mu_j v(x_j)u(x_j)$ for some positive weights $(\mu_j)_j$, which consists in the minimization of the squared norm $\|\sum_{k=1}^K \alpha_k\varphi_k - w\|_\mu^2$. The optimal coefficients $(\alpha_k)_k$ are hence solution of a linear system $A\alpha = b$ with $A_{ik} = < \varphi_i, \varphi_k >_\mu$, $b_k = < w, \varphi_k >_\mu$. As $\alpha = A^{-1}b$, we get $\alpha_k = \sum_{i=1}^N A_{ik}^{-1} \sum_{j=1}^N \mu_j\varphi_k(x_j)w(x_j)$ and we are still in a linear form of type (3.3) in letting

$$\mathcal{C}_j(x) = \mu_j \sum_{k=1}^K \sum_{i=1}^N A_{ik}^{-1}\varphi_k(x_j)\varphi_k(x).$$

If the $(\varphi_k)_k$ are moreover orthonormal with respect to $< \cdot, \cdot >_\mu$, we simply have $\mathcal{C}_j(x) = \mu_j \sum_{k=1}^K \varphi_k(x_j)\varphi_k(x)$. A slightly different situation is the computation of the projection of the function w on orthonormal polynomials $(T_n)_n$ with respect to the inner product $< u, v >_\nu = \int_{[a,b]} v(x)u(x)\nu(x)dx$ where ν is a positive weight function on the interval $D = [a, b]$. We have $w(x) \simeq \sum_{k=0}^N \alpha_k T_k(x)$ with $\alpha_k = < w, T_k >_\nu$ and the points $(x_i)_i$ are used to build quadrature formulas to compute accurately the coefficients $(\alpha_k)_k$. These points are usually chosen as the zeros of T_{N+1} which makes the quadrature formula exact for all polynomials of degree $\leq 2N + 1$. Note that in this case, this approximation is equal to the interpolation at the same points. Another possibility is the Gauss-Lobatto formulas where the boundaries of the interval are chosen as quadrature points. In higher dimensions, the approximations are built using tensor products. In any case the approximations are still linear and they will be described in detail in Section 4. Note that the stability property (3.4) holds for all of these approximations. Once one of the above approximations has been chosen we just write

$$u_n = u_{n-1} + \mathcal{P}\bar{y}_n = u_{n-1} + \sum_{j=1}^N \bar{y}_n(x_j)\mathcal{C}_j \tag{3.5}$$

and we can proceed to the next iteration. We assume furthermore in the sequel that

(H3) The functions $(\mathcal{C}_j)_{1\leq j\leq N}$ are of class $C^2(\bar{D})$. Furthermore, for any $x \in D$ we have $\mathbf{V}\mathrm{ar}(\Psi(\mathcal{A}\mathcal{C}_j, \mathcal{B}\mathcal{C}_j, X(x))) < +\infty$ and

$$\mathbf{E}(\Psi(\mathcal{A}\mathcal{C}_j, \mathcal{B}\mathcal{C}_j, X(x))) = \mathcal{C}_j(x). \tag{3.6}$$

In other words, $\mathcal{C}_j$ (formally) solves (2.2) with data $f = \mathcal{A}\mathcal{C}_j$ and $g = \mathcal{B}\mathcal{C}_j$. Hence, $\mathcal{P}w \in C^2(\bar{D})$ for any function w. In particular, u_n is of class $C^2(\bar{D})$ for any n and satisfies the representation $u_n(x) = \mathbf{E}(\Psi[\mathcal{A}u_n, \mathcal{B}u_n, X(x)])$, which makes our algorithm valid. Note also that the stability property (3.4) written for u_n gives

$$\mathcal{P}u_n = u_n \quad \text{for any } n \geq 0. \tag{3.7}$$

3.2. **Convergence results.** Our goal is now to estimate the convergence of

$$m_n := \sup_{1\le i\le N} |\mathbf{E}(u_n(x_i) - u(x_i))|, \quad v_n := \sup_{1\le i\le N} \mathbf{V}\mathrm{ar}(u_n(x_i)). \tag{3.8}$$

It is possible to derive other measures of the error, like $\frac{1}{N}\sum_{i=1}^N |\mathbf{E}(u_n(x_i) - u(x_i))|$, without major differences. However in this work, rather than finding the optimal way to measure the error, we prefer focusing on the phenomenon of geometric convergence. We need to introduce some extra notations regarding the scheme X^Δ. For a deterministic smooth function $\tilde g$ we set

$$\begin{aligned} e(\tilde g, \Delta, x) &= \mathbf{E}\big(\Psi[\mathcal{A}\tilde g, \mathcal{B}\tilde g, X^\Delta(x)]\big) - \mathbf{E}\big(\Psi[\mathcal{A}\tilde g, \mathcal{B}\tilde g, X(x)]\big), \\ V(\tilde g, \Delta, x) &= \mathbf{V}\mathrm{ar}\big(\Psi[\mathcal{A}\tilde g, \mathcal{B}\tilde g, X^\Delta(x)]\big). \end{aligned}$$

We state mild assumptions on the discretization scheme X^Δ, which allows great generality on the procedures that can be used.

(H4)
1) The map $(\tilde f, \tilde g) \mapsto \Psi(\tilde f, \tilde g, X^\Delta(x))$ is linear.
2) The discretization errors $[e(\mathcal{C}_j, \Delta, x_i)]_{1\le i,j\le N}$ converge to 0 as $\Delta \to 0$.
3) The variances $[V(u, \Delta, x_i)]_i$ and $[V(\mathcal{C}_k, \Delta, x_i)]_{i,k}$ are finite.
4) The latter are uniformly bounded for Δ close to 0: $\limsup_{\Delta\to 0} V(\mathcal{C}_k, \Delta, x_i) < \infty$ for any i and k.

The first assumption is natural since the initial map $(\tilde f, \tilde g) \mapsto \Psi(\tilde f, \tilde g, X(x))$ is linear. The second one is minimal since it requires only the weak convergence of the discretization scheme for the $C^2(\bar D)$-functions $(\mathcal{C}_k)_k$. The last two ones are also very natural since they are satisfied for X (see **(H1-H3)**). A practical verification of **(H4)** will be given in Section 4.1. It easily follows from statement 3) that $V(u - \mathcal{P}u, \Delta, x_i) < +\infty$ for any i. This justifies the finiteness of the terms which appear in Theorem 3.2.

We first state a convergence result for the bias.

Theorem 3.1. *Assume* **(H1-H2-H3-H4)**. *Then, for any $n \ge 1$, one has*

$$m_n \le \rho_m\, m_{n-1} + \sup_{1\le i\le N} |[\mathcal{P}u - u](x_i) + \mathcal{P}[e(u - \mathcal{P}u, \Delta, \cdot)](x_i)|. \tag{3.9}$$

where $\rho_m = \sup_{1\le i\le N}\big[\sum_{j=1}^N |\mathcal{P}[e(\mathcal{C}_j, \Delta, \cdot)](x_i)|\big]$. For Δ small enough, one has $\rho_m < 1$. Thus, the convergence of $(m_n)_n$ is geometric at rate ρ_m, up to a threshold equal to

$$\limsup_n m_n \le \frac{1}{1-\rho_m} \sup_{1\le i\le N} |[\mathcal{P}u - u](x_i) + \mathcal{P}[e(u - \mathcal{P}u, \Delta, \cdot)](x_i)|. \tag{3.10}$$

The upper limit for the bias strongly depends on the quality of the approximation of u by the operator $\mathcal{P}$. Note that if u is in the right approximation space ($\mathcal{P}u \equiv u$), the first term in the r.h.s. of (3.10) cancels and the bias m_n converges geometrically to 0. In other words, even if the simulations are biased because of Δ, the bias vanishes at the limit. This is a surprising and very interesting phenomenon. However, unlike the direct Monte Carlo procedure, there is no guarantee that $\lim_{\Delta\to 0}\limsup_n m_n = 0$, except in the case of the interpolation operator $\mathcal{P}$ (i.e. $\mathcal{P}u(x_i) = u(x_i)$ for any x_i). We now state the convergence of the variance $(v_n)_n$.

Theorem 3.2. *Assume* **(H1-H2-H3-H4)** *and set*

$$C(\Delta,N) = 2\sup_{1\le i\le N}\sum_{j=1}^{N}\mathcal{C}_j^2(x_i)\big[\sum_{k=1}^{N}\sqrt{V(\mathcal{C}_k,\Delta,x_j)}\,\big]^2,$$

$$\rho_v = \sup_{1\le i\le N}\big(\sum_{j=1}^{N}|\mathcal{P}[e(\mathcal{C}_j,\Delta,\cdot)](x_i)|\big)^2 + \frac{C(\Delta,N)}{M}.$$

Then, for any $n\ge 1$, one has

$$v_n \le \rho_v v_{n-1} + \frac{1}{M}\big\{2\sup_{1\le i\le N}\sum_{j=1}^{N}\mathcal{C}_j^2(x_i)V(u-\mathcal{P}u,\Delta,x_j) + C(\Delta,N)m_{n-1}^2\big\}. \tag{3.11}$$

For Δ small enough and M large enough, one has $\rho_v<1$. Thus, the convergence of $(v_n)_n$ is geometric at rate ρ_v, up to a threshold equal to

$$\limsup_n v_n \le \frac{1}{(1-\rho_v)M}\big\{2\sup_{1\le i\le N}\sum_{j=1}^{N}\mathcal{C}_j^2(x_i)V(u-\mathcal{P}u,\Delta,x_j) + C(\Delta,N)\limsup_n m_n^2\big\}. \tag{3.12}$$

Note that when $\rho_v<1$, $\rho_m<1$, so that the geometric convergence holds simultaneously for the bias and for the variance. As for the bias, if $\mathcal{P}u=u$, $\limsup_n m_n=0$ and thus $\limsup_n v_n=0$: the variance v_n converges geometrically to 0, provided that $1/\Delta$ and M are large enough.

3.3. **Proofs of convergence.** To make the distinction between what is simulated before stage n and at stage n, we define the usual conditional expectations and variances

$$\mathbf{E}^{n-1}(Y) = \mathbf{E}\Big(Y\big|\sigma(X^{\Delta,n',m}(x_i): 1\le n'\le n-1; 1\le m\le M; 1\le i\le N)\Big)$$

and $\mathbf{V}\mathrm{ar}^{n-1}(Y) = \mathbf{E}^{n-1}(Y^2) - [\mathbf{E}^{n-1}(Y)]^2$. Note that the construction of the algorithm yields that the discretized processes $[X^{\Delta,n,m}(x_i)]_{m,i,n}$ are independent.

3.3.1. *Proof of Theorem 3.1.* (3.10) is a straightforward consequence of (3.9). Before proving (3.9), we transform the expression of $u_n(x_i)$ for a fixed x_i. Using (3.5), (3.2) and the PDE solved by u, we get $u_n(x_i) = u_{n-1}(x_i) + \frac{1}{M}\sum_{j=1}^{N}\sum_{m=1}^{M}\Psi[\mathcal{A}(u-u_{n-1}),\mathcal{B}(u-u_{n-1}),X^{\Delta,n,m}(x_j)]\mathcal{C}_j(x_i)$. In view of (3.7), note that

$$u-u_{n-1} = u-\mathcal{P}u + \sum_{k=1}^{N}(u-u_{n-1})(x_k)\mathcal{C}_k \tag{3.13}$$

and that $\Psi[\mathcal{A}(u-u_{n-1}),\mathcal{B}(u-u_{n-1}),X^{\Delta,n,m}(x_j)]$ equals

$$\Psi[\mathcal{A}(u-\mathcal{P}u),\mathcal{B}(u-\mathcal{P}u),X^{\Delta,n,m}(x_j)] + \sum_{k=1}^{N}(u-u_{n-1})(x_k)\Psi[\mathcal{A}\mathcal{C}_k,\mathcal{B}\mathcal{C}_k,X^{\Delta,n,m}(x_j)],$$

because of the linearity of $\Psi[(\cdot,\cdot), X^{\Delta}(x)]$ under **(H4)**. Thus, we obtain

$$u_n(x_i) = u_{n-1}(x_i) + \frac{1}{M}\sum_{j=1}^{N}\sum_{m=1}^{M}\Big[\Psi[\mathcal{A}(u-\mathcal{P}u), \mathcal{B}(u-\mathcal{P}u), X^{\Delta,n,m}(x_j)] + \sum_{k=1}^{N}(u-u_{n-1})(x_k)\Psi[\mathcal{AC}_k, \mathcal{BC}_k, X^{\Delta,n,m}(x_j)]\Big]\mathcal{C}_j(x_i). \tag{3.14}$$

• **Computation of $\mathbf{E}^{n-1}(u_n(x_i))$.** As $[X^{\Delta,n,m}(x_j)]_{m,j}$ is independent of u_{n-1}, we readily get

$$\begin{aligned}\mathbf{E}^{n-1}(u_n(x_i)) =& u_{n-1}(x_i) + \sum_{j=1}^{N}\mathbf{E}\left(\Psi[\mathcal{A}(u-\mathcal{P}u), \mathcal{B}(u-\mathcal{P}u), X^{\Delta}(x_j)]\right)\mathcal{C}_j(x_i)\\ &+ \sum_{k=1}^{N}(u-u_{n-1})(x_k)\sum_{j=1}^{N}\mathbf{E}\left(\Psi[\mathcal{AC}_k, \mathcal{BC}_k, X^{\Delta}(x_j)]\right)\mathcal{C}_j(x_i).\end{aligned} \tag{3.15}$$

Note that

a) using (2.1) and (3.6), we have $\mathbf{E}\big(\Psi[\mathcal{A}(u-\mathcal{P}u), \mathcal{B}(u-\mathcal{P}u), X^{\Delta}(x_j)]\big) = e(u-\mathcal{P}u, \Delta, x_j) + (u-\mathcal{P}u)(x_j)$.
b) using (3.6), we have $\mathbf{E}\left(\Psi[\mathcal{AC}_k, \mathcal{BC}_k, X^{\Delta}(x_j)]\right) = e(\mathcal{C}_k, \Delta, x_j) + \mathcal{C}_k(x_j)$.
c) owing to (3.4), we have $\sum_{j=1}^{N}\mathcal{C}_k(x_j)\mathcal{C}_j(x_i) = \mathcal{C}_k(x_i)$.

Plugging these identities in (3.15), it readily follows that

$$\begin{aligned}\mathbf{E}^{n-1}(u_n(x_i)) &= u_{n-1}(x_i) + \sum_{j=1}^{N}\big[e(u-\mathcal{P}u, \Delta, x_j) + (u-\mathcal{P}u)(x_j)\big]\mathcal{C}_j(x_i)\\ &\quad + \sum_{k=1}^{N}(u-u_{n-1})(x_k)\big[\mathcal{C}_k(x_i) + \sum_{j=1}^{N}e(\mathcal{C}_k, \Delta, x_j)\mathcal{C}_j(x_i)\big]\\ &= \mathcal{P}u(x_i) + \mathcal{P}[e(u-\mathcal{P}u, \Delta, \cdot)](x_i) + \sum_{k=1}^{N}(u-u_{n-1})(x_k)\mathcal{P}[e(\mathcal{C}_k, \Delta, \cdot)](x_i),\end{aligned} \tag{3.16}$$

simplifications at the last line arising from the equality (3.4).

• **Computation of $\mathbf{E}(u_n(x_i))$.** Taking the expectation in (3.16) we obtain

$$\begin{aligned}\mathbf{E}\big(u_n(x_i) - u(x_i)\big) &= \mathcal{P}u(x_i) - u(x_i) + \mathcal{P}[e(u-\mathcal{P}u, \Delta, \cdot)](x_i)\\ &\quad + \sum_{k=1}^{N}\mathbf{E}((u-u_{n-1})(x_k))\mathcal{P}[e(\mathcal{C}_k, \Delta, \cdot)](x_i).\end{aligned}$$

It remains to take absolute values and the supremum over i on both sides to complete the proof of (3.9). □

3.3.2. *Proof of Theorem 3.2.* Note that the inequality $\rho_v < 1$ holds for Δ small enough and M large enough. Indeed, under **(H4)** $C(\Delta, N)$ remains uniformly bounded w.r.t. Δ close to 0. We only prove (3.11). Taking some fixed x_i, we have

$$\mathbf{V}\mathrm{ar}(u_n(x_i)) = \mathbf{V}\mathrm{ar}\big[\mathbf{E}^{n-1}(u_n(x_i))\big] + \mathbf{E}\big[\mathbf{V}\mathrm{ar}^{n-1}(u_n(x_i))\big]. \tag{3.17}$$

• **Computation of $\mathbf{V}\mathrm{ar}[\mathbf{E}^{n-1}(u_n(x_i))]$.** In view of (3.16), we have

$$\mathbf{V}\mathrm{ar}\big[\mathbf{E}^{n-1}(u_n(x_i))\big] = \mathbf{V}\mathrm{ar}\big[\sum_{j=1}^N u_{n-1}(x_j)\mathcal{P}[e(\mathcal{C}_j,\Delta,\cdot)](x_i)\big]$$

$$\leq v_{n-1}\big(\sum_{j=1}^N |\mathcal{P}[e(\mathcal{C}_j,\Delta,\cdot)](x_i)|\big)^2 \tag{3.18}$$

where we have used the standard inequality

$$\mathbf{V}\mathrm{ar}(\sum_{j=1}^N \alpha_j Y_j) = \sum_{j_1,j_2=1}^N \alpha_{j_1}\alpha_{j_2}\mathbf{C}\mathrm{ov}(Y_{j_1},Y_{j_2})$$

$$\leq \sum_{j_1,j_2=1}^N |\alpha_{j_1}|\ |\alpha_{j_2}|\sqrt{\mathbf{V}\mathrm{ar}(Y_{j_1})}\sqrt{\mathbf{V}\mathrm{ar}(Y_{j_2})} = \Big[\sum_{j=1}^N |\alpha_j|\sqrt{\mathbf{V}\mathrm{ar}(Y_j)}\Big]^2 \tag{3.19}$$

for any real numbers $(\alpha_j)_j$ and any square integrable real random variables $(Y_j)_j$.
• **Computation of $\mathbf{V}\mathrm{ar}^{n-1}(u_n(x_i))$.** We invoke the independence of $[X^{\Delta,n,m}(x_i)]_{m,i}$ and u_{n-1} in (3.14) to derive

$$\begin{aligned}
&\mathbf{V}\mathrm{ar}^{n-1}(u_n(x_i))\\
&= \sum_{j=1}^N \frac{\mathcal{C}_j^2(x_i)}{M}\mathbf{V}\mathrm{ar}^{n-1}\Big[\Psi[\mathcal{A}(u-\mathcal{P}u),\mathcal{B}(u-\mathcal{P}u),X^\Delta(x_j)]\\
&\qquad\qquad + \sum_{k=1}^N (u-u_{n-1})(x_k)\Psi[\mathcal{AC}_k,\mathcal{BC}_k,X^\Delta(x_j)]\Big]\\
&\leq \sum_{j=1}^N \frac{\mathcal{C}_j^2(x_i)}{M}\Big[\sqrt{V(u-\mathcal{P}u,\Delta,x_j)} + \sum_{k=1}^N |u-u_{n-1}|(x_k)\sqrt{V(\mathcal{C}_k,\Delta,x_j)}\ \Big]^2,
\end{aligned}$$

using (3.19) at the last inequality. Applying the inequality $\mathbf{E}(\alpha_0+\sum_{k=1}^N \alpha_k Y_k)^2 \leq 2\alpha_0^2 + 2\big(\sum_{k=1}^N |\alpha_k|\sqrt{\mathbf{E}(Y_k^2)}\big)^2$ which can be proved as (3.19), we get

$$\begin{aligned}
\mathbf{E}(\mathbf{V}\mathrm{ar}^{n-1}(u_n(x_i))) \leq\ & 2\sup_{1\leq i\leq N}\sum_{j=1}^N \frac{\mathcal{C}_j^2(x_i)}{M}\big\{V(u-\mathcal{P}u,\Delta,x_j)\\
&+ \sup_{1\leq k\leq N}\mathbf{E}((u-u_{n-1})^2(x_k))\big[\sum_{k=1}^N \sqrt{V(\mathcal{C}_k,\Delta,x_j)}\ \big]^2\big\}.
\end{aligned}$$

Combine this estimate with (3.18) and (3.17), use $\sup_{1\leq k\leq N}\mathbf{E}((u-u_{n-1})^2(x_k)) \leq v_{n-1}+m_{n-1}^2$, and take the supremum over i to complete the proof of (3.11). □

4. Influence of parameters of the algorithm

We mainly focus on the first example of Brownian SDEs with a Dirichlet boundary condition. We first give a set of explicit assumptions implying **(H1-H2)** and (3.6) in this case.

(H2') i) The functions b and σ are Lipschitz continuous on $\bar{D}$ and $\sigma\sigma^*(x) \geq \epsilon_0 \mathrm{I}_d$ uniformly w.r.t. $x \in D$ ($\epsilon_0 > 0$).

ii) D is a bounded domain and each point of its boundary ∂D satisfies the *exterior cone condition*: for any $x \in \partial D$, there exists a finite right circular cone $\mathcal{K}$, with vertex x, such that $\bar{\mathcal{K}} \cap \bar{D} = \{x\}$.

iii) The function g is continuous on ∂D, f and $c \geq 0$ are uniformly Hölder continuous in $\bar{D}$ (with exponent $\alpha \in]0,1[$).

Proposition 4.1. *Under* **(H2')**, **(H1)** *and* **(H2)** *are fulfilled. Furthermore, if* $\mathcal{C}_j$ *is of class* $C^2(\bar{D})$, **(H3)** *holds.*

Proof. The variance in **(H1)** is finite. Indeed, on the one hand the functions g and f are bounded and c is non negative. On the other hand, we have $\mathbf{E}_x(\tau) < +\infty$ (which automatically induces exponential moments for τ, see [Fre85] Section 3.3).

The proof of **(H2)** is somewhat classic except that D is not smooth here. We just recall the two main steps. First, the existence of a solution to (2.2) follows from the remark after Theorem 6.13 in [GT83], noting that under **(H2')** every point of the boundary has a barrier (see Problem 6.3 in [GT83]). Second, Theorem 2.1 from Section 2.2 in [Fre85] states that the solution is given by (2.1). This is achieved by applying Itô's formula to u and using some careful localization procedures because derivatives of u explode near the boundary (see also Appendix B). **(H3)** is proved analogously. □

4.1. **Verification of assumption (H4).** We propose to check it when we use the so-called *discrete Euler scheme* [Gob00], which is the simplest procedure that can be used for general stopped diffusions. An alternative is the WOS scheme, which is especially efficient when we are dealing with the Brownian motion (see [HMG03] and references therein). Some refinements to the discrete Euler scheme are also possible, using Brownian bridge simulations [Bal95, Gob00, Gob01].

For a given time step Δ and discretization times $t_k = k\Delta$, the Euler scheme is defined by $X_0^\Delta = x$ and $X_{t_{k+1}}^\Delta - X_{t_k}^\Delta = b(X_{t_k}^\Delta)(t_{k+1} - t_k) + \sigma(X_{t_k}^\Delta)(W_{t_{k+1}} - W_{t_k})$, which can be written in continuous time as

$$X_t^\Delta = x + \int_0^t b(X_{\phi(s)}^\Delta)ds + \int_0^t \sigma(X_{\phi(s)}^\Delta)dW_s. \tag{4.1}$$

Here, $\phi(s) = t_k$ for $t_k \leq s < t_{k+1}$. The approximated exit time is defined by $\tau^\Delta = \inf\{t_k \geq 0 : X_{t_k}^\Delta \notin D\}$. Thus, to approximate $\Psi(f, g, X(x))$, we simply propose

$$\Psi(f, g, X^\Delta(x)) = g(X_{\tau^\Delta}^\Delta)Z_{\tau^\Delta}^\Delta - \int_0^{\tau^\Delta} f(X_{\phi(s)}^\Delta)Z_{\phi(s)}^\Delta ds,\ Z_s^\Delta = e^{-\int_0^s c(X_{\phi(u)}^\Delta)du}. \tag{4.2}$$

Here, g is evaluated at $X_{\tau^\Delta}^\Delta$ which is not a priori on ∂D: hence, in (4.2) g has to understood as a bounded continuous function on the whole space. In view of (4.2), **(H4)**-1) is clearly fulfilled. To verify 2-3-4) of **(H4)**, our main tool is the following theorem, which is original in this context of elliptic problems and whose proof is postponed to Appendix A.

Theorem 4.1. *Assume* **(H2')**. *Then, the following assertions hold.*

a) *For any* $p \geq 1$, $\sup_{x \in \bar{D}} \mathbf{E}_x(\tau^p) + \limsup_{\Delta \to 0} \sup_{x \in \bar{D}} \mathbf{E}_x([\tau^\Delta]^p) < \infty$.

b) $\lim_{\Delta \to 0} \tau^\Delta = \tau$ *in* $\mathbf{L}_p$ *for any* $p \geq 1$.

c) $\lim_{\Delta \to 0} X_{\tau^\Delta}^\Delta = X_\tau$ *in probability.*

d) *For any bounded continuous* γ, $\limsup_{\Delta\to 0} \int_0^{\tau^D\wedge\tau} |\gamma(X_s) - \gamma(X^\Delta_{\phi(s)})|ds = 0$ *in* $\mathbf{L}_p$ *for any* $p \geq 1$.

Since $|\Psi[\mathcal{A}u, u, X^\Delta(x)]| = |\Psi[f, u, X^\Delta(x)]| \leq C(1+\tau^\Delta)$ and $|\Psi[\mathcal{A}\mathcal{C}_k, \mathcal{C}_k, X^\Delta(x)]| \leq C(k)(1+\tau^\Delta)$, we easily get that **(H4)**-3) and **(H4)**-4) are fulfilled in view of a). It remains to prove that $e(\mathcal{C}_k, \Delta, x) = \mathbf{E}\big(\Psi[\mathcal{A}\mathcal{C}_k, \mathcal{C}_k, X^\Delta(x)] - \Psi[\mathcal{A}\mathcal{C}_k, \mathcal{C}_k, X(x)]\big)$ converges to 0 when $\Delta \to 0$, for any $\mathcal{C}_k$ of class $C^2(\bar{D})$. Using c), we get the convergence of $\mathcal{C}_k(X^\Delta_{\tau^\Delta})$ to $\mathcal{C}_k(X_\tau)$ in probability, thus in $\mathbf{L}_1$ since $\mathcal{C}_k$ is bounded. Since $\exp(\cdot)$ is 1-Lipschitz on $\Re^-$ and $c(\cdot)$ is non-negative, we have $|Z^\Delta_{\tau^\Delta} - Z_\tau| \leq |\int_0^{\tau^\Delta} c(X^\Delta_{\phi(u)})du - \int_0^\tau c(X_u)du|$, which converges to 0 in $\mathbf{L}_1$, using b) and d). For the convergence in $\mathbf{L}_1$ of $\int_0^{\tau^\Delta} f(X^\Delta_{\phi(s)})Z^\Delta_{\phi(s)}ds$ to $\int_0^\tau f(X_s)Z_s ds$, the previous arguments apply and this completes the verification of **(H4)**-2).

4.2. **Impact of the approximation operator.** The discretization parameter Δ has to be chosen small enough to ensure the geometric convergence of the bias. This convergence depends on the approximation operator as we must have $\rho_m < 1$. As it is mainly described by the sensitivity of regular functions to the discretization error, it actually depends very little on the approximation operator. The geometric convergence of the variance (described by the condition $\rho_v < 1$) depends a lot more on the choice of the approximation, but in the same way concerning the discretization parameter. In order to study this convergence, we can hence focus on the case $\Delta = 0$. In this ideal case, we have

$$v_n \leq v_{n-1}\frac{C(0,N)}{M} + \frac{2}{M}\sup_{1\leq i\leq N}\sum_{j=1}^N \mathcal{C}_j^2(x_i)V(u-\mathcal{P}u, 0, x_j)$$

with

$$C(0,N) = 2\sup_{1\leq i\leq N}\sum_{j=1}^N \mathcal{C}_j^2(x_i)\left(\sum_{k=1}^N \sqrt{V(\mathcal{C}_k, 0, x_j)}\right)^2.$$

The quantities $V(u-\mathcal{P}u, 0, x_j)$ and $V(\mathcal{C}_k, 0, x_j)$ can be computed as

$$V(u-\mathcal{P}u, 0, x_j) = \mathbf{E}_{x_j}\left[\int_0^\tau Z_s^2 \left|\nabla_x(u-\mathcal{P}u)\sigma\right|^2(X_s)ds\right],$$
$$V(\mathcal{C}_k, 0, x_j) = \mathbf{E}_{x_j}\left[\int_0^\tau Z_s^2 \left|\nabla_x \mathcal{C}_k\sigma\right|^2(X_s)ds\right]$$

using Lemma B.1 given in Appendix B. The first term enables to control the final error and the second one the speed of convergence of the algorithm. They both depend only on the gradient of the basis functions and not on the second derivatives of these functions. It is quite difficult to make a general discussion on the optimal choice of the approximation in a general domain. We prefer focusing on polynomial interpolations on square domains and give explicit computations of the convergence parameters in this case. The process X remains general.

4.3. **Gauss-Tchebychef interpolation on** $]-1,1[^d$**.** The Tchebychef polynomials $T_n(x) = \cos(n\arccos(x))$ are the orthogonal polynomials on $]-1,1[$ with respect to the inner product $< P, Q >_w = \int_{-1}^1 P(x)Q(x)w(x)dx$ where $w(x) = \frac{1}{\sqrt{1-x^2}}$. We

have $\|T_0\|_w^2 = \pi$ and $\|T_n\|_w^2 = \frac{\pi}{2}$ if $n \geq 1$. In dimension one, the interpolation polynomial $\mathcal{P}_N(u)$ of the function u at the Tchebychef abscissae

$$x_k = \cos\big(\frac{2k+1}{N+1}\frac{\pi}{2}\big),\ k = 0, 1..N,$$

is given by [Bjö96]

$$\mathcal{P}_N(u) = \sum_{n=0}^{N} \alpha_n T_n$$

with

$$\alpha_n = \frac{\pi}{\|T_n\|_w^2 (N+1)} \sum_{k=0}^{N} u(x_k) T_n(x_k).$$

This interpolation is optimal with respect to the sup norm. The control of u and of its derivative is given in the following theorem (see [CHQZ88] p.298).

Theorem 4.2. *Denote by H_w^m the w-weighted Sobolev space with regularity $m \in \mathbf{N}^*$. Then $\exists\, c_1, c_2 > 0$ such that $\forall u \in H_w^m$, we have*

$$\|u - \mathcal{P}_N(u)\|_w \leq c_1 N^{-m} \|u\|_{H_w^m}, \quad \|u - \mathcal{P}_N(u)\|_{H_w^1} \leq c_2 N^{2-m} \|u\|_{H_w^m}.$$

The Tchebychef interpolation of a function $u : D =]-1,1[^d \mapsto \mathbf{R}$ is built using the same process than in dimension one. The interpolation polynomial $\mathcal{P}_N(u)$ at the N^d points of a tensored Tchebychef grid and evaluated at $z = (z_1, \cdots, z_d)$ is

$$\mathcal{P}_N(u)(z) = \sum_{n_1,\cdots,n_d=1}^{N} \alpha_{n_1,\cdots,n_d} T_{n_1}(z_1) \cdots T_{n_d}(z_d)$$

where the $\alpha_{n_1,\cdots,n_d}$ are defined by

$$\alpha_{n_1,\cdots,n_d} = \prod_{i=1}^{d} \Big(\frac{\pi}{\|T_{n_i}\|_w^2 (N+1)}\Big) \sum_{m_1,\cdots,m_d=1}^{N} u(x_{m_1}, \cdots, x_{m_d}) T_{n_1}(x_{m_1}) \cdots T_{n_d}(x_{m_d}).$$

The quality of this interpolation is exactly the same than in Theorem 4.2. In dimension one, the basis functions $\mathcal{C}_k$ write

$$\mathcal{C}_k(x) = \sum_{n=0}^{N} \frac{\pi}{\|T_n\|_w^2 (N+1)} T_n(x_k) T_n(x).$$

As $T_n'(x) = \frac{-n \sin(n \arccos(x))}{\sqrt{1-x^2}}$ and because $\frac{\pi}{\|T_n\|_w^2 (N+1)} T_n(x_k) \leq \frac{2}{N+1}$, we have $|\nabla_x \mathcal{C}_k(x)|^2 \leq [\sum_{n=0}^{N} \frac{2}{N+1} \frac{n}{\sqrt{1-x^2}}]^2 = \frac{N^2}{1-x^2} \leq \frac{N^2}{1-|x|}$. Using $Z_s^2 \leq 1$, the occupation time formula [RY94] and Lemma 4.3 (which is proved in Appendix C), we finally have $V(\mathcal{C}_k, 0, x_j) \leq \int_{-1}^{1} |\nabla_x \mathcal{C}_k(y)|^2 \mathbf{E}_{x_j}(L_\tau^y(X)) dy \leq C N^2$.

Lemma 4.3. *For $D =]-1,1[$ and under* **(H2')**, *we have $\mathbf{E}_x(L_\tau^y(X)) \leq C(1-|y|)$ uniformly in $x \in D$.*

As furthermore $\mathcal{C}_k(x_j) = \delta_{k,j}$, we have

$$C(0,N) = 2 \sup_{0 \leq j \leq N} \Big(\sum_{k=0}^{N} \sqrt{V(\mathcal{C}_k, 0, x_j)}\Big)^2 \leq 2\Big(\sum_{k=0}^{N} \sqrt{C} N\Big)^2 = O(N^4).$$

This means that we need to take $M = O(N^4)$ to ensure the geometric convergence. Using the same tools, we can also prove that if $u \in H^m_w$,

$$V(u-\mathcal{P}_N(u),0,x_j) \leq C\int_{-1}^{1} |\nabla_x(u-\mathcal{P}_N(u))|^2\,(y)dy \leq C\|u-\mathcal{P}_N(u)\|^2_{H^1_w} = O(N^{-2m+4})$$

and so that the final error on the solution is a $O(N^{-m})$. We now go back to the interpolation on the multidimensional Tchebychef grid. In this case, the basis functions simply write $\mathcal{C}_{k_1,\cdots,k_d}(x_1,\cdots,x_d) = \mathcal{C}_{k_1}(x_1)\cdots\mathcal{C}_{k_d}(x_d)$. As $|\nabla_x\mathcal{C}_{k_1,\cdots,k_d}(x)|^2 \leq C\sum_{i=1}^d |\nabla_x\mathcal{C}_{k_i}(x_i)|^2$, we deduce immediately that $V(\mathcal{C}_{k_1,\cdots,k_d},0,x) = O(N^2)$ and that $C(0,N) = O(N^{2+2d})$. The error estimates, using the order m of regularity of u, are the same in dimension d than in dimension one. As the convergence is geometric and the solution is computed at N^d points with a source term constituted of N^d terms, the complexity of the algorithm is essentially $C(0,N)N^{2d}$. The upper bound on $C(0,N)$ may not be tight. We shall especially see that in all the numerical experiments $C(0,N)$ is a lot smaller than N^{2+2d} and we should also keep in mind that the spectral methods are used for very smooth solutions so that N is usually small. As a comparison, the usual spectral method requires to solve a linear system of size N^d which involves a complexity of a $O(N^{3d})$ using a direct method and of N^{2d} at each step of an iterative method. The resolution induces moreover an additional error on the solution due to the condition number of the matrix which can grow very quickly with N [BM97]. This can also make the speed of convergence of the iterative method quite slow. A big advantage of our method is that it keeps all the advantages of the Monte Carlo method in terms of parallel computing. One can for example use one processor for each of the N^d points of the grid.

4.4. **Gauss-Lobatto-Tchebychef Interpolation.** To reach a slightly better accuracy, one can also use the Gauss-Lobatto points [BM97]

$$y_k = \cos\big(\frac{N-k}{N}\pi\big),\ k = 0,1..N,$$

where the boundaries of the interval are taken as interpolation points. The coefficients (β_n) of the interpolation polynomial $\mathcal{Q}_N(u) = \sum_{n=0}^N \beta_n T_n$ satisfy the relation $\beta_n\|T_n\|^2_w = \int_{-1}^1 \mathcal{Q}_N(u)(x)T_n(x)w(x)dx$. For $n < N$, the integral equals $\frac{\pi}{N}\big(\frac{u(-1)T_n(-1)+u(1)T_n(1)}{2} + \sum_{k=1}^{N-1} u(y_k)T_n(y_k)\big)$ using the relative quadrature formula, which is exact on polynomials of degree smaller than $2N-1$. This gives the values of β_n for $n < N$. Moreover, as $T_n(1) = 1$, we have $\beta_N = u(1) - \sum_{n=0}^{N-1}\beta_n$. Hence, the basis functions write

$$\mathcal{C}_N = T_N + \sum_{n=0}^{N-1} \frac{\pi(T_n - T_N)}{2N\|T_n\|^2_w},\ \mathcal{C}_0 = \sum_{n=0}^{N-1} \frac{\pi(-1)^n(T_n - T_N)}{2N\|T_n\|^2_w}$$

and if $j \neq 0, N$

$$\mathcal{C}_j = \sum_{n=0}^{N-1} \frac{\pi T_n(y_j)}{N\|T_n\|^2_w}(T_n - T_N).$$

As for the Gauss-Tchebychef case, the d-dimensional extension on tensored domains is obtained by setting $\mathcal{C}_{k_1,\cdots,k_d}(x_1,\cdots,x_d) = \mathcal{C}_{k_1}(x_1)\cdots\mathcal{C}_{k_d}(x_d)$. This interpolation enables a better control of the derivative of u than the previous one as we can see in the following theorem [BM97] (valid in any dimension).

Theorem 4.4. *$\forall m \in \mathbf{N}^*$, $\exists c_1, c_2 > 0$ such that $\forall u \in H_w^m$, we have*

$$\|u - \mathcal{Q}_N(u)\|_w \leq c_1 N^{-m} \|u\|_{H_w^m}, \quad \|u - \mathcal{Q}_N(u)\|_{H_w^1} \leq c_2 N^{1-m} \|u\|_{H_w^m}.$$

Using the previous tools, one can easily prove that $C(0,N) = O(N^{2+2d})$ and $V(u-\mathcal{P}u, 0, x_j) = O(N^{-2m+2})$ so that the final error on the solution is a $O(N^{-m-1})$ which is compared to a $O(N^{-m})$ for the previous interpolation.

5. Numerical results

The aim of this numerical part is not to give the optimal way to solve a general problem using our algorithm. We just study various classic situations to illustrate the convergence and the accuracy of our algorithm. Different approximations and discretization schemes are tested to confirm our theoretical estimates and the great efficiency and generality of our approach.

5.1. **Poisson equation in dimension one.** Our first example is the numerical resolution of the Poisson equation in dimension one using a Monte Carlo scheme with no discretization error [GM03] and Tchebychef interpolations on either the $(x_k)_k$ or the $(y_k)_k$. The solution of this Poisson equation

$$\frac{1}{2}u'' = f \quad \text{in }]-1,1[$$

with boundary conditions $u(-1) = a, u(1) = b$ is $u(x) = a\mathbf{P}_x(W_{\tau_D} = -1) + b(1 - \mathbf{P}_x(W_{\tau_D} = -1)) - \mathbf{E}_x(\int_0^{\tau_D} f(W_s)ds)$. As $\mathbf{P}_x(W_{\tau_D} = -1) = \frac{1-x}{2}$ the contribution of the boundary conditions to the solution can be easily simulated. To achieve the contribution of the source term with no discretization error, we use the representation $\mathbf{E}_x(\int_0^{\tau_D} f(W_s)ds) = (1-x^2)\mathbf{E}(f(Y_x))$ where the density of the random variable Y_x is $\frac{1+r}{1+x}1_{-1\leq r\leq x} + \frac{1-r}{1-x}1_{x\leq r\leq 1}$.

We study the example $f(x) = (x+2)\exp(x)/2, a = -\frac{1}{e}, b = e$, so that the solution of this equation is $u(x) = x\exp(x)$. We give in the following table the error

$$e(j) = \sup_{0\leq k\leq N} |u(x_k) - u^{(j)}(x_k)|$$

as a function of the number j of steps and of the number M of sample values to compute the pointwise approximation at the x_k. Even if our algorithm is based on independent random drawings, we have observed in [GM03] that one could use low-discrepancy sequences to speed up the convergence of the algorithm. We hence use here a version of the algorithm based on Halton sequences, which is twice faster than the Monte Carlo version. The accuracy of the crude Quasi-Monte Carlo method with M sample values is given by $e(0)$. L is the number of steps until convergence for a given value of M. All the CPU times are less than one second.

N	M	L	$e(0)$	$e(5)$	$e(L)$
5	80	16	2×10^{-1}	2×10^{-2}	5×10^{-4}
7	200	30	1×10^{-1}	9×10^{-3}	4×10^{-6}
10	800	45	9×10^{-2}	2×10^{-3}	8×10^{-10}

M has to be chosen large enough with respect to N to make the algorithm converge but is significantly smaller than N^4. The error at convergence $e(L)$ corresponds exactly to the interpolation error of the interpolation polynomial $\mathcal{P}_N(u)$ of the exact solution at the Tchebychef abscissae. We now study the same example using the Gauss-Lobatto-Tchebychef interpolation.

N	M	L	$e(0)$	$e(5)$	$e(L)$
5	40	10	4×10^{-1}	1×10^{-3}	7×10^{-4}
7	100	26	2×10^{-1}	2×10^{-2}	2×10^{-6}
10	400	35	1×10^{-1}	1×10^{-2}	2×10^{-9}

We can see that we can take twice less drawings to achieve the same final accuracy for similar number of steps. This means that the geometric convergence is twice faster using this kind of interpolation, maybe because there is no error on the correction at the two boundary points. We do not notice any major difference on the final error as we could have expected.

5.2. **The bidimensional case.** We consider the Poisson equation on the square domain $D =]-1,1[^2$ with Dirichlet boundary conditions. We use an interpolation at the bidimensional Tchebychef grid, two types of discretization schemes and Monte Carlo simulations. The first one is the modified WOS method [HMG03, GM03] which can take into account the source term f. This walk goes from a sphere to another until the motion reaches the ε-absorption layer. The second one is based on the continuous Euler scheme with parameter $\triangle t$ [Gob01]. We study the equation $\frac{1}{2}\triangle u = \frac{1}{2}(4x^2+3)\exp(x^2+y)$ with Dirichlet boundary conditions chosen so that the solution of this equation is $u(x,y) = \exp(x^2+y)$. We begin with the WOS scheme taking respectively $\varepsilon_1 = 10^{-2}$ and $\varepsilon_2 = 10^{-3}$ for the absorption layer thickness and use the same notations than in the previous examples.

N	M	L	$e_1(0)$	$e_1(L)$	CPU	$e_2(0)$	$e_2(L)$	CPU
6	200	11	0.24	1×10^{-3}	1.4	0.18	1×10^{-3}	2.4
8	400	13	0.17	8×10^{-5}	6.7	0.14	7×10^{-5}	10.5
10	800	17	0.13	3×10^{-6}	28	0.14	4×10^{-6}	44

We observe a geometric convergence on both the bias and the variance up to the interpolation error of the exact solution. We do not notice any difference on the final accuracy for the two values ε_1 and ε_2. CPU times are of course smaller for ε_1 and also about eight times smaller than in [GM03] by using a recursive computation of the Tchebychef polynomials instead of their trigonometric expression. We now use the corrected Euler scheme with discretization parameters $\triangle t_1 = 0.005$ and $\triangle t_2 = 0.002$.

N	M	L	$e_1(0)$	$e_1(L)$	CPU	L	$e_2(0)$	$e_2(L)$	CPU
6	30	10	0.7	3×10^{-3}	1.4	12	0.6	2×10^{-3}	4
8	100	40	0.37	2×10^{-4}	28	10	0.38	1×10^{-4}	16
10	200	10	0.4	5	27	50	0.3	3×10^{-5}	300

We still observe a geometric convergence on both the bias and the variance except in the case $\triangle t_1 = 0.005$ and $N = 10$ because the discretization parameter is not small enough (see the condition $\rho_v < 1$). Nevertheless, we do not achieve the same accuracy at the limit. A bias due to the discretization scheme still remains. When $\triangle t$ decreases this bias decreases and the convergence is faster. Using the naive Euler scheme on the same example, the bias at the limit was twice bigger and the convergence twice slower. We can take M a lot smaller than using the WOS method to achieve the convergence.

5.3. **Parabolic problem.** We consider a regular Up and Out Call option with maturity $T = 1$, corresponding to the domain $D =]0,2[$ (actually, 0 is a natural boundary) and the Cauchy-Dirichlet boundary condition $g(t,x) = (x-1)_+$ if $t = 1$

and $g(t,x) = 1$ if $x = 2$. The dynamics is given by $X_t = x\exp(\sigma W_t - \frac{1}{2}\sigma^2 t)$. The quantity $u(0,x) = \mathbf{E}_x((X_\tau - 1)_+)$ gives the risk-neutral price of the option at time 0, when the interest rate equals 0. The solution u can be computed by a closed formula [Zha97]. The derivatives of u have singularities around $(t,x) = (1,1)$. Thus, the solution is less smooth than in previous examples and the numerical results below show that the benefit of our method is less important in that case. For the interpolation procedure, we propose a piecewise linear interpolation w.r.t. the time variable and a Tchebychef interpolation w.r.t. the space variable. The interpolation times are $(t_i = iT/(K-1))_{0\le i<K}$ and the $N+1$ Tchebychef points at each interpolation time are the (x_n) or (y_n) (on the interval $[0,2]$ instead of $[-1,1]$).

Notations relative to the errors remain the same. We first compare the accuracy of the Gauss-Tchebychef (GT) and Gauss-Lobatto-Tchebychef (GLT) interpolations, in the case $K = 5$, $N = 5$. The simulation of X can be exactly performed at discretization times $(k\triangle t)$, from which we can derive a naive approximation of τ. Here, we take $\triangle t = 0.05$ and $M = 10$ simulated paths. All the CPU times are less than one second.

	K	N	M	$\triangle t$	L	$e(0)$	$e(1)$	$e(L)$
GT	5	5	10	0.05	4	1.32×10^{-1}	7.92×10^{-2}	3.88×10^{-2}
GLT	5	5	10	0.05	4	1.72×10^{-1}	4.7×10^{-2}	1.44×10^{-2}

The GLT interpolation converges faster and the relative final error is slightly smaller. Note also that from the first iteration, the accuracy of our method compared to a crude Monte Carlo method is improved by a factor 3.6 using the GLT interpolation: this accuracy could be achieved using 13 times more crude Monte Carlo simulations. To diminish the bias, we now use Brownian bridge corrections to simulate the exit event. To have another evidence of the better convergence of the GLT interpolation, we increase the value of N to 15: the contraction constant ρ_v should increase and may become larger than 1. This is confirmed by the following result where for the GLT interpolation, the convergence still holds but not for the GT interpolation.

	K	N	M	$\triangle t$	L	$e(0)$	$e(1)$	$e(L)$
GT	5	15	10	0.01	-	1.98×10^{-1}	5.64×10^{-1}	2.7 for $L = 15$
GLT	5	15	10	0.01	10	2.13×10^{-1}	1.24×10^{-1}	5.65×10^{-3}

In view of this nice behavior, the next experiments are done using Gauss-Lobatto-Tchebychef interpolation. The figure below illustrates the geometric convergence till the interpolation error. It has been obtained with $K = 10$, $N = 20$, $M = 20$ and $\triangle t = 0.001$. The optimal choice of (K,N) will be analyzed in a future research.

6. Conclusion

We have developed and studied a sequential Monte Carlo method for the numerical solution of linear partial differential equations. This method provides a regular global approximation of this solution by combining pointwise approximations via the Feynman-Kac formula and a linear approximation on some basis functions. As the pointwise approximations are computed by means of a Monte Carlo method, statistical and discretization errors occur. We have proved the geometric reduction of these two kinds of errors up to a limit linked to the linear approximation. Numerical experiments on simple diffusion equations using various discretization schemes and different kind of approximations have confirmed this geometric convergence and the efficiency of our method. Further numerical examples should be developed

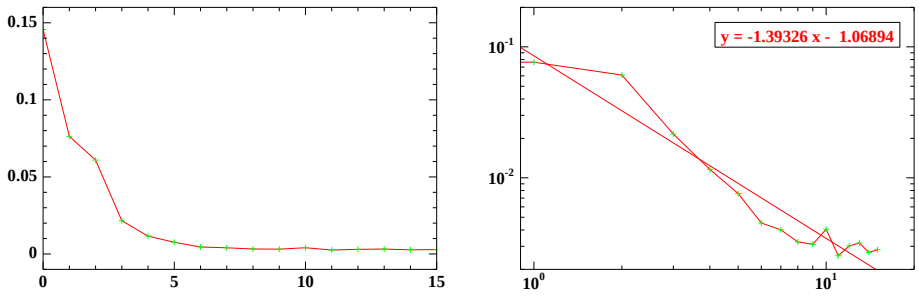

FIGURE 1. On the left, error $e(j)$ w.r.t. the number of iterations. On the right, the same in logarithmic scales.

on more complex domains, in higher dimensions or for less regular solutions. In higher dimensions, one needs to diminish the dimensional effect by making a good choice of the basis functions [Mai03]. For more complex domains or less regular solutions, new versions of the algorithm based on finite elements approximations or domain decomposition methods can certainly be developed. In all those situations, our algorithm could at least be a variance reduction tool by computing a rather poor approximation on few basis functions. As a final remark, we think that our method could replace advantageously usual deterministic methods in many situations, especially if it is used in a parallel version.

REFERENCES

[Bal95] P. Baldi. Exact asymptotics for the probability of exit from a domain and applications to simulation. *The Annals of Probability*, 23(4):1644–1670, 1995.

[BCP00] K. Baggerly, D. Cox, and R. Picard. Exponential convergence of adaptive importance sampling for Markov chains. *J. Appl. Prob.*, 37:342–358, 2000.

[Bjö96] A. Björck. *Numerical methods for least squares problems.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1996.

[BL84] A. Bensoussan and J.L. Lions. *Impulse control and quasivariational inequalities. μ.* Gauthier-Villars, Montrouge, 1984. Translated from the French by J. M. Cole.

[BM97] C. Bernardi and Y. Maday. Spectral methods. In *Handbook of numerical analysis, Vol. V*, Handb. Numer. Anal., V, pages 209–485. North-Holland, Amsterdam, 1997.

[Boo89] T.E. Booth. Zero-variance solutions for linear Monte Carlo. *Nuclear science and engineering*, 102:332–340, 1989.

[Cat92] P. Cattiaux. Stochastic calculus and degenerate boundary value problems. *Annales de l'Institut Fourier*, 42(3), 1992.

[CDL+89] M. Cessenat, R. Dautray, G. Ledanois, P.L. Lions, E. Pardoux, and R. Sentis. *Méthodes probabilistes pour les équations de la physique.* Collection CEA, Eyrolles, 1989.

[CHQZ88] C. Canuto, M.Y. Hussaini, A. Quarteroni, and T.A. Zang. *Spectral methods in fluid dynamics.* Springer Series in Computational Physics. Springer-Verlag, New York, 1988.

[CPS98] C. Costantini, B. Pacchiarotti, and F. Sartoretto. Numerical approximation for functionnals of reflecting diffusion processes. *SIAM J. Appl. Math.*, 58(1):73–102, 1998.

[DG95] D. Duffie and P. Glynn. Efficient Monte Carlo simulation of security prices. *Ann. Appl. Probab.*, 5(4):897–905, 1995.

[Fre85] M. Freidlin. *Functional integration and partial differential equations.* Annals of Mathematics Studies - Princeton University Press, 1985.

[Fri75] A. Friedman. *Stochastic differential equations and applications. Vol. 1.* New York - San Francisco - London: Academic Press, a subsidiary of Harcourt Brace Jovanovich, Publishers. XIII, 1975.

[GM92] M.G. Garroni and J.L. Menaldi. *Green functions for second order parabolic integro-differential problems*, volume 275 of *Pitman Research Notes in Mathematics Series*. Longman Scientific & Technical, Harlow, 1992.

[GM03] E. Gobet and S. Maire. A spectral Monte Carlo method for the Poisson equation. CMAP 517, Ecole Polytechnique, 2003. Submitted for publication. Presented at the IVth IMACS Seminar on Monte Carlo Methods MCM-2003 (15-19 September 2003, Berlin).

[Gob00] E. Gobet. Euler schemes for the weak approximation of killed diffusion. *Stochastic Processes and their Applications*, 87:167–197, 2000.

[Gob01] E. Gobet. Euler schemes and half-space approximation for the simulation of diffusions in a domain. *ESAIM: Probability and Statistics*, 5:261–297, 2001.

[GT83] D. Gilbarg and N.S. Trudinger. *Elliptic partial differential equations of second order*. Springer Verlag, second edition, 1983.

[Hal62] J.H. Halton. Sequential Monte Carlo. *Proc. Camb. Phil. Soc.*, 58:57–78, 1962.

[Hal70] J.H. Halton. A retrospective and prospective survey of the Monte Carlo method. *SIAM Rev.*, 12:1–63, 1970.

[HMG03] C.O. Hwang, M. Mascagni, and J.A. Given. A Feynman-Kac path-integral implementation for Poisson's equation using an h-conditioned Green's function. *Math. Comput. Simulation*, 62(3-6):347–355, 2003. 3rd IMACS Seminar on Monte Carlo Methods—MCM 2001 (Salzburg).

[Lie96] G.M. Lieberman. *Second order parabolic differential equations*. World Scientific Publishing Co. Inc., River Edge, NJ, 1996.

[Mai03] S. Maire. An iterative computation of approximations on Korobov-like spaces. *J. Comput. Appl. Math.*, 157(2):261–281, 2003.

[New94] N.J. Newton. Variance reduction for simulated diffusions. *SIAM Journal on Applied Mathematics*, 54(6):1780–1805, 1994.

[RY94] D. Revuz and M. Yor. *Continuous martingales and Brownian motion. 2nd ed.* Grundlehren der Mathematischen Wissenschaften. 293. Berlin: Springer, 1994.

[Sab91] K.K. Sabelfeld. *Monte Carlo methods in boundary value problems*. Springer Series in Computational Physics. Springer-Verlag, Berlin, 1991. Translated from the Russian.

[Zha97] P.G. Zhang. *Exotic options. A guide to second generation options*. Singapore: World Scientific., 1997.

Appendix A. Proof of Theorem 4.1

Statement *a*)**.** Since D is bounded, it is sufficient to prove the result when $D = D_R = [-R, R]^d$ for an arbitrary R. We proceed in several steps.

Step 1: τ^Δ has exponential moments, not necessarily uniformly bounded in Δ. If we use the Markov property at time $(t_k)_k$ for the Euler scheme, we get the rough estimate $\mathbf{P}(\tau^\Delta > m\Delta) \leq [\sup_{x\in\bar{D}} \mathbf{P}_x(X_\Delta^\Delta \in D)]^m$. Under **(H2')**, $\sup_{x\in\bar{D}} \mathbf{P}_x(X_\Delta^\Delta \in D) < 1$ and thus, τ^Δ has exponential moments. The point is to prove that they are uniformly bounded w.r.t. Δ, which is not clear from the computations above.

Step 2: the first moment $\sup_{x\in\bar{D}} \mathbf{E}_x(\tau^\Delta)$ is uniformly bounded w.r.t. Δ close to 0. We adapt the arguments from [Fre85], where it is proved that $\sup_{x\in\bar{D}} \mathbf{E}(\tau) < \infty$. Take $x \in D$ and set $\mathbf{1} = (1, ..., 1)^*$; the ellipticity assumption combined with Itô's formula applied to $e^{\lambda \mathbf{1}.x}$ (for λ large enough such that $-\lambda|b|_\infty d + \frac{1}{2}\lambda^2\epsilon_0 d \geq 1$) gives

$$\begin{aligned}\mathbf{E}_x(e^{\lambda \mathbf{1}.X^\Delta_{\tau^\Delta}}) &\geq e^{\lambda \mathbf{1}.x} + \mathbf{E}_x\left(\int_0^{\tau^\Delta} e^{\lambda \mathbf{1}.X_s^\Delta} ds\right) \\ &\geq e^{\lambda \mathbf{1}.x} + \min_{z\in D_{R+1}} e^{\lambda \mathbf{1}.z}\left[\mathbf{E}_x(\tau^\Delta) - \mathbf{E}_x\Big(\int_0^{\tau^\Delta} \mathbf{1}_{X_s^\Delta \notin D_{R+1}} ds\Big)\right]. \qquad \text{(A.1)}\end{aligned}$$

On the one hand, we have

$$\mathbf{E}_x(e^{\lambda \mathbf{1}.X^\Delta_{\tau^\Delta}}) \leq \sup_{z\in D_{R+1}} e^{\lambda \mathbf{1}.z} + \sum_{k=0}^{\infty} \mathbf{E}_x\big[\mathbf{1}_{t_k<\tau^D}\mathbf{1}_{X^\Delta_{t_{k+1}}\notin D_{R+1}} e^{\lambda \mathbf{1}.X^\Delta_{t_{k+1}}}\big].$$

Standard large deviations estimates (see Lemma 4.1. in [Gob00]) give $\mathbf{E}(e^{\lambda \mathbf{1}.X^\Delta_{t_{k+1}}}\mathbf{1}_{X^\Delta_{t_{k+1}}\notin D_{R+1}}|\mathcal{F}_{t_k}) \leq C(\lambda)e^{-c/\Delta}$ for some constants $c>0$ and $C(\lambda)$ uniform w.r.t. $\Delta \leq 1$ and $X^\Delta_{t_k}\in D_R$. For Δ small enough such that $C(\lambda)e^{-c/\Delta} \leq \frac{1}{3}\min_{z\in D_{R+1}} e^{\lambda \mathbf{1}.z}$, we obtain

$$\mathbf{E}_x(e^{\lambda \mathbf{1}.X^\Delta_{\tau^\Delta}}) \leq \sup_{z\in D_{R+1}} e^{\lambda \mathbf{1}.z} + \mathbf{E}_x(\tau^\Delta)\,\frac{1}{3}\min_{z\in D_{R+1}} e^{\lambda \mathbf{1}.z}.$$

On the other hand, from Fubini's theorem, we have $\mathbf{E}_x(\int_0^{\tau^\Delta}\mathbf{1}_{X^\Delta_s\notin D_{R+1}}ds) = \int_0^\infty \mathbf{P}_x(\phi(s) < \tau^\Delta; X^\Delta_s \notin D_{R+1})ds$. The previous arguments give $\mathbf{E}_x(\int_0^{\tau^\Delta}\mathbf{1}_{X^\Delta_s\notin D_{R+1}}ds) \leq \int_0^\infty \mathbf{P}_x(\phi(s)<\tau^\Delta)Ce^{-c/\Delta} \leq \frac{1}{3}\mathbf{E}_x\tau^\Delta$ for Δ small enough. Plugging all theses estimates into (A.1), we get

$$\sup_{z\in D_{R+1}} e^{\lambda \mathbf{1}.z} \geq e^{\lambda \mathbf{1}.x} + \mathbf{E}_x(\tau^\Delta)\,\frac{1}{3}\min_{z\in D_{R+1}} e^{\lambda \mathbf{1}.z}$$

uniformly in $x\in D$, for Δ small enough. This proves our assertion.
Step 3: the p-th moment $\sup_{x\in\bar D}\mathbf{E}_x([\tau^\Delta]^p)$ $(p\geq 1)$ is uniformly bounded w.r.t. Δ close to 0. This is a standard consequence of $\sup_{x\in\bar D}\mathbf{E}_x(\tau^\Delta)$ and of the Markov property at times $(k\Delta)_k$. We refer to Section 3.3 in [Fre85] for a proof in the diffusion case, which can be adapted to the discrete Euler scheme in a straightforward way. Statement *a*) is proved.

Statement *b*)**.** In view of the uniform integrability conditions a), it is enough to prove that τ^Δ converges in probability to τ, which follows from the weak convergence of (τ^Δ,τ) to (τ,τ) (stable convergence). Thus, we aim at showing that for any s_1,s_2, we have

$$\lim_{\Delta\to 0}\mathbf{P}_x(\tau^\Delta\leq s_1,\tau\leq s_2) = \mathbf{P}_x(\tau\leq s_1,\tau\leq s_2). \tag{A.2}$$

We introduce the signed distance to ∂D, defined by $F(x)=d(x,\partial D)$ if $x\in D$ and $F(x)=-d(x,\partial D)$ if $x\notin D$. Without additional regularity on D, F is at least a Lipschitz continuous function. Note that $\{\tau^\Delta\leq s_1\}=\{\inf_{t\leq s_1}F(X^\Delta_{\phi(t)})\leq 0\}$ and $\{\tau\leq s_1\}=\{\inf_{t\leq s_1}F(X_t)\leq 0\}$. From the *a.s.* uniform convergence of $(X^\Delta_{\phi(t)})_t$ to $(X_t)_t$ on $[0,s_1]$ (see [Gob00] for instance), we have $\lim_{\Delta\to 0}\inf_{t\leq s_1}F(X^\Delta_{\phi(t)}) = \inf_{t\leq s_1}F(X_t)$ *a.s.* Thus, (A.2) holds true if $0=\mathbf{P}(\inf_{t\leq s_1}F(X_t)=0)$, which writes using the strong Markov property

$$0=\mathbf{E}(\mathbf{1}_{\tau<s_1}\mathbf{P}_{X_\tau}(\forall t\leq s_1-\tau : X_t\in\bar D)) + \mathbf{P}(\tau=s_1). \tag{A.3}$$

In fact, for any $r>0$ and $x\in\partial D$, under **(H2')** we have $\mathbf{P}_x(\forall t\leq r : X_t\in\bar D)\leq \mathbf{P}_x(X_r\in\bar D)<1$ (in [Gob00], see inequality (68) and the comments before Remark 5.1). Thus, by the Blumenthal Zero-One law, the probability $\mathbf{P}_x(\forall t\leq r : X_t\in\bar D)$ must be equal to 0. Hence, (A.3) is reduced to $\mathbf{P}(\tau=s_1)=0$. This equality is true except for the countable number of points of discontinuity of $s_1\mapsto\mathbf{P}(\tau\leq s_1)$, which is enough to derive (A.2) for any s_1.

Statements *c)* **and** *d)*. Both statements easily follow from the *a.s.* uniform convergence of X^{Δ} to X on compact sets, from the uniform integrability in *a)* and from the convergence *b)*.

Appendix B. A technical lemma

Lemma B.1. *Assume* **(H2')** *and that $\hat{u}$ is the $C^0(\bar{D}) \cap C^2(D)$-solution of $L_0\hat{u} - c\hat{u} = \hat{f}$ in D and $\hat{u} = \hat{g}$ on ∂D, where $\hat{f}$ and $\hat{g}$ are bounded continuous functions. Then*

$$V(\hat{u}, 0, x) = \mathbf{E}_x \left[\int_0^{\tau} Z_s^2 \, |\nabla_x \hat{u} \, \sigma|^2(X_s) ds \right] < +\infty.$$

Proof. The technical difficulty comes from the fact that $\hat{u}$ may have derivatives exploding near the boundary. To circumvent this problem, set $D_\varepsilon = \{x \in D : d(x, \partial D) > \varepsilon\}$ for $\varepsilon > 0$ and denote by τ_ε the associated exit time. By standard interior estimates [GT83], $\hat{u}$ has a bounded gradient in D_ε. Furthermore, it is straightforward to see that $\tau_\varepsilon \uparrow \tau$ *a.s.* as $\varepsilon \downarrow 0$. An application of Itô's formula gives

$$\hat{u}(X_{\tau_\varepsilon}) Z_{\tau_\varepsilon} + \int_0^{\tau_\varepsilon} Z_s(-L\hat{u} + c\hat{u})(X_s) ds = \hat{u}(x) + \int_0^{\tau_\varepsilon} Z_s \, [\nabla_x \hat{u} \, \sigma](X_s) dW_s.$$

Owing to the localization in D_ε, it is easy to see that $\mathbf{E}_x\big[\int_0^{\tau_\varepsilon} Z_s^2 \, |\nabla_x \hat{u} \, \sigma|^2(X_s) ds\big] < \infty$. Hence, by the isometry property, we obtain

$$\mathbf{E}_x\Big[\int_0^{\tau_\varepsilon} Z_s^2 \, |\nabla_x \hat{u} \, \sigma|^2(X_s) ds\Big] = \mathbf{V}\mathrm{ar}_x\Big[\hat{u}(X_{\tau_\varepsilon}) Z_{\tau_\varepsilon} + \int_0^{\tau_\varepsilon} Z_s(-L\hat{u} + c\hat{u})(X_s) ds\Big].$$

Take the limit when ε goes to 0: the l.h.s. converges using the monotone convergence theorem and the r.h.s. using the dominated convergence theorem. The limit writes $\mathbf{E}_x\big[\int_0^{\tau} Z_s^2 \, |\nabla_x \hat{u} \, \sigma|^2(X_s) ds\big] = V(\hat{u}, 0, x)$, which is our statement. □

Appendix C. Proof of Lemma 4.3

We can assume $y \geq 0$. Tanaka's formula [RY94] yields

$$\frac{1}{2}\mathbf{E}_x(L_\tau^y(X)) = \mathbf{E}_x(X_\tau - y)_+ - (x - y)_+ - \mathbf{E}_x\Big(\int_0^{\tau} b(X_s) \mathbf{1}_{X_s \geq y} ds\Big).$$

Hence, we get $\mathbf{E}_x(L_\tau^y(X)) \leq C$ uniformly in x, y. Using the occupation time formula in the equality above and the previous uniform estimate, we obtain $\frac{1}{2}\mathbf{E}_x(L_\tau^y(X)) \leq (1 - y) + C \int_y^1 \mathbf{E}_x(L_\tau^z(X)) dz \leq C(1 - y)$. If $y < 0$, same arguments apply. □

ECOLE POLYTECHNIQUE - CENTRE DE MATHÉMATIQUES APPLIQUÉES - 91128 PALAISEAU CEDEX - FRANCE

E-mail address: emmanuel.gobet@polytechnique.fr

ISITV - UNIVERSITÉ DE TOULON ET DU VAR - AVENUE G. POMPIDOU, BP56 - 83262 LA VALETTE DU VAR CEDEX - FRANCE

E-mail address: maire@univ-tln.fr