Mancini, S., Bernal, F., Acebrón, J. A. (2016). An Efficient Algorithm for Accelerating Monte Carlo Approximations of the Solution to Boundary Value Problems. *Journal of Scientific Computing*, 66(2), 577–597.

An Efficient Algorithm for Accelerating Monte Carlo Approximations of the Solution to Boundary Value Problems.

Sara Mancini [*]   Francisco Bernal [†]   Juan A. Acebrón[‡§]

April 6, 2015

Abstract

The numerical approximation of boundary value problems by means of a probabilistic representations often has the drawback that the Monte Carlo estimate of the solution is substantially biased due to the presence of the domain boundary. We introduce a scheme, which we have called the leading-term Monte Carlo regression, which seeks to remove that bias by replacing a 'cloud' of Monte Carlo estimates –carried out at different discretization levels– for the usual single Monte Carlo estimate. The practical result of our scheme is an acceleration of the Monte Carlo method. Theoretical analysis of the proposed scheme, confirmed by numerical experiments, shows that the achieved speedup can be well over 100.

Keywords: Monte Carlo method, Romberg extrapolation, bounded diffusion, Feynman-Kac formula, first exit time, parallel computing.

1   Introduction

For a bounded domain $\Omega \subset \mathbb{R}^d$ of dimension $d \geq 2$, we consider the evaluation of expected values of the form

$$\mathbb{E}[\,\phi(\mathbf{X}_\tau)\,] := \mathbb{E}\Big[\,g(\mathbf{X}_\tau)e^{\int_0^\tau c(\mathbf{X}_s)ds} + \int_0^\tau f(\mathbf{X}_s)e^{\int_0^s c(\mathbf{X}_q)dq}ds\,\Big], \tag{1}$$

where $g : \partial\Omega \mapsto \mathbb{R}$, $f : \Omega \mapsto \mathbb{R}$, and $c : \Omega \mapsto (-\infty, 0]$ are functions regular enough, and $\mathbf{X}_\tau \in \partial\Omega$ is the solution of the following stochastic differential equation (SDE):

$$d\mathbf{X}_t(t > 0) = \begin{cases} b(\mathbf{X}_t)dt + \sigma(\mathbf{X}_t)d\mathbf{W}_t, & 0 < t < \tau, \\ 0, & t \geq \tau, \end{cases} \tag{2}$$

$$\mathbf{X}_t(t = 0) = \mathbf{x_0}.$$

[*]Dipartimento di Matematica 'Federigo Enriques', Università degli Studi di Milano. Via Cesare Saldini 50, 20133 Milan, Italy. (`sara.mancini1@studenti.unimi.it`)

[†]Center for Mathematics and its Applications (CEMAT), Department of Mathematics, Instituto Superior Técnico. Av. Rovisco Pais 1049-001 Lisbon, Portugal. (`francisco.bernal@ist.utl.pt`)

[‡]ISCTE - Instituto Universitário de Lisboa Departamento de Ciências e Tecnologias de Informação. Av. das Forças Armadas 1649-026 Lisbon, Portugal. (`juan.acebron@ist.utl.pt`)

[§]INESC-ID\IST, TU Lisbon. Rua Alves Redol 9, 1000-029 Lisbon, Portugal.

1

In (2), $\mathbf{x_0} \in \Omega \backslash \partial\Omega$, $b : \mathbb{R}^d \mapsto \mathbb{R}^d$ is called the drift, $\sigma : \mathbb{R}^d \mapsto \mathbb{R}^{d \times d}$ is called the diffusion matrix, and $\mathbf{W}_t$ is a standard Wiener process in $\mathbb{R}^d$. Then, $\mathbf{X}_t$ is a continuous diffusion which starts at point $\mathbf{x_0}$ at $t = 0$ and stops upon hitting the boundary for the first time at $t = \tau := \inf\{s : \mathbf{X}_s \in \partial\Omega\}$. The quantities τ and $\mathbf{X}_\tau$ are called first-exit time and first-exit point, respectively. Expectations like (1) arise in the probabilistic representation of boundary value problems (BVPs). For instance, the solution $u : \Omega \subset \mathbb{R}^d \mapsto \mathbb{R}$ of the linear elliptic partial differential equation (PDE) with Dirichlet boundary conditions (BCs)

$$\sum_{i=1}^{d} \sum_{j=1}^{d} a_{ij} \frac{\partial^2 u}{\partial x_i \partial x_j} + \sum_{k=1}^{d} b_k \frac{\partial u}{\partial x_k} + cu + f = 0,$$
$$u(\mathbf{x} \in \partial\Omega) = g, \tag{3}$$

with $\sigma\sigma^T/2 = [a_{ij}]$ and $c \leq 0$, can be expressed as (1) via Dynkin's formula (assuming the appropriate regularity conditions). There exist similar representations for the parabolic case (the well-known Feynman-Kac formula) and for other BCs [5]. Although much of the work in this paper could be extended to them, we will restrict our analysis to the pair (1) and (3).

In practice, the expected value (1) is usually estimated with a Monte Carlo (MC) method:

$$\mathbb{E}[\,\phi(\mathbf{X}_\tau)\,] \approx \frac{1}{N} \sum_{j=1}^{N} \phi^h(\mathbf{X}_{\tau_h}^j), \tag{4}$$

which involves $N >> 1$ independent realizations of a discrete approximation to (2), and (ϕ^h, τ_h) are the discrete counterparts to (ϕ, τ). Such discrete approximation (which can be thought of as the polygonal trajectory of a 'particle') is performed by numerically integrating (2) with a numerical scheme Ξ_h (such as the Euler-Maruyama method), and a time step $h > 0$ (which we will assume constant). Neglecting the correlations in any digital random numbers generator, the total error of the MC approximation (4) to (1) can be bounded as [1]

$$\epsilon_{\mathrm{MC}}(N, h) := \left| \mathbb{E}[\,\phi(\mathbf{X}_\tau)\,] - \frac{1}{N} \sum_{j=1}^{N} \phi^h(\mathbf{X}_{\tau_h}^j) \right| \leq \epsilon_N(N, h) + \epsilon_h(h). \tag{5}$$

In (5), ϵ_N is the statistical error, which results from the substitution of the expected value in (1) by the arithmetic mean. According to the central limit theorem [15], as $h \to 0$ and $N \to \infty$, ϵ_N tends to be normally distributed, and to be smaller than $q = 1, 2$, and 3 times $\sqrt{\mathrm{Var}\big(\phi^h(\mathbf{X}_{\tau_h})\big)/N}$ with probability (confidence intervals Γ_q) 68.3%, 95.5%, and 99.7%, respectively. However, at finite time steps $h > 0$, ϵ_{MC} does not vanish as $N \to \infty$, i.e. the discretization of (2) turns (4) into a biased estimator of (1). But, importantly, the fact that the domain is bounded induces larger errors in the numerical computation of (2) than would be the case for a PDE or a SDE in an unbounded domain. They stem from the ambiguity in determining the first intersection of a discrete stochastic trajectory with the boundary, and may lead to poor approximations of τ (by τ_h) and $\mathbf{X}_\tau$ (by $\mathbf{X}_{\tau_h}$). We refer to the latter as the first-exit error, and call ϵ_h the bias.

We now examine more closely the structure of the bias. In absence of boundaries, ϵ_h is the weak error of the discretization scheme used, and is typically $O(h^r)$, $r \geq 1$. An important case is that of the Euler-Maruyama method, for which $\epsilon_h = O(h)$ [11]. This was further refined by Talay and Tubaro, who showed that the second term was $O(h^2)$ [18]. The availability of the second term in the error expansion enabled them to combine two discretizations of the SDE with different values of h to yield a second-order-accurate integrator. On the other hand, the presence of boundaries renders those results invalid, due to the larger contribution in ϵ_h posed by the first-exit error. The simplest way to modify the Euler-Maruyama scheme to deal with boundary absorption (as needed in (2)) is to stop the discrete trajectory $\mathbf{X}_0 = \mathbf{x_0}, \mathbf{X}_1, \mathbf{X}_2, \ldots$ between the last iterate inside $\Omega \backslash \partial\Omega$ (say the kth), and the first one outside. Then $kh \leq \tau_h \leq (k+1)h$, and $\mathbf{X}_{\tau_h}$ for that particular trajectory must be somehow interpolated between $\mathbf{X}_k$ and $\mathbf{X}_{k+1}$; for instance by taking the projection of the latter onto $\partial\Omega$. This procedure leads to $\epsilon_h = O(\sqrt{h})$, which is much worse than for free diffusions. This convergence rate was proved rigorously by Gobet and Menozzi for parabolic BVPs in [7] and for elliptic BVPs in [8]. Still, no error expansion in powers of h is available. In the last two decades, a number of improved schemes have been devised that have a smaller bias than that of Euler-Maruyama, generally at the cost of a computationally more involved time-stepping. Higher error rates than $O(\sqrt{h})$ have been experimentally observed in the schemes proposed in [13, 10, 3, 8], while $O(h)$ bias has been rigorously proved–under the corresponding technical assumptions–for the schemes in [17, 14]. However, the lack of an error expansion for them (or at least, the two lowest powers of h) precludes the possibility of Romberg extrapolation such as in the Talay-Tubaro method. And anyways, we stress that the Talay-Tubaro method pertains the expected values, rather than Monte Carlo simulations inevitably contaminated with statistical noise.

Even in the much better situation when $\epsilon_h = O(h)$, the interplay between the statistical error and the bias in (5) may lead to very time-consuming simulations. The reason is that, for a set accuracy A (the largest tolerable total error for a given confidence interval Γ_q), both h and N must be set small and large enough, respectively, so that

$$\epsilon_N, \epsilon_h \leq A. \tag{6}$$

Thus, not only does a better (i.e. smaller) accuracy target involve more trajectories overall, but also each of them will take longer due to the finer discretization. Seeking to accelerate the simulations, we propose in this paper a novel method of approximating (1). The idea is to exploit the fact that the leading power in ϵ_h is theoretically known for many triplets SDE (2)/ integrator Ξ_h/ bounded domain Ω, in order to extrapolate to $h = 0$, where the MC approximation is unbiased. More specifically, we propose to carry out a 'cloud' of $n \geq 2$ independent, but relatively 'noisy', MC simulations at n equispaced values of h, and to fit the coefficient of the leading term of ϵ_h from them - by means of a regression. (The reason why we have chosen equispaced time steps is that this leads to analytical formulas which facilitate the theoretical study). We have called this algorithm the leading-term Monte Carlo regression (LTMCR).

Figure (1) sketches the main features of LTMCR. The cloud used for fitting
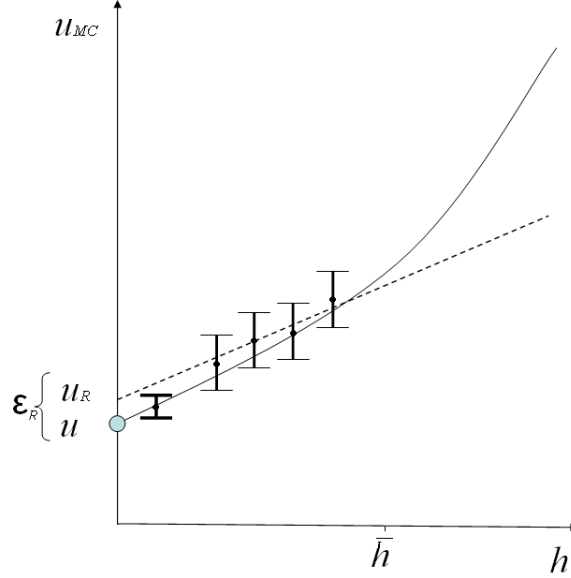
Figure 1: Intuitive idea of the leading-term Monte Carlo regression (LTMCR).

purposes consists of the four MC simulations depicted on the right. They have an ever smaller bias as their h approaches 0; on the other hand, their error bars (statistical errors) are comparatively large, and rather independent of h. The solid curve plotting ϵ_h passes through each of the error bars with a probability Γ_q. (Notice that this curve is not known, and only plotted for the sake of illustration.) In the hypothetical problem in Figure (1), the leading term is linear, and the linear behaviour dominates for $0 < h \lesssim \bar{h}$. The LTMCR yields an approximation to (1) given by u_R. The geometric interpretation of the LTMCR solution is thus the intersection of the curve fitting the data with the vertical axis. We stress the fact that the leading term of ϵ_h (in this case, one) is assumed known from the integrator Ξ_h used to produce the cloud data. The error

$$\epsilon_R := u - u_R \tag{7}$$

is a stochastic variable whose precise distribution will be discussed in Section 2. Notwithstanding the details, it is expected that ϵ_R will decrease as the error bars become narrower, n increases, and the cloud of individual MC simulations is clustered towards 0. However, for the LTMCR to be useful, the sole characterization of the error is inadequate. Clearly, it must also be faster than the single, competing MC simulation, which yields the same accuracy A under the same confidence interval Γ_q. We will call that single MC simulation the reference simulation (the leftmost one on Figure (1)). Defining the computational cost of the reference simulation and of the LTMCR as C_{ref} and C_R, respectively, the potential of the LTMCR boils down to the speedup (8) being greater than one.

$$S = \left.\frac{C_{\text{ref}}}{C_R}\right|_{A,\Gamma_q} \tag{8}$$

4

The multilevel method by Giles [6] is another method which also uses variable time steps in order to accelerate MC simulations. Unlike the LTMCR it does so by reducing the variance, rather than by extrapolation. It was initially designed for unbounded diffusions, although has recently been extended to bounded Brownian motion [9]. The LTMCR differs from Talay-Tubaro extrapolation in the fact that the LTMCR includes the statistical error and requires just the leading term of the bias expansion. In all three methods, a threshold time step below which the respective asymptotic error expansions are valid (the parameter $\bar{h}$ in the LTMCR) is assumed *a priori*.

The remainder of the paper is organized as follows. In Section 2, for simplicity, the case when $\epsilon_h = O(h)$ is considered first. Under some technical assumptions, the distribution of the error of the LTMCR is analyzed and characterized according to confidence intervals, as well as its cost. The optimal speedup is then derived by solving a constrained minimization problem. In Section 3, those results are carried over to the general (nonlinear) case, $\epsilon_h = O(h^r)$. Section 4 validates the previous analytical results on two numerical experiments, which use a different integrator Ξ_h each. Some final remarks are given in Section 5.

2 Linear bias

We first analyze the case in which the bias is asymptotically a linear function of the time step:

$$\epsilon_h = \beta h, \qquad \text{if } 0 < h \le \bar{h}, \tag{9}$$

for some $\bar{h} > 0$. This can happen, for instance, in the integrators Ξ_h by Milstein [14]. We consider n uniformly spaced time steps $\{h_i\}_{i=1}^n$, such that $0 < h_1 < h_2 < \ldots < h_n \le \bar{h}$. Let us call the fixed point at which we want to approximate the solution $\mathbf{x}_0$. For each value h_i we compute $u_0 = u(\mathbf{x}_0)$ of (3) via the estimator (1) with $N \gg 1$. On this 'cloud' of data we perform a regression analysis according to the model:

$$u_{\mathrm{MC}}(N, h_i) \sim u_0 + \epsilon_h(h_i) + \epsilon_N(N, h_i) = \alpha + \beta h_i + \delta_i, \tag{10}$$

where $a \sim b$ means that a is drawn from distribution b. The 'noise' δ_i, ($i = 1, \ldots, n$) is the statistical error of each MC simulation. According to the central limit theorem it is asymptotically normally distributed:

$$\delta_i \underset{N \to \infty}{\sim} N(0, \sigma_i^2). \tag{11}$$

It is also h-dependent because

$$\sigma_i^2 := \mathrm{Var}\Big(\phi^{h_i}(\mathbf{X}_{\tau_{h_i}})\Big)/N. \tag{12}$$

In practice, for each timestep h_i, we defined the value of σ_i^2 as the empirical variance of the MC mean estimator divided by the number of trajectories used in the simulation, being the variance computed using the sampling data. Moreover, as long as the n MC simulations in the cloud are independent,

$$cov(\delta_i, \delta_j) = 0 \qquad \text{if } i \ne j. \tag{13}$$

Defining the weights

$$w_i = 1/\sigma_i^2 \tag{14}$$

the coefficients (α, β) of the fit (10) can be approximated by the estimators $(\hat{\alpha}_w, \hat{\beta}_w)$ obtained using a classical weighted linear least squares technique [4] to minimize $\sum_{i=1}^{n} \sigma_i^{-2} (u_{\mathrm{MC}}(h_i, N) - \alpha - \beta h_i)^2$. We thus have

$$(\hat{\alpha}_w, \hat{\beta}_w)^{\mathsf{T}} = (X^{\mathsf{T}} W X)^{-1} X^{\mathsf{T}} W U \tag{15}$$

where

$$X = \begin{pmatrix} 1 & 1 & \dots & 1 \\ h_1 & h_2 & \dots & h_n \end{pmatrix}^T, \tag{16}$$

$$W = \begin{pmatrix} w_1 & 0 & \dots & 0 \\ 0 & w_2 & \dots & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & \dots & w_n \end{pmatrix}, \tag{17}$$

$$U = \left(u_{\mathrm{MC}}(N, h_1), u_{\mathrm{MC}}(N, h_2), \dots, u_{\mathrm{MC}}(N, h_n) \right)^T. \tag{18}$$

Under conditions (11) and (13), the estimators of a linear regression are normally distributed. Since the weighted estimators $(\hat{\alpha}_w, \hat{\beta}_w)$ are an affine combination of those ones, then they are normally distributed too; in particular, the error $u_0 - \hat{\alpha}_w$ is

$$u_0 - \hat{\alpha}_w \sim N\left(0, \mathrm{Var}(\hat{\alpha}_w) \right), \tag{19}$$

where

$$\mathrm{Var}(\hat{\alpha}_w) = \frac{1}{\sum w_i} + \frac{(\bar{h}^w)^2}{\sum w_i (h_i - \bar{h}^w)^2} \tag{20}$$

and $\bar{h}^w = \sum w_i h_i / \sum w_i$ is the weighted mean of the $\{h_i\}$. (See [15], chapter 10.) Therefore, the error of the regression can be expressed in terms of the same confidence intervals Γ_q as a plain MC simulation by simply taking q standard deviations as error bar for it, where q is the quantile of a normal distribution. For Γ_q let us define the error of the LTMCR approximation as

$$\epsilon_{\mathrm{R}} := q \sqrt{\mathrm{Var}(\hat{\alpha}_w)}. \tag{21}$$

Note that, contrary to a plain MC simulation, this error is purely probabilistic and unbiased. Let us define

$$v_{\max} := \max_{h_1 \leq h \leq \bar{h}} \mathrm{Var}\left(\phi^h(\mathbf{X}_{\tau_h}) \right) \approx \max_{1 \leq i \leq n} \mathrm{Var}\left(\phi^{h_i}(\mathbf{X}_{\tau_h}) \right) \tag{22}$$

and let $\hat{\alpha}_{\max}$ be the estimator of α if every point of the sample had the largest variance $\sigma_{\max}^2 = v_{\max}/N$. Proposition 1 shows that the variance of the estimator $\hat{\alpha}_w$ is bounded above. Since we are not aware of the existence of this result in the literature, we give also a proof of it.

Proposition 1. *Let $\hat{\alpha}_w$ and $\hat{\alpha}_{\max}$ be the estimators for the exact solution, obtained respectively from the weighted regression and the regression with the largest variance $\sigma_{\max}^2$ for all the n points. It holds:*

$$Var(\hat{\alpha}_w) \leq Var(\hat{\alpha}_{\max}). \tag{23}$$

Proof. First, we show that the variance is decreasing with respect to the weights. Let us define $\mathbf{w} := (w_1, \ldots, w_n)$ and $P = P(\mathbf{w}) := \sum_{i=1}^{n} w_i$. Then by definition

$$P\bar{h}^w = \sum_{i=1}^{n} w_i h_i \tag{24}$$

and by (20)

$$\text{Var}(\hat{\alpha}_w) = \frac{1}{P} + \frac{(\bar{h}^w)^2}{\sum_{i=1}^{n} w_i(h_i - \bar{h}^w)^2} = \frac{\sum_{i=1}^{n} w_i h_i^2}{P\sum_{i=1}^{n} w_i(h_i - \bar{h}^w)^2}, \tag{25}$$

where we have used the fact that

$$\sum_{i=1}^{n} w_i(h_i - \bar{h}^w)^2 = \sum_{i=1}^{n} w_i h_i^2 - P(\bar{h}^w)^2. \tag{26}$$

Taking logarithms and computing the partial derivatives with respect to w_j, for every $j = 1, \ldots, n$,

$$\frac{\partial}{\partial w_j} \log\left(\text{Var}(\hat{\alpha}_w)\right) = -\frac{\partial}{\partial w_j} \log(P) + \frac{\partial}{\partial w_j} \log\left(\sum_{i=1}^{n} w_i h_i^2\right) - \frac{\partial}{\partial w_j} \log\left(\sum_{i=1}^{n} w_i(h_i - \bar{h}^w)^2\right). \tag{27}$$

The three terms can be evaluated to yield

$$\frac{\partial}{\partial w_j} \log(P) = \frac{1}{P}, \tag{28}$$

$$\frac{\partial}{\partial w_j} \log\left(\sum_{i=1}^{n} w_i h_i^2\right) = \frac{h_j^2}{\sum_{i=1}^{n} w_i h_i^2}, \tag{29}$$

$$\frac{\partial}{\partial w_j} \log\left(\sum_{i=1}^{n} w_i(h_i - \bar{h}^w)^2\right) = \frac{(h_j - \bar{h}^w)^2}{\sum_{i=1}^{n} w_i(h_i - \bar{h}^w)^2}, \tag{30}$$

where we have used $\sum_{i=1}^{n} w_i(h_i - \bar{h}^w) = 0$. Substituting these expressions into (27) and summing them, we have a fraction whose denominator is the product of positive factors. The numerator $\mathcal{N}$ is

$$\mathcal{N} = -\sum w_i h_i^2 \sum w_i(h_i - \bar{h}^w)^2 + h_j^2 P \sum w_i(h_i - \bar{h}^w)^2 - P(h_j - \bar{h}^w)^2 \sum w_i h_i^2$$
$$= -\left(\sum w_i h_i^2 - P h_j \bar{h}^w\right)^2. \tag{31}$$

This means that the logarithm of the variance is a non-increasing function of the weights, which takes the maximum value in $w_{\min}$. By the monotonicity of the logarithm and the positivity of the variance, the function $\text{Var}(\hat{\alpha}_w)$ behaves in the same way. Therefore, $\text{Var}(\hat{\alpha}_w) \leq \text{Var}(\hat{\alpha}_{w_{\min}}) = \text{Var}(\hat{\alpha}_{\max})$, since $w_{\min} = 1/\sigma_{\max}^2$. $\square$

Remark 1. The proof considers the variance as a function of the weights and not of the time steps $\{h_i\}$ - meaning that (23) also holds if we substitute $\{h_i\}$ with different variables $\{\xi_i\}$, as we need in the general case (see Section 3).

Now, by (23), the definition of $\sigma_{\max}^2$, $\text{Var}(\hat{\alpha}_{\max}) = v_{\max}\left[\frac{1}{n} + \frac{\bar{h}^2}{\sum(h_i - \bar{h})^2}\right]$ and the fact that the points in the sample are equispaced, we can derive an upper bound related to the Gaussian quantile q for ϵ_R

$$\epsilon_R \leq \epsilon_{\max} := q\sqrt{\frac{v_{\max}}{N}}\sqrt{\frac{1}{n} + 3l^2\frac{n-1}{n(n+1)}}, \tag{32}$$

where

$$L := h_n - h_1, \qquad l := (h_n + h_1)/L \tag{33}$$

are positive constants.

2.1 Cost of the regression.

Having characterized the error of the LTMCR (for the linear bias case), we estimate its computational cost, C_R. Since the cost of the regression analysis is negligible, C_R is equal to the aggregate cost of all the MC simulations involved. The computational cost of (4) is

$$C_{MC} \propto N\bar{k}, \tag{34}$$

where $\bar{k}$ is the average number of steps before hitting the boundary $\partial\Omega$ from $\mathbf{x}_0$. Let us assume that asymptotically

$$\bar{k} \propto 1/h, \tag{35}$$

which is the most common situation (see for instance [14], chapter 5). Then,

$$C_{MC}(N, h) = \rho N/h, \tag{36}$$

with $\rho \in \mathbb{R}_+$, which for a given BVP depends on the integrator Ξ_h. Using $h_i = h_1 + (h_n - h_1)(i-1)/(n-1)$, the cost of the LTMCR is

$$C_R = \sum_{i=1}^{n} C_{MC}(N, h_i(n)) = \rho\sum_{i=1}^{n}\frac{N}{h_i(n)} = \rho\frac{N}{h_1}\sum_{i=1}^{n}\left(1 + \frac{i-1}{n-1}\frac{L}{h_1}\right)^{-1}. \tag{37}$$

We now seek to optimize the cloud parameters for a given accuracy A, i.e. the constrained minimum (N^*, n^*) of C_R subject to the restriction $\epsilon_R \leq A$:

$$(N^*, n^*) = \underset{\left\{\begin{matrix}\epsilon_{\max}(N,n) \leq A \\ n \geq 2 \\ N \geq N_{\min}\end{matrix}\right\}}{\arg\min} C_R(N, n). \tag{38}$$

$N_{\min}$ is the minimum number of simulations per point so that the statistical error is small enough (discussed later on).

2.2 Approximations leading to an analytic solution.

In order to solve the constrained minimization problem (38), we will approximate sums like (37) with integrals:

$$\sum_{i=1}^{n}\xi_i^t = h_1^{-r}\sum_{i=1}^{n}\left[1 + \frac{i-1}{n-1}\frac{L}{h_1}\right]^{-r} \approx 1 + \int_{y=1}^{y=n}\left[1 + \frac{y-1}{n-1}\frac{L}{h_1}\right]^{-r}dy. \tag{39}$$

Then,

$$C_R(N, n) = \rho \frac{N}{h_1} \sum_{i=1}^{n} \left(1 + \frac{i-1}{n-1} \frac{L}{h_1}\right)^{-1} \approx \rho \frac{N}{h_1} \left[1 + \frac{h_1(n-1)}{L} \log\left(1 + \frac{L}{h_1}\right)\right]. \quad (40)$$

We may rewrite the approximated cost C_R

$$C_R(N, n) \approx aN\big(1 + b(n-1)\big), \quad \text{with } a > 0, \quad 0 < b < 1, \quad (41)$$

where

$$a := \frac{\rho}{h_1}, \quad (42)$$

$$b := \frac{h_1}{L} \log\left(1 + \frac{L}{h_1}\right). \quad (43)$$

Regarding the error function (32), we just need to check its behavior when n becomes large; by assuming n large enough so that $\frac{n-1}{n+1} \approx 1$, we have

$$\epsilon_{max}(N, n) \approx \frac{c}{\sqrt{nN}}, \quad (44)$$

with

$$c := q \sqrt{v_{max}(1 + 3l^2)}. \quad (45)$$

2.3 Theoretical feasibility and speedup.

With the purpose of finding an analytical solution of problem (38), when n is large enough we can replace $\epsilon_R(N, n)$ and $C_R(N, n)$ by the approximations (44) and (41), which are differentiable with respect to n and N. Notice that the constants $a, b, c \in \mathbb{R}_+$ in (44) and (41) are strictly positive, and $b < 1$. The solution (N^*, n^*) of (38) is such that satisfies the Karush-Kuhn-Tucker (KKT) conditions, solvability of which is also a sufficient condition for the existence of the minimum (38) (see [16], chapter 12). Computing the partial derivatives, they are

$$\begin{cases} \frac{C_R}{N} - \mu \frac{\epsilon_{max}}{2N} - \delta = 0, \\[2mm] abN - \mu \frac{\epsilon_{max}}{2n} - \lambda = 0, \\[2mm] \epsilon_{max} \leq A, \ n \geq 2, \ N \geq N_{min}, \\[2mm] \mu \geq 0, \ \lambda \geq 0, \ \delta \geq 0, \\[2mm] \mu(\epsilon_{max}(n, N) - A) = 0, \ \lambda(2 - n) = 0, \ \delta(N_{min} - N) = 0. \end{cases} \quad (46)$$

The optimal parameters turn out to be

$$\left(N^*, n^*, \mu^*, \lambda^*, \delta^*\right) = \left(N_{min}, \frac{c^2}{A^2 N_{min}}, \frac{2abc^2}{A^3}, 0, a(1-b)\right), \quad (47)$$

yielding the minimal cost

$$C_R^* = aN_{\min}(1-b) + \frac{abc^2}{A^2}.\tag{48}$$

The LTMCR method has to be competitive with respect to a reference MC solution $u_{\text{ref}}(\mathbf{x}_0)$, which is defined as an approximate solution given by a single MC simulation yielding the accuracy

$$\epsilon_{\text{ref}} = \epsilon_{\text{MC}}(N_{\text{ref}}, h_{\text{ref}}) = A,\tag{49}$$

and using the same confidence interval Γ_q and the same integrator Ξ_h as the LTMCR. According to (6), we require that both the bias and the statistical error be smaller than the accuracy. This allows us to compute N_{ref} and h_{ref} by

$$q\sqrt{\frac{v_{\max}}{N_{\text{ref}}}} \approx A \quad \Rightarrow \quad N_{\text{ref}} \approx \frac{q^2 v_{\max}}{A^2},\tag{50}$$

$$\epsilon_{h_{\text{ref}}} = A \quad \Rightarrow \quad h_{\text{ref}} \approx A/|\beta|,\tag{51}$$

thus yielding

$$C_{\text{ref}} = C_{\text{MC}}(N_{\text{ref}}, h_{\text{ref}}) = \rho \frac{q^2 v_{\max}|\beta|}{A^3}.\tag{52}$$

Comparing this cost with the optimal cost (48), we get

$$C_R^* \le C_{\text{ref}} \Leftrightarrow aN_{\min}(1-b) + \frac{abc^2}{A^2} \le \rho \frac{q^2 v_{\max}|\beta|}{A^3}.\tag{53}$$

Condition (53) is satisfied if the following inequalities hold:

$$\frac{N_{\min}}{h_1}(1-b) \le \frac{q^2 v_{\max}|\beta|}{A^3},\tag{54}$$

$$\frac{1+3l^2}{L}\log\left(1+\frac{L}{h_1}\right) \le \frac{|\beta|}{A}.\tag{55}$$

In (54) and (55) we have considered the expressions (42), (43) and (45) for the parameters a, b, c. Constraint (54) provides an upper bound for $N_{\min}$, but since the accuracy A is usually quite small, $N_{\min}$ can be picked nearly freely still fulfilling that $N_{\min} < \frac{c^2}{2A^2}$. On the other hand, (55) determines the value of A for which the LTMCR achieves a speedup larger than one, once the sample $\{h_i\}_{i=1}^n$ has been fixed. In sum, the theoretical speedup (8) is

$$S(A, h_1, h_n, |\beta|) = \frac{h_1 q^2 v_{\max}|\beta|}{A^3\left(N_{\min}(1-b) + \frac{bc^2}{A^2}\right)}.\tag{56}$$

Remark 2. The LTMCR will deliver a better speedup if: *i*) the accuracy goal A is small; *ii*) the bias decreases fast with h ($|\beta|$ is large);   and/or *iii*) $\phi(\mathbf{X}_\tau)$ has a large variance.

3 General bias

Let us extend the analysis in the preceding section to the case when the leading term of the bias for $h < \bar{h}$ is any power larger or equal than 1/2 (which is the weak convergence rate of the most general integrator, the Euler-Maruyama scheme, for bounded diffusions). In this case the model is

$$\epsilon_{\mathrm{MC}}(N, h_i) \sim u_0 + \epsilon_h(h_i) + \epsilon_{\mathrm{N}}(N, h_i) = \alpha + \beta h_i^r + \delta_i, \tag{57}$$

with $r \geq 1/2$. Again, we consider an evenly spaced cloud $\{h_i\}_{i=1}^n$. Let us define a new variable ξ as

$$\xi = h^r. \tag{58}$$

We may perform the analysis of Section 2 in this case by substitution of h with ξ. The variance of the LTMCR approximation is

$$\mathrm{Var}(\hat{\alpha}_w) = \frac{1}{\sum w_i} + \frac{(\bar{\xi}^w)^2}{\sum w_i(\xi_i - \bar{\xi}^w)^2}. \tag{59}$$

According to Remark 1, $\epsilon_{\mathrm{R}} \leq \epsilon_{\max}$, where

$$\epsilon_{\max}(N, n) \approx q\sqrt{\frac{v_{\max}}{N}}\sqrt{\frac{1}{n} + \frac{\bar{\xi}^2}{\sum \xi_i^2 - n\bar{\xi}^2}} = q\sqrt{\frac{v_{\max}}{N}}\sqrt{\frac{1}{n}\frac{\sum \xi_i^2}{\sum \xi_i^2 - n\bar{\xi}^2}}. \tag{60}$$

Applying the same approximations as before with $r = 1$ and $r = 2$ yields

$$\sum \xi_i \approx h_1^r\left\{1 + \frac{h_1(n-1)}{L}\frac{1}{r+1}\left[\left(1 + \frac{L}{h_1}\right)^{r+1} - 1\right]\right\} = h_1^r\left\{1 + f_1(n-1)\right\}, \tag{61}$$

$$\sum \xi_i^2 \approx h_1^{2r}\left\{1 + \frac{h_1(n-1)}{L}\frac{1}{2r+1}\left[\left(1 + \frac{L}{h_1}\right)^{2r+1} - 1\right]\right\} = h_1^{2r}\left\{1 + f_2(n-1)\right\}, \tag{62}$$

where the positive constants f_1 and f_2 are defined as

$$f_j := \frac{h_1}{L}\frac{1}{jr+1}\left[\left(1 + \frac{L}{h_1}\right)^{jr+1} - 1\right], \qquad j = 1, 2. \tag{63}$$

After some manipulation, recalling that $n\bar{\xi} = \sum \xi_i$, when n is large enough the radicand of (60) can be written as

$$\frac{\sum \xi_i^2}{n \sum \xi_i^2 - n^2\bar{\xi}^2} \sim \frac{1}{n}\frac{f_2}{f_2 - f_1^2}. \tag{64}$$

From (64), the difference $f_2 - f_1^2$ must be positive, or, equivalently, $Q = f_2/f_1^2 > 1$. By making the substitution $y = 1 + L/h_1 = h_n/h_1 > 1$, we can study the function

$$Q(y, r) = \frac{(r+1)^2}{2r+1}\frac{(y-1)(y^{2r+1} - 1)}{(y^{r+1} - 1)^2}, \tag{65}$$

which, according to Maple, fulfils $Q(y, r) > 1$ if $y > 1$. This means that the constants always satisfy $f_2/(f_2 - f_1^2) > 0$ and that we can approximate the error function as

$$\epsilon_{\max}(N, n) \approx \frac{c}{\sqrt{nN}}, \quad \text{with } c := q\sqrt{\frac{v_{\max}f_2}{f_2 - f_1^2}}. \tag{66}$$

The cost is again

$$C_R(N, n) \approx aN\big(1 + b(n-1)\big), \quad \text{with } a := \frac{k\rho}{h_1} \quad b := \frac{h_1}{L}\log\Big(1 + \frac{L}{h_1}\Big). \tag{67}$$

Solving system (46) yields the (constrained) minimal cost

$$C_R^* = C_R(N^*, n^*) = aN_{\min}(1-b) + \frac{abc^2}{A^2}. \tag{68}$$

Notice that the solution is formally the same as the one of the linear case, but the constant c changes. Concerning the single Monte Carlo reference simulation for nonlinear bias, N_{ref} remains as in (50), while

$$h_{\text{ref}} = \big(A/|\beta|\big)^{1/r}, \tag{69}$$

resulting in a reference cost

$$C_{\text{ref}} = C_{\text{MC}}(h_{\text{ref}}, N_{\text{ref}}) = k\rho\frac{q^2 v_{\max}}{A^2}\Big(\frac{|\beta|}{A}\Big)^{1/r}. \tag{70}$$

The equivalent to conditions (54) and (55) are

$$\frac{N_{\min}}{h_1}(1-b) \leq \frac{q^2 v_{\max}}{A^2}\Big(\frac{|\beta|}{A}\Big)^{1/r}, \tag{71}$$

$$\frac{f_2}{f_2 - f_1^2}\frac{1}{L}\log\Big(1 + \frac{L}{h_1}\Big) \leq \Big(\frac{|\beta|}{A}\Big)^{1/r}, \tag{72}$$

with a similar interpretation as in the linear case. In conclusion, the theoretical speedup function for the case of general bias is

$$S(A, h_1, h_n, |\beta|, r) = \frac{C_{\text{ref}}}{C_R^*} = \frac{h_1 q^2 v_{\max}|\beta|^{1/r}}{A^{2+1/r}\big(N_{\min}(1-b) + \frac{bc^2}{A^2}\big)}. \tag{73}$$

Remark 3. The speedup will be better if a low-order integrator Ξ_h (such that $r \gtrsim 1/2$) is used. The reason is that, in that case, h_{ref} must be smaller.

3.1 Complete LTMCR algorithm.

Algorithm 1 is a practical pseudocode. Note that we have three kinds of input data. First, the asymptotic weak convergence rate of the integrator Ξ_h, r, is assumed known (from the integrator theory) and valid within the interval $[h_1, \bar{h}]$. (While in the theory we allow $h_n \leq \bar{h}$, here we just set $h_n = \bar{h}$ for simplicity.) Second, there is an accuracy target (A with probability Γ_q). Finally, in order to get the optimal performance from the LTMCR (i.e. the least computational time overall), $v_{\max}$ and $N_{\min}$ within $[h_1, \bar{h}]$ must be estimated. Recall that $N_{\min}$ is the least number of trajectories such that the sample variance is deemed meaningful. This can be gauged via the variance of the sample variance, which is [15]

$$\text{Var}(\sigma^2) = \frac{2\sigma^4}{N-1}. \tag{74}$$

In order to do so, a preprocessing set of fast MC simulations are run on $\tilde{n}$ points $\{\tilde{h}_j\}$ in $[h_1, \bar{h}]$ until the variance of the variance is some small fraction $\tilde{a}$ of the variance itself. The details of this are necessarily vague. We have used $\tilde{n} = 5$ and $\tilde{a} = .05$ and have simply picked the largest values in the set for $v_{\max}$. For $N_{\min}$ we could have done the same but instead took $N_{\min} = 10^6$, which is more that necessary and leaves scope for further improvement in the speedups reported in Section 4.

Note that, while there may seem to be many parameters, most of them would be needed with the plain MC method as well. For instance, in Section 4 we will compare the LTMCR with plain MC simulations. In order to achieve an accuracy A for a plain MC simulation at h, we need to know r, whether h is small enough for r to hold, and $N_{\min}$ at h such that the sample variance is a safe estimate–just like in the LTMCR method.

Finally, let us emphasize that the LTMCR is fully parallelizable except for the preprocessing runs and the final regression computations (of negligible cost).

4   Numerical experiments

In this section we solve pointwise two BVPs of the type (3), combining the LTMCR algorithm with a different numerical integrator Ξ_h in each case (more examples can be found in [12]). The code for the MC approximations, both for the reference MC simulation and for those used in the LTMCR, is written in C++ and run on a 64-core cluster using MPI. In both examples, $q = 2$ ($\Gamma_2 = 95.5\%$), $N_{\min} = 10^6$, r is known; and we report:

1. The detailed results of a single LTMCR cloud, for the purpose of illustration, where we choose to fix (n, N, h_1, h_n) (there is no accuracy target A).

2. The error distribution of the LTMCR cloud with the same (n, N, h_1, h_n) as above, approximated with a histogram of the experimental error over many LTMCR simulations.

3. The agreement between observed, predicted, and theoretically worst-case (i.e. lower bound) speedups. Here, we set *a priori* values for $N_{\min}$ and h_n and investigate the cases with A fixed and h_1 variable, and vice versa. Recall that now the LTMCR uses the values for $N^* = N_{\min}$ and $n^* \geq 2$ which maximize the expected speedup under an accuracy target A, q.

Example 1: Euler-Maruyama integrator. Let us consider a two dimensional BVP like (3) with

$$\sum_{i=1}^{d} \sum_{j=1}^{d} a_{ij} \frac{\partial^2 u}{\partial x_i \partial x_j} = \nabla^2, \quad \mathbf{b} = \begin{pmatrix} x \\ y/2 \end{pmatrix}, \quad c = y + 1/2, \quad f = -2xy - xy^2, \quad g = xy. \quad (75)$$

The diffusion matrix for the Laplacian in $\mathbb{R}^d$ is $\sigma = \sqrt{2}I_d$ (where I_d is the $d \times d$ identity matrix). The domain Ω is a ball centered in $(3/2, 0)^T$ with radius $R = 1.8$. The solution of (75) is $u(x, y) = g(x, y)$ and we computed it on $\mathbf{x_0} = (1.85714, 0.5)$, where $u_0 = u(\mathbf{x_0}) = 0.9286$. In the following LTMCR test, we used a cloud of $n = 26$ MC simulations with $N = 10^6$ trajectories each. The cloud time steps

---

Algorithm 1 LTMCR algorithm

Require: r and $[h_1, \bar{h}]$
Data: A and q

1. Estimate $v_{\max}$ and $N_{\min}$ in $[h_1, \bar{h}]$:
let $\tilde{n} \approx 5$ and $\tilde{a} \approx .05$
let $\{\tilde{h}_k\}$ be $\tilde{n}$ equispaced points in $[h_1, \bar{h}]$
for $k = 1, \dots, \tilde{n}$ **do**
   let $N_k = 0$ and $\tilde{\sigma}_k^2 = \infty$
   repeat
      set a new seed of the random number generator
      compute one more realization of (2) with Ξ_h and $\tilde{h}_k$
      let $N_k = 1 + N_k$
      compute the sample variance $\tilde{\sigma}_k^2$ according to (12).
   until $\frac{2\tilde{\sigma}_k^4}{N_k - 1} < \tilde{a}\tilde{\sigma}_k^2$
end for
estimate $v_{\max}$ from $\{\tilde{\sigma}_k^2\}$ and $N_{\min}$ from $\{N_k\}$

2. Find the optimal parameters for LTMCR, n^* and N^*:
calculate L, l as in (33), f_1, f_2 as in (63), and a, b, c as in (66) and (67)
let $N^* = N_{\min}$ and $n^* = \lceil \frac{c^2}{A^2 N_{\min}} \rceil$ as in (47)
let $\{h_i\}$ be n^* equispaced points in $[h_1, \bar{h}]$

3. Perform n^* independent MC simulations at $h_1, \dots, \bar{h}$:
for $i = 1, \dots, n^*$ **do**
   for $j = 1, \dots, N^*$ **do**
      set a new seed of the random number generator
      solve the SDE (2) with Ξ_h and h_i
   end for
   compute $u_{MC}(h_i)$ as in (4), σ_i as in (12), and the weight w_i as in (14)
end for

4. Perform the weighted regression between $\{u_{MC}(h_i)\}_{i=1}^n$ and $\{h_i^r\}$:
construct W and U according to (17)-(18)
construct X like in (16), but substituting h_i^r for h_i $(i = 1, \dots, n^*)$
with this X, obtain the LTMCR solution $\hat{\alpha}_w$ from (15)

---

are uniformly distributed in $[h_1, h_{26}] = [0.001, 0.026]$. For each of the n MC simulations in the cloud, the Euler-Maruyama integrator ($r = 1/2$) was used [11].

Figure 2 shows the results of the LTMCR simulation described above. The fitted bias has been removed from the solutions to yield the data labelled as 'clean'. The fact that they hover on average around a well-defined value suggests that the bias has actually been removed. More precisely, the coefficient of determination between $\{u_{MC}(N, h_i)\}$ and $\{h_i^{1/2}\}$ is $R^2 = 0.9976$. Using $r = 1/2$ in (57), we obtain as outputs of the regression the estimator values $u_R = \hat{\alpha}_w = 0.9307$ and $\hat{\beta}_w = -1.4568$.
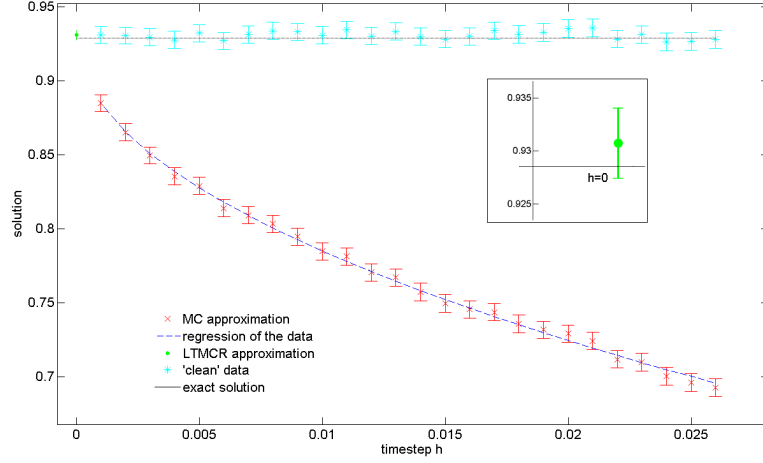
Figure 2: Bias behavior and solution extrapolation with LTMCR for Example 1. The horizontal line is u_0, and the inset zooms in on the LTMCR solution, u_R, at $h = 0$.

In Figure 3 we compare the histogram of $u_R - u_0$ with the fitted normal distribution $N(\hat{\mu}, \hat{\sigma}^2)$, where $\hat{\mu} = 0.0035$ and $\hat{\sigma} = 0.0018$. We address now the attained speedup, S. Formulas (51) and (50) yield $h_{\text{ref}} = (A/|\hat{\beta}_w|)^{1/r}$ and $N_{\text{ref}} = (q/A)^2 v_{\text{max}}$, where $v_{\text{max}} = 9.1$ is the largest sample variance in the cloud. Results concerning S are given in Tables 1 and 2 and Figure 4, showing good agreement with the theory and that the lower bounds for the speedup (56) are conservative. T_R is the CPU times, in seconds, for the LTMCR simulation.

Interestingly, the maximum speedup for a set accuracy ($h_1^* \approx 0.0025$) roughly coincides with that of the lower bound estimate for S -see Figure 4 (left).

A	theoretical S	numerical S	n^*	N_{ref}	h_{ref}	T_R
0.0021	323.4	280.0	98	8.6×10^6	1.96×10^{-6}	138
0.0025	203.2	197.6	64	5.6×10^6	3.00×10^{-6}	99
0.0030	141.5	139.4	45	3.9×10^6	4.16×10^{-6}	67
0.0040	70.7	72.2	26	2.3×10^6	7.59×10^{-6}	41
0.0050	42.2	44.8	17	1.5×10^6	11.4×10^{-6}	29
0.0060	25.1	26.6	12	1.0×10^6	17.0×10^{-6}	22
0.0070	16.6	16.7	9	0.7×10^6	22.8×10^{-6}	19

Table 1: Speedup $S(A, h_1 = 0.001)$ for Example 1. Data of Figure 4 (right).
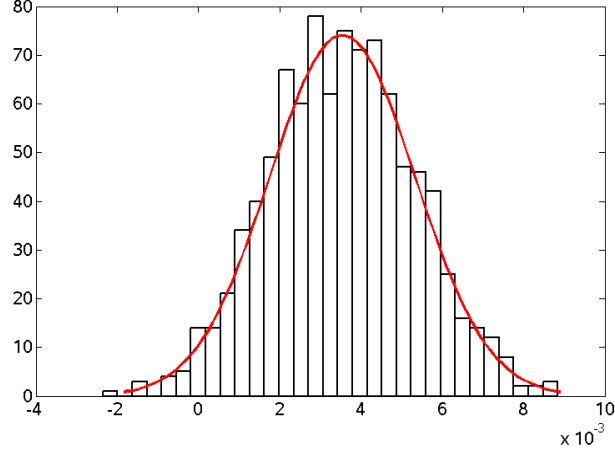
Figure 3: Histogram over 950 samples of the LTMCR error of Example 1.

h_1	theoretical S	numerical S	n^*
0.0010	42.2	42.8	17
0.0019	49.6	49.6	20
0.0028	51.1	47.7	34
0.0047	44.2	41.3	33
0.0084	29.8	26.4	64
0.0121	17.5	15.7	129
0.0158	9.6	7.3	315
0.0195	3.4	2.6	952

Table 2: Speedup S(A=.005, h_1) for Example 1 with $N_{\text{ref}} = 1.5 \times 10^6$ and $h_{\text{ref}} = 11.4 \times 10^{-6}$. Data of Figure 4 (left).

Example 2: Gobet-Menozzi integrator. The second example is a three-dimensional problem with functions

$$c(x, y, z) = 0, \qquad g(x, y, z) = xyz, \qquad \mathbf{b}(x, y, z) = [y, z, x]^{\mathrm{T}},$$

$$f(x, y, z) = -y^2 z - z^2 x - x^2 y - \tfrac{1}{2}\left(z \sqrt{1 + |x|} \sqrt{1 + |z|} + x \sqrt{\tfrac{3}{4}} \sqrt{1 + |x|} \sqrt{1 + |y|}\right) \quad (76)$$

$$\sigma(x, y, z) = \begin{pmatrix} \sqrt{1 + |z|} & 0 & 0 \\ \tfrac{1}{2} \sqrt{1 + |x|} & \sqrt{\tfrac{3}{4}} \sqrt{1 + |x|} & 0 \\ 0 & \tfrac{1}{2} \sqrt{1 + |y|} & \sqrt{\tfrac{3}{4}} \sqrt{1 + |y|} \end{pmatrix}$$

The domain is a ball with radius $R = 1.8$ centered at the origin. The solution is $u(x, y) = g(x, y)$ and we computed it on $\mathbf{x_0} = (-0.7, 0.3, 0.3)$, where $u_0 = u(\mathbf{x_0}) = -0.0630$. In this problem, we use the integrator proposed by Gobet and Menozzi in [8]. Theoretically, the Gobet-Menozzi integrator has a bias $o(\sqrt{h})$ (meaning that the leading power is at least 1/2), but in this specific problem,
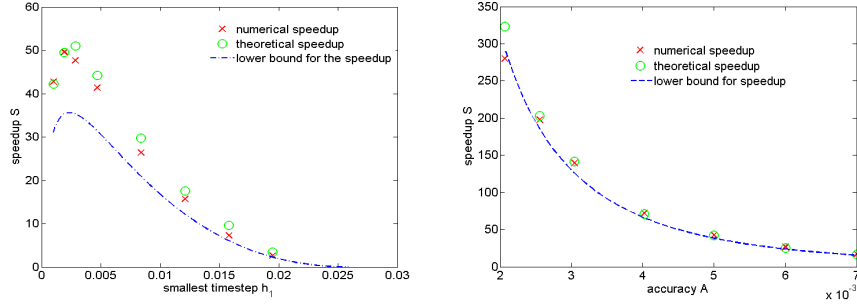
16

Figure 4: Speedup for Example 1. Left: $S(A = 0.005, h_1)$, see also Table 2. Right: $S(A, h_1 = 0.001)$, see also Table 1.

(76), it experimentally yielded a linear bias in the tests in [8]. Since our goal here is just to check the LTMCR in a case with linear bias, we chose this BVP and the Gobet-Menozzi integrator, due to its extreme simplicity. (Many integrators with rigorously $O(h)$ bias can be found in [14], chapter 6.) For the illustrative LTMCR test below we used a cloud with $n = 45$ timesteps uniformly distributed in $[h_1, h_{45}] = [0.0277, 0.1]$ and $N = 10^6$ trajectories per timestep.
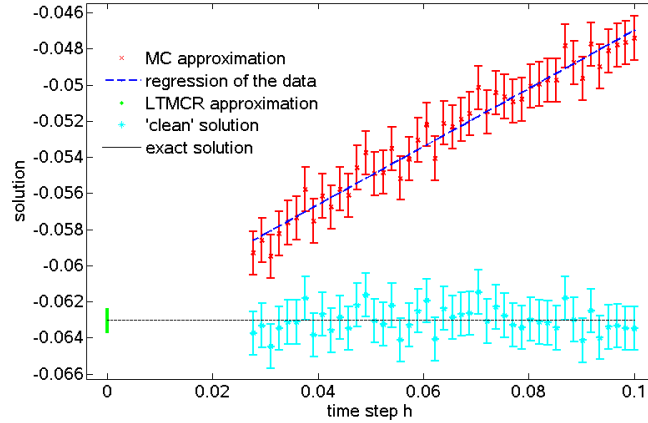

Figure 5: Bias behavior and solution extrapolation with LTMCR for Example 2.

The output of the regression analysis is $u_R = \hat{\alpha}_w = -0.06303$, $\hat{\beta}_w = -0.1607$, and $R^2 = 0.9609$. In Figure 6, a histogram of the LTMCR error is shown. The fitted normal distribution has parameters $\hat{\mu} = -0.0097$ and $\hat{\sigma} = 5.3078e{-}4$: as in the previous case it was not possible to remove all of the bias in order to have a distribution centered at zero. The speedup results are compiled in Tables 3 and 4. The data in Table 3 are also represented in Figure 7. Contrary to Example 1, Table 4 shows apparently monotonical speedup. The reason is simply that the value h_1^* for which $S(A = 0.0001, h_1)$ hits the maximum happens to be below the range of timesteps considered in the table.
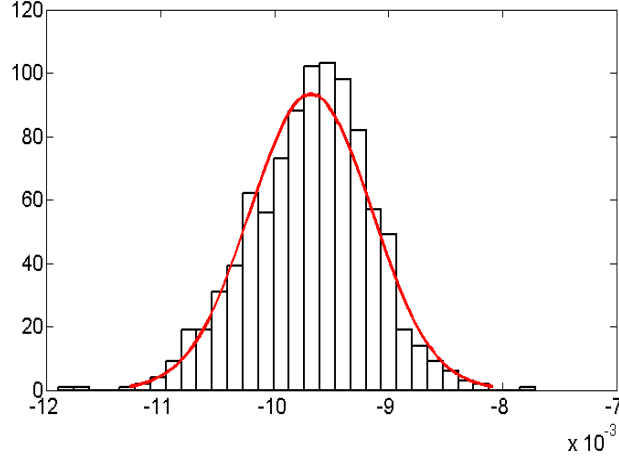
17

Figure 6: Histogram over 950 samples of the LTMCR error for Example 2.

Same problem, different integrator. Due to the linear bias of the Gobet-Menozzi integrator in this example, S is not as large as in the previous example with the Euler-Maruyama integrator. However, the CPU time should be less using a higher-order integrator. In order to check this, we solved Example 2 with the LTMCR using Euler-Maruyama instead (Figure 7, right). Because the leading, $O(\sqrt{h})$, term kicks in closer to zero now, the cloud must be shifted towards the left. For $A = 0.43 \times 10^{-3}$ (see Table 3), CPU times (in seconds) for the LTMCR are 55 using the Gobet-Menozzi integrator and 384 using Euler-Maruyama. However, S is better in the second case because the reference MC simulations took 107 s. and 120355 s., respectively.

A	theoretical S	numerical S	n^*	N_{ref}	h_{ref}	T_{R}
0.05×10^{-3}	17.68	17.02	6124	592×10^6	0.3×10^{-3}	4065
0.145×10^{-3}	6.10	5.95	729	70×10^6	0.9×10^{-3}	485
0.24×10^{-3}	3.68	3.56	266	25×10^6	1.5×10^{-3}	178
0.43×10^{-3}	2.00	1.95	83	8×10^6	2.7×10^{-3}	55
0.62×10^{-3}	1.38	1.42	40	3.9×10^6	3.8×10^{-3}	26
0.81×10^{-3}	1.07	1.00	24	2.3×10^6	4.8×10^{-3}	17
0.84×10^{-3}	1.04	1.00	22	2.1×10^5	4.9×10^{-3}	15
1.0×10^{-3}	0.85	0.75	16	1.5×10^6	5.9×10^{-3}	12

Table 3: Speedup $S(A, h_1 = 0.0277)$ for Example 2.

Remark 4. The LTMCR stands on the choices that i) $N^* \geq N_{\text{min}}$ is the same for all the cloud points, and ii) the cloud points are equispaced. Assumptions i) and ii) enable semi-analytic approximations leading to closed formulas for the solution of the constrained minimization problem.

18

h_1	theoretical S	numerical S	n^*
0.0277	4.39	4.05	1531
0.0300	4.18	3.83	1666
0.0324	3.91	3.61	1835
0.0371	3.43	3.10	2241
0.0466	2.46	2.19	3460

Table 4: Speedup $S(A = 0.0001, h_1)$ for Example 2, with $N_{\text{ref}} = 14.7 \times 10^7$ and $h_{\text{ref}} = 0.0012$.
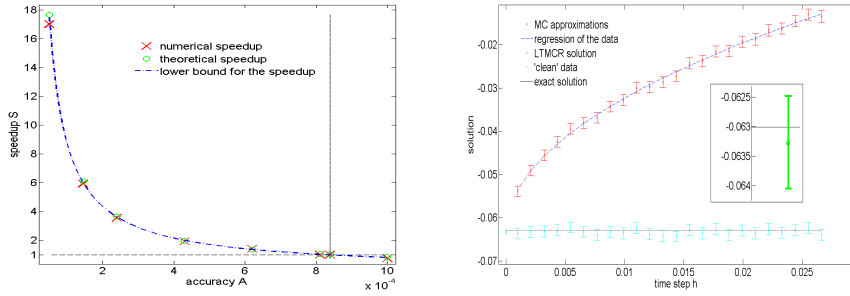

Figure 7: Left: Speedup $S(A, h_1 = 0.0277)$ for Example 2. The lower bound was calculated with the observed value $v_{\max} = 0.37$. The crossover value (for which $S = 1$) was accurately predicted using (56) (black vertical line). Right: LTMCR extrapolation for Example 2 but using the Euler-Maruyama integrator rather than Gobet-Menozzi's (compare the scale with that of Figure 5). The inset zooms in on the LTMCR solution (i.e. at $h = 0$).

Imagine that $N_i = N(h_i)$ were variable rather than constant. The minimization problem would then be n^*+1-dimensional instead of bidimensional. Assuming, for the sake of the argument, that the minimum $(n^*, N_1^*, ..., N_{n^*}^*)$ could still be found in a practical and accurate way such as with the LTMCR, one should still require $N_i^* \geq N_{\min}(h_i)$ for consistency. But it is unrealistic to expect that an h-dependent $N_{\min}(h_i)$ can be known for exploiting this.

Also, note that our semi-analytic approximations break down if $n = O(1)$, so it might happen that taking a 'cloud' with only $2, 3, ...$ points would yield better speedups. This depends on the parameters $N_{\min}, h_1, h_n$. A convenient advantage of taking the cloud with n^* points is that it allows one to check, after computing the highest (and least time-consuming) values of h_i in the cloud (i.e. $h_n, h_{n-1}, ...$), whether those parameters were acceptable indeed, and abort the simulation otherwise. As an illustration, consider Figure 8, where the statistical error has been suppressed for the sake of clarity. After computing the MC solution corresponding to the black dots, it could be inferred that h_n has been picked too large (recall that $\bar{h}$ is unknown), because the black dots do not follow the curve $u + \beta h^r$ (dashed line). On the other hand, just the end points (squares) do not reveal the shape in between, and would lead here to wrong

19

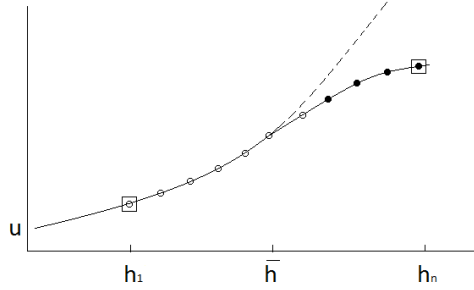results that could not be detected even *a posteriori*.

Figure 8: Advantage of a cloud (circles) over just two points (squares), see Remark 4. Dashed line: asymptotic behavior $u + \beta h^r$. Solid line: MC solution with actual bias. Notice that the noise has been suppressed. Compare with Figs. 2 and 5.

5 Conclusions

The LTMCR exploits the fact the power of the leading term is often known for the triplet integrator Ξ_h / BVP (3) / domain Ω. (An up-to-date survey and comparison of existing integrators is [2].) Another advantage of the LTMCR is that it is extremely easy to implement and to combine with existing codes for Monte Carlo solutions of BVPs.

The error involved in the LTMCR has been rigourously studied in order to compare its complexity with that of a Monte Carlo simulation with single time step and the same confidence intervals. Numerical results support the theoretical analysis and show that the LTMCR can deliver a substantial speedup. Although the analysis here only considers linear elliptic BVPs with Dirichlet BCs, it could very easily be extended to other situations such as parabolic BVPs, Neumann BCs, and option pricing in finance.

6 Acknowledgments

References

[1] J.A. Acebrón, M.P. Busico, P. Lanucara, and R. Spigler, *Domain decomposition solution of elliptic problems via probabilistic methods*. SIAM J. Sci. Comput., **27** 440-457 (2005).

[2] F. Bernal and J.A. Acebrón, *A comparison of higher-order weak numerical schemes for stopped stochastic differential equations in bounded domains*. Submitted (2015).

[3] F.M. Buchmann, *Simulation of stopped diffusions*. J. Comput. Phys. **202(2)**, 446-462 (2005).

[4] S. Chatterjee, and A.S. Hadi, *Regression analysis by example*. Wiley (2012).

[5] M. Freidlin, *Functional integration and Partial differential equations*. Annals of Mathematics studies, Princeton University Press (1985).

[6] M.B. Giles, *Multi-level Monte Carlo path simulation*. Operations Research, **56**(3), 607-617 (2008).

[7] E. Gobet and S. Menozzi, *Exact approximation rate of killed hypoelliptic diffusions using the discrete Euler scheme*. Stoch. Proc. Appl. **112**(2), 201-223 (2004).

[8] E. Gobet and S. Menozzi, *Stopped diffusion processes: overshoots and boundary correction*. Stoch. Proc. Appl., **120**, 130-162 (2010).

[9] D.J. Higham, X. Mao, M. Roj, Q. Song, and G. Yin, *Mean Exit Times and the Multilevel Monte Carlo Method*. SIAM/ASA J. Uncert. Quant., **1**(1), 2-18 (2013).

[10] K.M. Jansons and G.D. Lythe, *Exponential timestepping with boundary test for stochastic differential equations*. SIAM J. Sci.Comput. **24**, 1809-1822 (2003).

[11] P.E. Kloeden and E. Platten, *Numerical Solution of Stochastic Differential Equations*. Springer, Applications of Mathematics 23 (1999).

[12] S. Mancini, *Monte Carlo Approximations of Boundary Value Problems: an Efficient Algorithm*. Master thesis, Università degli Studi di Milano, 2013.

[13] R. Mannella, *Absorbing boundaries and optimal stopping in a stochastic differential equation*. Phys. Lett. A **254**, 257-62 (1999).

[14] G.N. Milstein and M.V. Tretyakov, *Stochastic Numerics for Mathematical Physics*. Springer (2004).

[15] J. Neter, M. Kutner, W. Wasserman, and C. Nachtsheim, *Applied linear statistical models*. Irwin, 4$^{\text{th}}$ edition (1996).

[16] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer (2000).

[17] K. K. Sabelfeld, *Monte Carlo methods in boundary value problems*. Series in Computational Physics, Springer-Verlag, Berlin (1991).

[18] D. Talay and L. Tubaro, *Expansion of the global error for numerical schemes solving stochastic differential equations*. Stoch. Anal. and App., **8(4)**, 94-120 (1990).