Improving MLMC for SDEs with application to the Langevin equation

Eike H. Müller Rob Scheichl Tony Shardlow

September 9, 2014

Abstract

This paper applies several well-known tricks from deterministic differential equations to improve the efficiency of the Multilevel Monte Carlo (MLMC) method for stochastic differential equations (SDEs) and especially the Langevin equation. We use modified equations analysis to circumvent the need for a strong-approximation theory for the integrator. This gives greater freedom in the application of MLMC. For example, this allows us to replace Gaussian increments with discrete approximations that satisfy certain moment conditions and that allow an exact integration on the coarser levels. To be able to use large time steps to exploit this fully, we employ stable geometric integrators based on operator splitting for Langevin-type equations. The final enhancement we investigate is extrapolation, which is a well-known technique to reduce the bias and combines in an very natural way with MLMC. Combined, our enhancements lead to an order of magnitude improvement in efficiency, as we show in numerical experiments for MLMC applied to the Langevin equation.

Keywords numerical solution of stochastic differential equations, modified equations, geometric integrators, weak approximation, extrapolation.

1 Introduction

This paper is concerned with the numerical solution of stochastic differential equations (SDEs) by the Multilevel Monte Carlo (MLMC) method. MLMC (Giles, 2008; Heinrich, 2001) is an important variance-reduction method that is well established by now and has been successfully applied to a wide class of problems in stochastic simulation and in uncertainty quantification; for example, (Anderson & Higham, 2011; Barth *et al.*, 2011; Cliffe *et al.*, 2011; Dereich & Heidenreich, 2011; Giles & Reisinger, 2012; Giles & Szpruch, 2013; Hoel *et al.*, 2012; Mishra *et al.*, 2012). The variance reduction in MLMC is achieved by computing approximations of the solution on different "levels" consisting, in the SDE case, of numerical integrators with different time-step sizes. These computations are then combined in an efficient way to define a multilevel estimator for the moments that has a smaller variance than the standard Monte Carlo estimator and can therefore be computed faster.

Let $(\Omega, \mathcal{F}, \mathbb{P})$ denote a probability space and let $\mathbb{E}$ and Var denote the expectation and variance with respect to $\mathbb{P}$. Consider first the initial-value problem

$$d\boldsymbol{X} = \boldsymbol{f}(\boldsymbol{X})\,dt + G(\boldsymbol{X})\,d\boldsymbol{W}(t), \qquad \boldsymbol{X}(0) = \boldsymbol{X}_0, \tag{1.1}$$

for $\boldsymbol{f}\colon \mathbb{R}^d \to \mathbb{R}^d$ and $G\colon \mathbb{R}^d \to \mathbb{R}^{d\times m}$ and initial data $\boldsymbol{X}_0 \in \mathbb{R}^d$. Here $\boldsymbol{W}(t)$ is a vector of m *iid* standard Brownian motions on $(\Omega, \mathcal{F}, \mathbb{P})$. Suppose that there exists a well-defined solution $\boldsymbol{X}(t)$ when Eq. (1.1) is interpreted as an Ito integral equation. For simplicity, we only consider approximating moments of the solution at a prescribed end time as the quantities of interest, but other, more complicated functionals could also be studied. That is, we are interested in computing $\mathbb{E}[\phi(\boldsymbol{X}(T))]$ for some $\phi\colon \mathbb{R}^d \to \mathbb{R}$ and time $T > 0$. Consider the approximation by

a sequence of random variables $\boldsymbol{X}_n \approx \boldsymbol{X}(t_n)$ for $t_n = nh$ with $n \in \mathbb{N}$ and a time step h. For example, $\boldsymbol{X}_n$ may result from the Euler–Maruyama method

$$\boldsymbol{X}_{n+1} = \boldsymbol{X}_n + \boldsymbol{f}(\boldsymbol{X}_n)h + G(\boldsymbol{X}_n)\sqrt{h}\,\boldsymbol{\xi}_n, \tag{1.2}$$

with $\boldsymbol{\xi}_n \sim \mathrm{N}(0, I)$ *iid*. In this case, $\boldsymbol{X}_n$ is a weak first-order approximation to $\boldsymbol{X}(t_n)$ so that, for any $\phi\colon \mathbb{R}^d \to \mathbb{R}$ in a suitable class of test functions $\mathcal{C}$,

$$\sup_{0\le t_n\le T} \mathbb{E}\big[\phi(\boldsymbol{X}(t_n))\big] - \mathbb{E}\big[\phi(\boldsymbol{X}_n)\big] = \mathcal{O}\big(h\big).$$

If $\boldsymbol{f}$ and G are sufficiently smooth, $\mathcal{C}$ contains all infinitely differentiable functions whose derivatives are polynomially bounded; for example, (Kloeden & Platen, 1992, Theorem 14.5.1).

In some cases (Shardlow, 2006; Zygalakis, 2011), it is possible to find a second SDE, called the modified SDE with solution $\boldsymbol{X}_h(t)$, such that $\boldsymbol{X}_n$ is a second-order weak approximation to $\boldsymbol{X}_h(t)$; that is,

$$\sup_{0\le t_n=nh\le T} \mathbb{E}\big[\phi(\boldsymbol{X}_h(t_n))\big] - \mathbb{E}\big[\phi(\boldsymbol{X}_n)\big] = \mathcal{O}\big(h^2\big). \tag{1.3}$$

Then, the solution of the modified equation $\boldsymbol{X}_h(t)$ is an order of magnitude closer to the numerical solution than $\boldsymbol{X}(t)$. The modified equation takes the form

$$d\boldsymbol{X}_h = \tilde{\boldsymbol{f}}(\boldsymbol{X}_h)\,dt + \widetilde{G}(\boldsymbol{X}_h)\,d\boldsymbol{W}(t), \qquad \boldsymbol{X}(0) = \boldsymbol{X}_0, \tag{1.4}$$

where $\tilde{\boldsymbol{f}} = \boldsymbol{f} + h\boldsymbol{f}_1$ and $\widetilde{G} = G + hG_1$ for some $\boldsymbol{f}_1\colon \mathbb{R}^d \to \mathbb{R}^d$ and $G_1\colon \mathbb{R}^d \to \mathbb{R}^{d\times m}$. This reduces to Eq. (1.1) with $h = 0$, and $\boldsymbol{f}_1$ and G_1 describe the correction in the drift and diffusion needed to achieve Eq. (1.3). Our results concern SDEs and numerical integrators where the second-order modified equation is available. Except in special cases (e.g., if G is independent of $\boldsymbol{X}$), this does not include the Euler–Maruyama method (Shardlow, 2006). It does include the Milstein method, which has a second-order modified equation (Zygalakis, 2011).

Using weak-approximation theory and modified equations, we develop an alternative method of analysis for MLMC in this paper. By doing this, we no longer depend on the strong-approximation properties of the integrator (as in other papers, e.g. (Giles, 2008)) and this gives greater freedom to apply MLMC. As one of these improvements, we show how discrete random variables, as an approximation to the Gaussian increments of a Brownian motion, can be used within MLMC. This would be difficult to analyse by the standard MLMC analysis, since all the approximation results for integrators based on discrete random variables are in distribution only (e.g., (Kloeden & Platen, 1992, §14.2)). In fact, for all our problems, a three-point approximation is sufficient and allows direct evaluation of the moments of the numerical solution without sampling. The cost of this direct evaluation grows exponentially with the number of time steps, but it requires no sampling and, for small numbers of time steps, its cost is significantly smaller than that for a Monte Carlo-type estimator. We employ it on the coarsest level of the multilevel estimator where the number of time steps is smallest and the number of samples is largest, thus leading to the largest gains.

In order to take large time-steps in the integration of Eq. (1.1), it is necessary to ensure the stability of the integrator. It is well known from deterministic differential equations that most explicit integrators will have a stability constraint on the time-step size. This is the same for SDEs and such stability constraints may severely restrict the number of coarse levels that can be employed in the MLMC method (Abdulle & Blumenthal, 2013; Hutzenthaler *et al.*, 2013) and thus its efficiency. However, there are explicit integrators that are unconditionally stable. We discuss a class of such integrators for an important model in molecular dynamics and atmospheric dispersion, the *Langevin equation*:

$$\begin{aligned} d\boldsymbol{P} &= -\lambda\boldsymbol{P}\,dt - \nabla V(\boldsymbol{Q})\,dt + \sigma\,d\boldsymbol{W}(t),\\ d\boldsymbol{Q} &= \boldsymbol{P}\,dt \end{aligned} \tag{1.5}$$

for parameters $\lambda, \sigma > 0$, a potential $V\colon \mathbb{R}^d \to \mathbb{R}$, and a d-dimensional vector $\boldsymbol{W}(t)$ of *iid* Brownian motions. We specify initial conditions $(\boldsymbol{Q}(0), \boldsymbol{P}(0)) = (\boldsymbol{Q}_0, \boldsymbol{P}_0) \in \mathbb{R}^{2d}$. This system is

used in molecular dynamics to simulate a system of particles in a heat bath and has equilibrium distribution with pdf $Z^{-1}\exp(-H(\boldsymbol{Q},\boldsymbol{P})/k_{\mathrm{B}}T)$, known as the *Gibbs canonical distribution*, where Z is a normalisation constant, $H(\boldsymbol{Q},\boldsymbol{P}) \coloneqq \frac{1}{2}\boldsymbol{P}^{\mathsf{T}}\boldsymbol{P} + V(\boldsymbol{Q})$, and $k_{\mathrm{B}}T = \sigma^2/2\lambda$. As usual, k_{B} denotes the Boltzmann constant and T temperature. Numerical integrators for the Langevin equation are well developed for example in (Beard & Schlick, 2000; Brunge *et al.*, 1984; Wang & Skeel, 2003). Recently, there has been a strong push to understand the invariant measure associated to the integrators (Bou-Rabee & Owhadi, 2010; Debussche & Faou, 2012; Kopec, 2013; Zygalakis, 2011). Second-order modified equations are available for the most important integrators for the Langevin equation. In particular, we study splitting methods based on exact sampling of an Ornstein–Uhlenbeck process and symplectic integrators (symplectic Euler and Störmer–Verlet) for the Hamiltonian part. Since these integrators are unconditionally stable, we can go to very small numbers of time steps on the coarsest level in the MLMC method and then evaluate the moments on the coarsest level directly at a dramatically reduced cost in comparison to Monte Carlo.

The final enhancement that we investigate is extrapolation (Talay & Tubaro, 1990). It is a natural addition to MLMC methods, already mentioned in the original work (Giles, 2008) and studied in more detail in (Lemaire & Pagès, 2013). It reduces the bias in the numerical approximation of the solution due to time stepping and relies on having a sharp estimate for the bias error. If such an estimate is available, it is possible to eliminate the leading-order error term in the bias error by extrapolating from a sequence of approximations with differing time-step sizes. These approximations are naturally available in MLMC and the extrapolation can significantly enhance the accuracy of the multilevel estimator, or conversely significantly reduce the cost, as our experiments confirm.

We provide a set of MLMC experiments comparing integrators based on splitting methods with Gaussian and discrete random variables, as well as extrapolation, for the Langevin equation with a harmonic and double-well potential. Our results confirm that the splitting methods are significantly more stable than the Euler–Maruyama method and are much more efficient for stiff SDEs when combined with MLMC, discrete random variables, and extrapolation. All methods have the same asymptotic ϵ-cost; that is, the cost always grows inverse proportionally to the mean-square error, but the proportionality constant is reduced by an order of magnitude from the standard Euler–Maruyama method through our enhancements.

The C++ source code that we developed for these numerical experiments is freely available for download under the LGPL 3 license.

The paper is organised as follows. §2 reviews MLMC, including the important complexity theorem. §3 uses modified equations to apply the complexity theorem, depending only on weak convergence of the integrators. §4 reviews splitting methods for the Langevin equation and defines a number of integrators where modified equations are available. It also introduces the three-point approximation for Gaussian increments as well as extrapolation. Numerical experiments are presented in §5 to demonstrate the effectiveness of this methodology for the Langevin equation and to give quantitative predictions of the possible gains.

2 Background on MLMC

When solving an SDE numerically, the total error consists of the bias due to the time-stepping method and the Monte Carlo sampling error. The sum of these two terms should be reduced below a given small tolerance ϵ. A standard Monte Carlo method achieves this by computing M sample paths, with $M^{-1} = \mathcal{O}(\epsilon^2)$, and taking time step $h = \mathcal{O}\bigl(\epsilon^{1/\alpha}\bigr)$, where α is the order of weak convergence (e.g., $\alpha = 1$ for the Euler–Maruyama method). Hence, we can achieve accuracy ϵ with total cost $\mathrm{Cost}^{(\mathrm{MC})}(\epsilon) = \mathcal{O}(h^{-1} \times M) = \mathcal{O}\bigl(\epsilon^{-(2+1/\alpha)}\bigr)$. In contrast, MLMC uses a series of coarse levels with larger time steps to construct an estimator. MLMC typically reduces the cost of the method to $\mathrm{Cost}^{(\mathrm{MLMC})}(\epsilon) = \mathcal{O}(M) = \mathcal{O}(\epsilon^{-2})$, which is the lower limit that can be achieved with a Monte Carlo method.

While MLMC is more efficient than standard Monte Carlo in the limit $\epsilon \to 0$, the actual value of the tolerance ϵ might be relatively large in practical applications. Hence, not only the

asymptotic rate of convergence, but also the cost of the method for a given ϵ is of interest. The exact value of the constant C_2 in the cost function $\mathrm{Cost}^{(\mathrm{MLMC})}(\epsilon) = C_2\epsilon^{-2} + \cdots$ and the size of higher-order corrections depends on the details of the method, such as the time-stepping scheme and coarse-level solver. In particular, choosing a time-stepping scheme that becomes unstable on the coarser levels can severely limit the performance as only a small number of levels can be used; see (Abdulle & Blumenthal, 2013; Hutzenthaler *et al.*, 2013).

Suppose that we are interested in the expectation of $\mathcal{P} \coloneqq \phi(\boldsymbol{X}(T))$, where $\boldsymbol{X}(T)$ is the solution to Eq. (1.1) at time T and $\phi\colon \mathbb{R}^d \to \mathbb{R}$ defines the quantity of interest. Assume that the number of time steps used to discretise the SDE is $N = N_{\mathrm{coarse}}2^L$, where $N_{\mathrm{coarse}}, L \in \mathbb{N}$. Our strategy is to approximate Eq. (1.1) using a numerical integrator with time step $h = T/N$ to define an approximate solution $\boldsymbol{X}_N \approx \boldsymbol{X}(T)$. Then, we compute many independent samples of $\boldsymbol{X}_N$ to define approximate samples $\mathcal{P}^{(i)}$ of $\mathcal{P}$. The classical Monte Carlo method approximates $\mathbb{E}[\mathcal{P}]$ by the sample average of $\mathcal{P}^{(i)}$.

Instead, the MLMC method constructs a sequence of approximations on levels indexed by $\ell = \{L, L-1, \ldots, \ell_0\}$ with $N_\ell = N_{\mathrm{coarse}}2^\ell$ time steps of size $h_\ell = T/N_\ell$. Let $\mathcal{P}_\ell^{(i)}$ denote independent samples of the approximation to $\mathcal{P}$ on level ℓ and let

$$\widehat{\mathcal{P}}_\ell \coloneqq \frac{1}{M_\ell}\sum_{i=1}^{M_\ell} \mathcal{P}_\ell^{(i)} \tag{2.1}$$

denote the Monte Carlo estimator on level ℓ based on M_ℓ samples. An estimator for the finest level where $N = N_L$ can be written as the telescoping sum

$$\widehat{\mathcal{P}}^{(\mathrm{MLMC})} \equiv \widehat{Y}_{\{M_\ell\}} \coloneqq \widehat{Y}_{\ell_0, M_{\ell_0}} + \sum_{\ell=\ell_0+1}^{L} \widehat{Y}_{\ell, M_\ell}, \tag{2.2}$$

where $\widehat{Y}_{\ell_0, M_{\ell_0}} \coloneqq \widehat{\mathcal{P}}_{\ell_0}$ and

$$\widehat{Y}_{\ell, M_\ell} \coloneqq \frac{1}{M_\ell}\sum_{i=1}^{M_\ell} Y_\ell^{(i)}, \qquad Y_\ell^{(i)} \coloneqq \mathcal{P}_\ell^{(i)} - \mathcal{P}_{\ell-1}^{(i)} \quad \text{for } \ell = \ell_0+1, \ldots, L. \tag{2.3}$$

The estimator does not introduce any additional bias, as we recover the numerical discretisation error on the finest level where $h = h_L$:

$$\mathbb{E}\Big[\widehat{\mathcal{P}}^{(\mathrm{MLMC})}\Big] = \mathbb{E}\Big[\widehat{Y}_{\{M_\ell\}}\Big] = \mathbb{E}\Big[\widehat{\mathcal{P}}_L\Big] = \mathbb{E}\Big[\widehat{\mathcal{P}}^{(\mathrm{MC})}\Big], \tag{2.4}$$

where $\widehat{\mathcal{P}}^{(\mathrm{MC})}$ is the standard Monte Carlo estimator for $N = N_L$ time steps. The two key ideas of the MLMC method are now:

- The number of time steps N_ℓ is smaller on the coarser levels $\ell < L$. Hence, the calculation of a single sample $\mathcal{P}_\ell^{(i)}$ is substantially cheaper.
- The success of the method depends on coupling the samples $\mathcal{P}_\ell^{(i)}$ and $\mathcal{P}_{\ell-1}^{(i)}$ so that the variance of $Y_\ell^{(i)} = \mathcal{P}_\ell^{(i)} - \mathcal{P}_{\ell-1}^{(i)}$ is small. By arranging for the variance of $Y_\ell^{(i)}$ to be small, a smaller number M_ℓ of samples are necessary to construct an accurate estimator $\widehat{Y}_{\ell, M_\ell}$. This allows the calculation of a MLMC estimator with fixed total variance $\sum_{\ell=\ell_0}^{L} \mathrm{Var}\left[\widehat{Y}_\ell\right]/M_\ell$ for less computational cost.

This is formalised in the following complexity theorem (Giles, 2008, Theorem 3.1):

Theorem 2.1 (MLMC complexity)**.** *Consider a real-valued random variable $\mathcal{P}$ and estimators $\widehat{\mathcal{P}}_\ell$ corresponding to a numerical approximation based on time step $h_\ell = T/N_\ell$ and M_ℓ samples. If there exist independent estimators $\widehat{Y}_{\ell, M_\ell}$ based on M_ℓ Monte Carlo samples, and positive constants $\alpha \geq \frac{1}{2}$, c_1, c_2, c_3 such that*

(i) $\left|\mathbb{E}\left[\widehat{\mathcal{P}}_\ell - \mathcal{P}\right]\right| \le c_1 h_\ell^\alpha$,

(ii) $\mathbb{E}\left[\widehat{Y}_{\ell,M_\ell}\right] = \begin{cases} \mathbb{E}\left[\widehat{\mathcal{P}}_{\ell_0}\right], & \ell = \ell_0, \\ \mathbb{E}\left[\widehat{\mathcal{P}}_\ell - \widehat{\mathcal{P}}_{\ell-1}\right], & \ell > \ell_0, \end{cases}$

(iii) $\mathrm{Var}\left[\widehat{Y}_{\ell,M_\ell}\right] \le c_2 M_\ell^{-1} h_\ell^2$, *and*

(iv) $\mathrm{Cost}_\ell^{(\mathrm{MLMC})}$, *the computational complexity of* $\widehat{Y}_\ell$, *is bounded by* $c_3 M_\ell h_\ell^{-1}$,

then there exists a positive constant c_4 *such that for any* $\epsilon < 1/e$, *there are values* L *and* M_ℓ *for which* $\widehat{Y}_{\{M_\ell\}}$ *from Eq.* (2.2) *has a mean-square error (MSE) with bound*

$$MSE \equiv \mathbb{E}\left[\left(\widehat{Y}_{\{M_\ell\}} - \mathbb{E}\left[\mathcal{P}\right]\right)^2\right] < \epsilon^2 \tag{2.5}$$

and a computational complexity $\mathrm{Cost}^{(\mathrm{MLMC})}$ *with bound*

$$\mathrm{Cost}^{(\mathrm{MLMC})} \le c_4 \epsilon^{-2}. \tag{2.6}$$

The theorem can be extended to allow the variance to decay as $\mathrm{Var}\,\widehat{Y}_\ell \le c_2 M_\ell^{-1} h_\ell^\beta$ (Giles, 2008). For all cases in this paper, $\beta = 2$ and the cost is concentrated on the coarsest level (as we see from Algorithm 1 line 12 and (iii) above). The asymptotic dependence of the computational complexity on ϵ is independent of the weak order of convergence α of the time-stepping method. However, the constant c_4 does depend on the particular time-stepping method.

To obtain the results in this paper, we used Algorithm 1 and our choices for the numbers of samples M_ℓ on each of the levels are defined there. Given a tolerance $\epsilon_{\max} > 0$, the algorithm gives an MLMC estimator $\widehat{\mathcal{P}}^{(\mathrm{MLMC})}$ with mean-square error ϵ in the range $\epsilon_{\max}/2 < \epsilon < \epsilon_{\max}$ as defined in Eq. (2.5).

3 Applying the complexity theorem

Our goal is to apply the complexity theorem to numerical integrators using only weak-approximation properties of the numerical methods. The complexity theorem makes assumptions on (i) the bias, (ii) the consistency of the estimators, and (iii) the variance of the corrections. (i) can be understood from existing weak-convergence analysis. Let $C^\infty_{\mathrm{poly}}(\mathbb{R}^d)$ be the set of infinitely differentiable functions $\mathbb{R}^d \to \mathbb{R}$ such that all derivatives are polynomially bounded.

Definition 3.1. For a time step $h > 0$, let $\boldsymbol{X}_n$ be a $\mathbb{R}^d$-valued random variable that approximates the solution $\boldsymbol{X}(t)$ to Eq. (1.1) at time $t = nh$. We say $\boldsymbol{X}_n$ is a *weak order-α* approximation if, for all $\phi \in C^\infty_{\mathrm{poly}}(\mathbb{R}^d)$ and $T > 0$, there exists $K > 0$ such that for h sufficiently small

$$\left|\mathbb{E}\left[\phi(\boldsymbol{X}(T))\right] - \mathbb{E}\left[\phi(\boldsymbol{X}_N)\right]\right| \le K h^\alpha, \qquad Nh = T.$$

There are many integrators that provide weak order-α approximations for $\alpha = 1$ or $\alpha = 2$ (e.g., Kloeden & Platen (1992) or §4). In the case that $\mathcal{P} = \phi(\boldsymbol{X}(T))$, $T = Nh_\ell$, and $\widehat{\mathcal{P}}_\ell = \phi(\boldsymbol{X}_N)$ for an $\boldsymbol{X}_N$ with step h_ℓ that is weak α-order, the bias condition (i) holds.

The consistency of the estimators (ii) is an easy consequence of the linearity of integration and Eq. (2.3).

Condition (iii) on the variance of corrections normally follows from the mean-square convergence of the integrator (Giles, 2008). Mean-square convergence measures the approximation of individual sample paths of the solution $\boldsymbol{X}(t)$ and hence is a tool for understand the coupling of successive levels. In this paper, we use an alternative method based on weak-approximation

Algorithm 1 Multilevel Monte Carlo. Input: $\epsilon_{\max}$, ℓ_0, and T. Output: Estimator $\widehat{\mathcal{P}}^{(\mathrm{MLMC})}$

Choose L, N_{coarse} such that, on the finest level with $N_L = 2^L N_{\mathrm{coarse}}$ time steps of size $h_L = T/N_L$, the bias ϵ_{bias} is smaller than $\epsilon_{\max}/\sqrt{2}$. Define $\epsilon \equiv \sqrt{2}\epsilon_{\mathrm{bias}}$.
Choose a minimum number of samples $M_{\min}$ (say 100 or 1000).
Set $M_\ell^- = 1$, $M_\ell^+ = M_{\min}$, $M_\ell = 0$ for all levels ℓ.
while $M_\ell < M_\ell^+$ for some level ℓ **do**
  for $\ell = L, \ldots, \ell_0$ **do**
    for $i = M_\ell^-, \ldots, M_\ell^+$ **do**
      Calculate $Y_\ell^{(i)}$ by applying the numerical integrator on levels ℓ and $\ell - 1$ (only on level ℓ_0 on the coarsest level). The integrators on level ℓ and $\ell - 1$ should be coupled (see §4.4).
      $M_\ell \mapsto M_\ell + 1$.
    end for
    Update estimators for the bias and variance:

$$\widehat{Y}_{\ell,M_\ell} = \frac{1}{M_\ell}\sum_{i=1}^{M_\ell} Y_\ell^{(i)}, \quad \widehat{V}_{\ell,M_\ell} = \frac{1}{M_\ell - 1}\left[\sum_{i=1}^{M_\ell}\left(Y_\ell^{(i)}\right)^2 - \frac{1}{M_\ell}\left(\sum_{i=1}^{M_\ell} Y_\ell^{(i)}\right)^2\right].$$

    $M_\ell^- = M_\ell^+ + 1$.
    Calculate the optimal M_ℓ^+ according to formula (12) in (Giles, 2008):

$$M_\ell^+ = \left\lceil 2\epsilon^{-2}\sqrt{\widehat{V}_{\ell,M_\ell} h_\ell}\sum_{j=\ell_0}^{L}\sqrt{\widehat{V}_{j,M_\ell}/h_j}\right\rceil.$$

  end for
end while
Return estimator $\widehat{\mathcal{P}}^{(\mathrm{MLMC})} = \widehat{Y}_{\{M_\ell\}} \equiv \sum_{\ell=\ell_0}^{L} \widehat{Y}_{\ell,M_\ell}$

theory and derive condition (iii) as a consequence of the existence of a second-order modified equation. To do this, we introduce the following doubled-up system for $\boldsymbol{Z} = [\boldsymbol{X}, \boldsymbol{Y}] \in \mathbb{R}^{2d}$:

$$\begin{aligned} d\boldsymbol{X} &= \boldsymbol{f}(\boldsymbol{X})\,dt + G(\boldsymbol{X})\,d\boldsymbol{W}(t), \qquad & \boldsymbol{X}(0) &= \boldsymbol{X}_0 \in \mathbb{R}^d, \\ d\boldsymbol{Y} &= \boldsymbol{f}(\boldsymbol{Y})\,dt + G(\boldsymbol{Y})\,d\boldsymbol{W}(t), \qquad & \boldsymbol{Y}(0) &= \boldsymbol{X}_0. \end{aligned} \tag{3.1}$$

The same initial data is applied and the same $\boldsymbol{W}(t)$ drives both components and so $\boldsymbol{X}(t) = \boldsymbol{Y}(t)$ a.s. for $t > 0$. We now have two copies of $\boldsymbol{X}(t)$ and we approximate each differently. Formally, we approximate $\boldsymbol{X}(t)$ and $\boldsymbol{Y}(t)$ by different numerical integrators with step $h > 0$ and denote the resulting approximation to $\boldsymbol{Z}(t_n)$ by $\boldsymbol{Z}_n = [\boldsymbol{X}_n, \boldsymbol{Y}_n]$ at $t_n = nh$. In MLMC, there is usually one integrator applied with time steps h for $\boldsymbol{X}$ and $h/2$ for $\boldsymbol{Y}$ (which is a little awkward for $\boldsymbol{Y}_n$, as one increment of n corresponds to two steps of the underlying integrator). The joint distribution of the $\boldsymbol{X}_n$ and $\boldsymbol{Y}_n$ contains all the required information about the coupling of the approximations of each component and, as we now show, a weak-convergence analysis of the system gives condition (ii).

For simplicity, we start by assuming that $\boldsymbol{Z}_n = [\boldsymbol{X}_n, \boldsymbol{Y}_n]$ is a weak second-order approximation to $\boldsymbol{Z}(t) = [\boldsymbol{X}(t), \boldsymbol{Y}(t)]$. Then, we can prove the following.

Theorem 3.2. *Fix $T > 0$ and let $\mathcal{P} = \phi(\boldsymbol{X}(T))$ for a $\phi \in C^\infty_{\text{poly}}(\mathbb{R}^d)$ that is globally Lipschitz continuous. Suppose that $\boldsymbol{Z}_n$ is a weak second-order approximation to $\boldsymbol{Z}(t)$. Conditions (i)–(iii) of Theorem 2.1 hold with $[\mathcal{P}_\ell^{(i)}, \mathcal{P}_{\ell-1}^{(i)}]$ given by* iid *samples of $[\phi(\boldsymbol{X}_N), \phi(\boldsymbol{Y}_N)]$ with $h = h_\ell$ and $Nh = T$.*

Proof. The condition on $\boldsymbol{Z}_n$ implies also that $\boldsymbol{X}_n$ and $\boldsymbol{Y}_n$ are weak second-order approximations to $\boldsymbol{X}(t)$. Then, by the above discussion, conditions (i) with $\alpha = 2$ and (ii) hold.

Let $\psi(\boldsymbol{Z}) \coloneqq \phi(\boldsymbol{X}) - \phi(\boldsymbol{Y})$. Then $\psi^2 \in C^\infty_{\text{poly}}(\mathbb{R}^{2d})$ since ϕ and hence ψ^2 are smooth and their derivatives are polynomially bounded. As $\boldsymbol{Z}_n$ is a weak second-order approximation to $\boldsymbol{Z}(t)$,

$$\mathbb{E}\big[\psi(\boldsymbol{Z}_N)^2 - \psi(\boldsymbol{Z}(T))^2\big] = \mathcal{O}\big(h^2\big).$$

By definition of ψ,

$$\mathbb{E}\Big[\big|\phi(\boldsymbol{X}_N) - \phi(\boldsymbol{Y}_N)\big|^2 - \big|\phi(\boldsymbol{X}(T)) - \phi(\boldsymbol{Y}(T))\big|^2\Big] = \mathcal{O}\big(h^2\big). \tag{3.2}$$

Using the fact that $\boldsymbol{X}(t) = \boldsymbol{Y}(t)$ a.s., we have

$$\mathbb{E}\Big[\big|\phi(\boldsymbol{X}_N) - \phi(\boldsymbol{Y}_N)\big|^2\Big] = \mathcal{O}\big(h^2\big).$$

Written in terms of $\mathcal{P}_\ell$ and $\mathcal{P}_{\ell-1}$, this means

$$\operatorname{Var}\big[\mathcal{P}_\ell - \mathcal{P}_{\ell-1}\big] \le \mathbb{E}\Big[\big(\mathcal{P}_\ell - \mathcal{P}_{\ell-1}\big)^2\Big] = \mathcal{O}\big(h^2\big).$$

In other words, the variance of each sample of the coarse–fine correction is order h^2. This implies that the sample average $\widehat{Y}_{\ell,N_\ell}$ of M_ℓ *iid* samples satisfies condition (iii) of Theorem 2.1. □

3.1 Modified equations

The above argument does not apply to weak first-order accurate methods, even though the complexity theorem only requires $\alpha > 1/2$. In this case, we use the theory of modified equations to extend the analysis. A modified equation is a small perturbation of the original SDE that the numerical method under consideration approximates more accurately. For the theory, we need a second-order modified equation for the doubled-up system and this contains second-order information about the coupling of the fine and coarse levels. In particular, we consider modified equations for the double-up system (3.1) of the form:

$$\begin{aligned} d\boldsymbol{X}_h &= \big[\boldsymbol{f}(\boldsymbol{X}_h) + h\boldsymbol{f}_1(\boldsymbol{X}_h)\big]\,dt + \big[G(\boldsymbol{X}) + hG_1(\boldsymbol{X}_h)\big]\,d\boldsymbol{W}(t), \qquad & \boldsymbol{X}(0) &= \boldsymbol{X}_0, \\ d\boldsymbol{Y}_h &= \big[\boldsymbol{f}(\boldsymbol{Y}_h) + h\boldsymbol{f}_2(\boldsymbol{Y}_h)\big]\,dt + \big[G(\boldsymbol{Y}_h) + hG_2(\boldsymbol{Y}_h)\big]\,d\boldsymbol{W}(t), \qquad & \boldsymbol{Y}(0) &= \boldsymbol{X}_0, \end{aligned} \tag{3.3}$$

for $\boldsymbol{f}_i\colon \mathbb{R}^d \to \mathbb{R}^d$ and $\boldsymbol{G}_i\colon \mathbb{R}^d \to \mathbb{R}^{d\times m}$ for $i = 1, 2$. (This could be extended to allow $\boldsymbol{f}_i, G_i$ to depend on both $\boldsymbol{X}_h$ and $\boldsymbol{Y}_h$.) When the same integrator is used for each component, but with time steps h and $h/2$, it must hold that $\boldsymbol{f}_2 = \boldsymbol{f}_1/2$ and $G_2 = G_1/2$. We show in Theorem 3.4 that, subject to regularity conditions on the coefficients, the MLMC complexity theorem applies if a second-order modified equation exists and therefore MLMC works with $\mathcal{O}(\epsilon^{-2})$ complexity.

The additional difficulty is that $\boldsymbol{X}_h \neq \boldsymbol{Y}_h$ and we must estimate the variance of $\phi(\boldsymbol{X}_h) - \phi(\boldsymbol{Y}_h)$. We use a mean-square analysis and the following lemma, which gives a first-order $L^2(\Omega, \mathbb{R}^d)$ bound on $\boldsymbol{Z}(t) - \boldsymbol{Z}_h(t)$. The lemma requires a number of regularity assumptions on the coefficients of the modified equation, which hold, for example, if $\boldsymbol{f}, \boldsymbol{f}_i$ and G, G_i are globally Lipschitz continuous.

Lemma 3.3. *For $t \in [0, T]$, let $\boldsymbol{Z}(t)$ satisfy the Ito SDE (3.1) and $\boldsymbol{Z}_h(t) = [\boldsymbol{X}_h(t), \boldsymbol{Y}_h(t)]$ satisfy the modified equation (3.3). Suppose that*

(i) *$\boldsymbol{f}\colon \mathbb{R}^d \to \mathbb{R}^d$ and $G\colon \mathbb{R}^d \to \mathbb{R}^{d\times m}$ are globally Lipschitz continuous with Lipschitz constant $L > 0$.*

(ii) *There exists $C_1 > 0$ such that, for all $h > 0$ sufficiently small,*

$$\mathbb{E}\Big[\big\|\boldsymbol{f}_1(\boldsymbol{X}_h(s))\big\|^2\Big], \mathbb{E}\Big[\big\|G_1(\boldsymbol{X}_h(s))\big\|_{\mathrm{F}}^2\Big] \leq C_1,$$
$$\mathbb{E}\Big[\big\|\boldsymbol{f}_2(\boldsymbol{Y}_h(s))\big\|^2\Big], \mathbb{E}\Big[\big\|G_2(\boldsymbol{Y}_h(s))\big\|_{\mathrm{F}}^2\Big] \leq C_1, \qquad s \in [0, T],$$

where $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius norm.

Then, if $\psi\colon \mathbb{R}^{2d} \to \mathbb{R}$ is globally Lipschitz continuous, we have, for some constant $C_2 > 0$ independent of h,

$$\mathbb{E}\Big[\big|\psi(\boldsymbol{Z}(t)) - \psi(\boldsymbol{Z}_h(t))\big|^2\Big] \leq C_2 h^2, \qquad \text{for } t \in [0, T].$$

Proof. This is an elementary calculation with the Gronwall inequality and Ito isometry. See Appendix A. □

We are now able to state and prove the main theorem of this article.

Theorem 3.4. *Fix $T > 0$. Let $\phi \in C^\infty_{\mathrm{poly}}(\mathbb{R}^d)$ be globally Lipschitz continuous. Suppose that*

(i) *$\boldsymbol{X}_n$ and $\boldsymbol{Y}_n$ are weak order-α approximations to $\boldsymbol{X}(t)$ for some $\alpha > 1/2$,*

(ii) *$\boldsymbol{Z}_n$ are second-order approximations to $\boldsymbol{Z}_h(t)$, and*

(iii) *the assumptions of Lemma 3.3 hold.*

Then Conditions (i)–(iii) of Theorem 2.1 hold with $[\mathcal{P}_\ell^{(i)}, \mathcal{P}_{\ell-1}^{(i)}]$ given by iid *samples of $[\phi(\boldsymbol{X}_N), \phi(\boldsymbol{Y}_N)]$ with $h = h_\ell$.*

Proof. As before, conditions (i) and (ii) are straightforward. It is the third condition, which normally follows from a strong-approximation theory, that requires the modified equation. Let $\psi(\boldsymbol{Z}) \coloneqq \phi(\boldsymbol{X}) - \phi(\boldsymbol{Y})$ and note that $\psi^2 \in C^\infty_{\mathrm{poly}}(\mathbb{R}^{2d})$. As $\boldsymbol{Z}_n$ is a second-order weak approximation to $\boldsymbol{Z}_h(t)$, we have

$$\mathbb{E}\big[\psi(\boldsymbol{Z}_N)^2 - \psi(\boldsymbol{Z}_h(T))^2\big] = \mathcal{O}\big(h^2\big).$$

By definition of ψ,

$$\mathbb{E}\Big[\big|\phi(\boldsymbol{X}_N) - \phi(\boldsymbol{Y}_N)\big|^2 - \big|\phi(\boldsymbol{X}_h(T)) - \phi(\boldsymbol{Y}_h(T))\big|^2\Big] = \mathcal{O}\big(h^2\big). \tag{3.4}$$

Using the fact that $\boldsymbol{X}(t) = \boldsymbol{Y}(t)$ a.s.,

$$\begin{aligned}\mathbb{E}\Big[\big|\phi(\boldsymbol{X}_h(T)) - \phi(\boldsymbol{Y}_h(T))\big|^2\Big] &= \mathbb{E}\Big[\big|\phi(\boldsymbol{X}_h(T)) - \phi(\boldsymbol{X}(T)) + \phi(\boldsymbol{Y}(T)) - \phi(\boldsymbol{Y}_h(T))\big|^2\Big] \\ &= \mathbb{E}\Big[\big|\psi(\boldsymbol{Z}_h(T)) - \psi(\boldsymbol{Z}(T))\big|^2\Big].\end{aligned}$$

Lemma 3.3 applies and the right-hand side in the last equation is $\mathcal{O}(h^2)$. Consequently,

$$\mathbb{E}\Big[\big|\phi(\boldsymbol{X}_h(T)) - \phi(\boldsymbol{Y}_h(T))\big|^2\Big] = \mathcal{O}\big(h^2\big). \tag{3.5}$$

Together, Eqs. (3.4) and (3.5) imply that

$$\mathbb{E}\Big[\big|\phi(\boldsymbol{X}_N) - \phi(\boldsymbol{Y}_N)\big|^2\Big] = \mathcal{O}\big(h^2\big).$$

The remainder of the proof is the same as for Theorem 3.2. □

In summary, subject to smoothness conditions, if MLMC is applied with an integrator that has a second-order modified equation like Eq. (3.3) then the variance of the coarse–fine correction is $\mathcal{O}(h^2)$ and the complexity of MLMC is $\mathcal{O}(\epsilon^{-2})$. Though the rate is fixed, the complexity of MLMC depends on the specific integrator used through the constant and, as we now show, this leads to large variations in efficiency.

4 Application to the Langevin equation

Before showing how they can be used for MLMC, we introduce several integrators for the Langevin equation.

4.1 Splitting methods

Splitting methods are an important class of numerical integrators for differential equations. In the case of ODEs, they allow the vector field to be broken down into meaningful parts and integrated separately over a single time step, before combining into an integrator for the full vector field. See for example (Hairer *et al.*, 2010; Leimkuhler & Reich, 2004). The Langevin equation breaks down into the sum of a Hamiltonian system and a linear SDE for an Ornstein–Uhlenbeck (OU) process. Then, for a splitting method, we define integrators for the Hamiltonian system

$$\begin{aligned} \frac{d\boldsymbol{Q}}{dt} &= \boldsymbol{P}, \\ \frac{d\boldsymbol{P}}{dt} &= -\nabla V(\boldsymbol{Q}), \end{aligned} \tag{4.1}$$

which should be a symplectic integrator, and for the OU process

$$d\boldsymbol{P} = -\lambda \boldsymbol{P}\, dt + \sigma\, d\boldsymbol{W}(t). \tag{4.2}$$

The OU process can be integrated exactly and we use this fact to define a so-called geometric integrator for Eq. (4.2). It is clear that the sum of the right-hand sides of these two systems gives Eq. (1.5). There are a number of ways of combining integrators of Eqs. (4.1) and (4.2) to define an integrator of the full system. The simplest, also known as the Lie–Trotter splitting, is to simulate Eqs. (4.1) and (4.2) alternately on time intervals of length h. In general, this technique can only be first-order accurate in the weak sense. Alternatively, if the underlying integrators are second order, we can define a second-order splitting method by applying Eq. (4.2) on a half step, then Eq. (4.1) for a full step, and finally apply again Eq. (4.2) on a half step. This is called the symmetric Strang splitting.

We now define specific integrators for Eqs. (4.1) and (4.2). Eq. (4.1) is a separable Hamiltonian system, and the symplectic Euler method and Störmer–Verlet methods provide simple, explicit, and stable methods for its numerical solution. The symplectic Euler method is first-order accurate and the Störmer–Verlet method is second-order accurate.

The solution of Eq. (4.2) is a multi-dimensional OU process and can be written as

$$\boldsymbol{P}(t) = \mathrm{e}^{-\lambda t}\boldsymbol{P}(0) + \sigma \boldsymbol{I}(0,t), \tag{4.3}$$

where

$$\boldsymbol{I}(t_1, t_2) \coloneqq \int_{t_1}^{t_2} \mathrm{e}^{-\lambda(t_2-s)}\, d\boldsymbol{W}(s).$$

Each component of $\boldsymbol{I}$ is *iid* with mean zero and variance

$$\operatorname{Var} I_i(t_1, t_2) = \int_{t_1}^{t_2} \mathrm{e}^{-2\lambda(t_2-s)}\, ds = \frac{1 - e^{-2\,\lambda\,(t_2-t_1)}}{2\,\lambda}, \tag{4.4}$$

so that $\boldsymbol{I}(t_1, t_2) \sim \mathrm{N}(\boldsymbol{0}, \alpha^2_{t_2-t_1} I)$ for $\alpha_t \coloneqq \sqrt{(1-\mathrm{e}^{-2\lambda t})/2\lambda}$. This suggests taking the following as the numerical integrator: for a time step $h > 0$,

$$\boldsymbol{P}_{n+1} = \mathrm{e}^{-\lambda h}\boldsymbol{P}_n + \sigma\alpha_h\boldsymbol{\xi}_n \tag{4.5}$$

for $\boldsymbol{\xi}_n \sim \mathrm{N}(0, I)$ *iid*. If $\boldsymbol{P}_n = \boldsymbol{P}(t_n)$, then $\boldsymbol{P}_{n+1}$ has the same distribution as $\boldsymbol{P}(t_{n+1})$ and this method is exact in the sense of distributions. Methods of this type, where the variation of constants formula (4.3) is used for the discretisation, are often called geometric integrators.

The full equations for the first order splitting (symplectic Euler) and second-order splitting (Störmer–Verlet) are written as follows:

Symplectic Euler/OU For $\boldsymbol{\xi}_n$ *iid* with distribution $\mathrm{N}(\boldsymbol{0}, I)$ or its three-point approximation,

$$\begin{aligned}
\boldsymbol{P}^*_{n+1} &= \mathrm{e}^{-\lambda h}\boldsymbol{P}_n + \sigma\alpha_h\boldsymbol{\xi}_n,\\
\boldsymbol{P}_{n+1} &= \boldsymbol{P}^*_{n+1} - h\,\nabla V(\boldsymbol{Q}_n),\\
\boldsymbol{Q}_{n+1} &= \boldsymbol{Q}_n + \boldsymbol{P}_{n+1}\,h.
\end{aligned} \tag{4.6}$$

Störmer–Verlet/OU For $\boldsymbol{\xi}_n, \boldsymbol{\xi}_{n+1/2}$ *iid* with distribution $\mathrm{N}(\boldsymbol{0}, I)$ or its three-point approximation,

$$\begin{aligned}
\boldsymbol{P}^*_{n+1/2} &= \mathrm{e}^{-\lambda h/2}\boldsymbol{P}_n + \sigma\alpha_{h/2}\boldsymbol{\xi}_n,\\
\boldsymbol{P}_{n+1/2} &= \boldsymbol{P}^*_{n+1/2} - \frac{1}{2}h\,\nabla V(\boldsymbol{Q}_n),\\
\boldsymbol{Q}_{n+1} &= \boldsymbol{Q}_n + h\boldsymbol{P}_{n+1/2},\\
\boldsymbol{P}^*_{n+1} &= \boldsymbol{P}_{n+1/2} - \frac{1}{2}h\,\nabla V(\boldsymbol{Q}_{n+1}),\\
\boldsymbol{P}_{n+1} &= \mathrm{e}^{-\lambda h/2}\boldsymbol{P}^*_{n+1} + \sigma\alpha_{h/2}\boldsymbol{\xi}_{n+1/2}.
\end{aligned} \tag{4.7}$$

4.2 Modified equations for the Langevin equation

Consider the Langevin equation (1.5). Using a computer algebra system to verify consistency of moments to fifth order, it is easy to find modified equations for the numerical integrators developed in §4.1. For example, for the first-order splitting method with $d = 1$, the doubled-up modified equation is as follows: Denote by $[Q_n, P_n]$ the numerical approximation on the coarse level (step h) and $[q_n, p_n]$ on the fine level (step $h/2$). The second-order modified equation is

$$\begin{aligned}
dQ &= \Big[P - \frac{1}{2}h\big(V'(Q) + \lambda P\big)\Big]\,dt + \sigma\frac{1}{2}h\,dW(t),\\
dP &= \Big[-\lambda P - V'(Q) - \frac{1}{2}h\big(\lambda V'(Q) - PV''(Q)\big)\Big]\,dt + \sigma\,dW(t),\\
dq &= \Big[p - \frac{1}{4}h\big(V'(q) + \lambda p\big)\Big]\,dt + \sigma\frac{1}{4}h\,dW(t),\\
dp &= \Big[-\lambda p - V'(q) - \frac{1}{4}h\big(\lambda V'(q) - pV''(q)\big)\Big]\,dt + \sigma\,dW(t),
\end{aligned} \tag{4.8}$$

where $W(t)$ is the same Brownian motion for p and P. We conclude then that this method leads to $\mathcal{O}(h^2)$ variances in the coarse–fine correction, if the coefficients are sufficiently well

behaved. Identifying when the coefficients are well behaved is hard. For example, it is sufficient that the drift and diffusion in both the original and modified equations are globally Lipschitz. These however are very strong conditions and do not hold for many realistic potentials.

For the second-order splitting method (based on Störmer–Verlet method and exact OU integration), we can apply Theorem 3.2 to see that the variance of the coarse–fine corrections is $\mathcal{O}(h^2)$. The regularity condition is on the original drift and diffusion and holds if $\nabla V \colon \mathbb{R}^d \to \mathbb{R}^d$ is sufficiently smooth (e.g., infinitely differentiable and Lipschitz).

4.3 Random variables with discrete distribution

The modified equations are unchanged if the Gaussian random variables in the integrator are replaced by random variables with the same moments to order five. For example, we can replace samples of *iid* $\mathrm{N}(0,1)$ random variables by *iid* samples of the random variable ζ with distribution

$$\mathbb{P}\big(\zeta = 0\big) = \frac{2}{3}, \qquad \mathbb{P}\big(\zeta = \pm\sqrt{3}\big) = \frac{1}{6}. \tag{4.9}$$

We refer to ζ as the three-point approximation to the Gaussian. This is a well-known trick for weak approximation of SDEs, for example (Kloeden & Platen, 1992, §14.2). The three-point approximation has a number of advantages, as ζ is quicker to sample than a Gaussian and, due to the finite number of states, averages of functionals of ζ can be computed exactly. As discussed in §4.5, this allows exact integration on the coarsest level, which reduces the computational cost of MLMC dramatically.

4.4 MLMC with splitting methods

Let $\boldsymbol{X} = [\boldsymbol{Q}, \boldsymbol{P}]$ denote the state-space variable. A key step in MLMC is computing approximations to $\boldsymbol{X}(t_{n+2})$ at $t_n = nh$ given $\boldsymbol{X}(t_n)$ based on integrators with time steps $h/2$ and h that are coupled so the difference between the approximations has small variance. For the Euler–Maruyama method, this is achieved by choosing increments $\Delta\boldsymbol{W}_n, \Delta\boldsymbol{W}_{n+1}$ for the computation with time step $h/2$, and choosing the sum $\Delta\boldsymbol{W}_n + \Delta\boldsymbol{W}_{n+1}$ for the corresponding interval of the computation with time step h.

It is hard to sample $\boldsymbol{I}(0,t)$ in Eq. (4.5) based on increments of the particular sample path of $\boldsymbol{W}(t)$ and, as a method for strong approximation, it is limited. It is easy however to sample $\boldsymbol{I}(0,t)$ as a Gaussian random variable and enough for many purposes to sample its three-point approximation. We now show how to couple fine–coarse integrators for the MLMC method, without the direct link to the increment. First, note that

$$\begin{aligned}
\boldsymbol{I}(0,2h) &= \int_0^{2h} \mathrm{e}^{-\lambda(2h-s)}\, d\boldsymbol{W}(s) \\
&= \int_0^{h} \mathrm{e}^{-\lambda(2h-s)}\, d\boldsymbol{W}(s) + \int_h^{2h} \mathrm{e}^{-\lambda(2h-s)}\, d\boldsymbol{W}(s) \\
&= rI(0,h) + I(h,2h), \qquad r \coloneqq \mathrm{e}^{-\lambda h}.
\end{aligned}$$

By Eq. (4.4),

$$\boldsymbol{I}(0,h), \boldsymbol{I}(h,2h) \sim \mathrm{N}\big(0, \alpha_h^2 I\big) \ \textit{iid}.$$

We can simulate $\boldsymbol{I}(0,h)$ and $\boldsymbol{I}(0,2h)$, by generating $\boldsymbol{\xi}_i \sim \mathrm{N}(\boldsymbol{0}, I)$ *iid* and computing

$$\boldsymbol{I}(0,h) = \alpha_h \boldsymbol{\xi}_1, \qquad \boldsymbol{I}(h,2h) = \alpha_h \boldsymbol{\xi}_2.$$

As $\alpha_h^2 = (1-r^2)/2\lambda$ and $\alpha_{2h}^2 = (1-r^4)/2\lambda$, we have $\alpha_h^2(1+r^2) = \alpha_{2h}^2$. Then,

$$\boldsymbol{I}(0,2h) = \alpha_{2h} \frac{1}{\sqrt{1+r^2}} (r\boldsymbol{\xi}_1 + \boldsymbol{\xi}_2).$$

Given $\boldsymbol{P}_n$ at time t_n, we find $\boldsymbol{P}_{n+2}$ using two time steps of size h by

$$\boldsymbol{P}_{n+1} = \mathrm{e}^{-\lambda h}\boldsymbol{P}_n + \sigma\alpha_h\boldsymbol{\xi}_n$$
$$\boldsymbol{P}_{n+2} = \mathrm{e}^{-\lambda h}\boldsymbol{P}_{n+1} + \sigma\alpha_h\boldsymbol{\xi}_{n+1},$$

for $\boldsymbol{\xi}_n \sim \mathrm{N}(0, I)$ *iid*. This is equivalent to a single time step of size $2h$ and

$$\boldsymbol{P}_{n+2} = \mathrm{e}^{-2\lambda h}\boldsymbol{P}_n + \sigma\alpha_{2h}\boldsymbol{\xi}_n^*, \qquad \boldsymbol{\xi}_n^* \coloneqq \frac{r\boldsymbol{\xi}_n + \boldsymbol{\xi}_{n+1}}{\sqrt{r^2+1}}.$$

This method is used to generate the random variables when using the splitting methods with MLMC.

4.5 Exact evaluation of the coarse-level expectiation

The evaluation of the coarse-level estimator $\widehat{\mathcal{P}}_{\ell_0}$ with time step $h_{\ell_0} = T/N_{\ell_0}$ is the computationally most expensive part of the MLMC algorithm: even though the number of time steps and hence the number of samples per path is small, a large number of individual paths needs to be evaluated to reduce the variance of the coarse-level estimator. This cost can be reduced dramatically if a discrete distribution as discussed in §4.3 is used for the individual samples $\boldsymbol{\xi}_n$. In this case, a significantly cheaper estimator, which does not rely on Monte Carlo sampling, can be constructed. If the random numbers $\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_{N_{\ell_0}}$ for each path are drawn from the three-point approximation in Eq. (4.9), there is only a finite number $n_{\boldsymbol{\xi}}$ of possible samples $\boldsymbol{\xi}^{(i)} = \{\boldsymbol{\xi}_1^{(i)}, \ldots, \boldsymbol{\xi}_{N_{\ell_0}}^{(i)}\}$, each with associated probability $\mathbb{P}(\boldsymbol{\xi}^{(i)}) = \mathbb{P}(\boldsymbol{\xi}_1 = \boldsymbol{\xi}_1^{(i)}) \cdots \mathbb{P}(\boldsymbol{\xi}_{N_{\ell_0}} = \boldsymbol{\xi}_{N_{\ell_0}}^{(i)})$. The expectation value of the quantity of interest can be calculated exactly on the coarsest level as

$$\widehat{Y}_{\ell_0}^{\text{exact}} = \widehat{\mathcal{P}}_{\ell_0}^{\text{exact}} = \sum_{i=1}^{n} \mathbb{P}(\boldsymbol{\xi} = \boldsymbol{\xi}^{(i)})\mathcal{P}_\ell^{(i)}. \tag{4.10}$$

For each of the N_{ℓ_0} time steps, we need to choose from the 3^d possible values of $\boldsymbol{\xi}_n$, so $n_{\boldsymbol{\xi}} = (3^d)^{N_{\ell_0}}$ is the number of different samples of $\boldsymbol{\xi}$. Since the estimator contains no sampling error, its variance is zero. In Algorithm 1, we can replace $\hat{Y}_{\ell_0,M_{\ell_0}} \mapsto \hat{Y}_{\ell_0}^{\text{exact}}$ and $\hat{V}_{\ell_0,M_{\ell_0}} \mapsto 0$ in lines 10 and 12. Effectively, this implies that the sum in line 12 only runs from $j = \ell_0 + 1$ to L and it is not necessary to evaluate $M_{\ell_0}^+$.

Naively, the computational complexity of evaluating Eq. (4.10) is given by the product of the number of different samples and the number of time steps, $n_{\boldsymbol{\xi}} \times N_{\ell_0} = N_{\ell_0}(3^d)^{N_{\ell_0}}$. However, using a recursive algorithm, the computational complexity can be reduced to the number of nodes in the product-probability tree, which is only $\mathcal{O}(n_{\boldsymbol{\xi}}) = \mathcal{O}((3^d)^{N_{\ell_0}})$ Nevertheless, this still grows exponentially with N_{ℓ_0} (which in turn is proportional to 2^{ℓ_0}) and so Eq. (4.10) is only competitive for small values of ℓ_0 and d. However, exact evaluation can reduce the overall cost of the algorithm dramatically and this is exploited to significant advantage in §5.

5 Numerical experiments

We developed an object-oriented C++ code to compare the performance of different numerical methods for two model problems. The modular structure of the templated code makes it easy to change key components, such as the time-stepping method or random-number distribution, without negative impacts on the performance. The source code is available under the LGPL 3 license as a git repository on `https://bitbucket.org/em459/mlmclangevin`[1].

Key to the choice of parameters in Algorithm 1 is the balance between bias error and statistical error. Assume that the bias error has the form in Theorem 2.1(i) for a proportionality

[1] All enquiries about the code should be addressed to `e.mueller@bath.ac.uk`.

constant c_1 and that the finest level $h_L = T/(N_{\text{coarse}}2^L)$. Then, for a bias error of size $\epsilon/\sqrt{2}$, we require that

$$c_1\left(\frac{T}{N_{\text{coarse}}2^L}\right)^\alpha = \frac{\epsilon}{\sqrt{2}}.$$

Given c_1, α, and T as well as a choice for N_{coarse}, this can be solved to determine ϵ from L or vice versa. The constant c_1 can be approximated by assuming that $\mathbb{E}\Big[\widehat{\mathcal{P}}_\ell - \mathcal{P}\Big] = \tilde{c}_1 h_\ell^\alpha$ for some $\tilde{c}_1 \in \mathbb{R}$, so that

$$\widehat{Y}_{\ell,M_\ell} \approx \tilde{c}_1 h_\ell^\alpha - \tilde{c}_1 h_{\ell-1}^\alpha = \tilde{c}_1(1-2^\alpha)h_\ell^\alpha$$

and calculating $c_1 = |\tilde{c}_1|$ after computing the left-hand side numerically.

The following numerical experiments for the Langevin equation use the following integrators with $\ell_0 = 3$ and $N_{\text{coarse}} = 1$:

EM–G Euler–Maruyama with Gaussian increments as given by Eq. (1.2).

EM Euler–Maruyama with Gaussian increments replaced by their three-point approximation, as defined by Eq. (4.9).

SE First-order splitting method with the symplectic Euler/exact OU and the three-point approximation. See Eq. (4.6).

SV Second-order splitting method with Störmer–Verlet/exact OU and the three-point approximation. See Eq. (4.7).

Richardson extrapolation is a well-known technique for increasing the accuracy of a numerical approximation by computing two approximations with different discretisation parameters and taking a linear combination that eliminates the lowest-order term for the error. Its extension to SDEs was developed by (Talay & Tubaro, 1990) and is particularly convenient for use with MLMC, as MLMC computes approximations on several levels. Thus, we take $\widehat{\mathcal{P}}_L$ and $\widehat{\mathcal{P}}_{L-1}$ and suppose that, for some constants $\tilde{c}_1$ and $\alpha' > \alpha$,

$$\begin{aligned}\mathbb{E}\Big[\widehat{\mathcal{P}}_L\Big] &= \mathbb{E}\big[\mathcal{P}\big] + \tilde{c}_1 h_L^\alpha + \mathcal{O}\Big(h_L^{\alpha'}\Big)\\ \mathbb{E}\Big[\widehat{\mathcal{P}}_{L-1}\Big] &= \mathbb{E}\big[\mathcal{P}\big] + \tilde{c}_1 h_{L-1}^\alpha + \mathcal{O}\Big(h_L^{\alpha'}\Big).\end{aligned}$$

A simple linear combination of the two gives a higher-order approximation to $\mathbb{E}[\mathcal{P}]$; in particular, for SE, we have $\alpha = 1$ and $\alpha' = 2$ and

$$2\mathbb{E}\Big[\widehat{\mathcal{P}}_L\Big] - \mathbb{E}\Big[\widehat{\mathcal{P}}_{L-1}\Big] = \mathbb{E}\big[\mathcal{P}\big] + \mathcal{O}\big(h_\ell^2\big).$$

An approximation to the left-hand side is given by $\widehat{\mathcal{P}}^{(\text{MLMC})} + \widehat{Y}_{L,M_L}$. For SV, we have $\alpha = 2$ and $\alpha' = 4$, and

$$\frac{1}{3}\Big(4\mathbb{E}\Big[\widehat{\mathcal{P}}_L\Big] - \mathbb{E}\Big[\widehat{\mathcal{P}}_{L-1}\Big]\Big) = \mathbb{E}\big[\mathcal{P}\big] + \mathcal{O}\big(h_\ell^4\big).$$

An approximation to the left-hand side is given by $\widehat{\mathcal{P}}^{(\text{MLMC})} + \frac{1}{3}\widehat{Y}_{L,M_L}$. To observe the improved accuracy, the statistical error must also be reduced to match the bias error. An increase in accuracy from second- to fourth-order accuracy is achieved because the integrator is symmetric.

In the experiments, we apply extrapolation with exact sampling on the coarsest level in the following scenarios:

SE+ This is the same as SE, except the coarsest MLMC level is computed exactly. Further, extrapolation is used to increase the weak order of convergence from one to two. We choose $\ell_0 = 3$ and $N_{\text{coarse}} = 1$.

SV+ This is the same as SV, except the coarsest MLMC level is computed exactly and extrapolation is used to increase the weak order of convergence from two to four. Due to the fourth-order convergence, it is sufficient to take large time steps and we choose $\ell_0 = 1$, $L = 2$, and vary N_{coarse} rather than L.

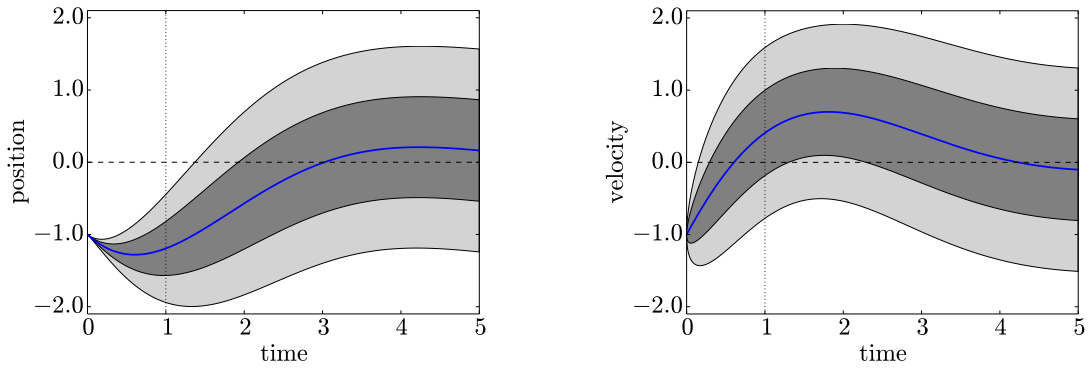

Figure 1: The position (left) and velocity (right) of a randomly forced harmonic oscillator with damping for $\omega_0 = 1 = \lambda = \sigma$. The mean value is shown together with one (dark gray) and two standard deviations (light gray). The numerical experiments consider the solution at time $t = 1$.

5.1 Langevin equation for the damped harmonic oscillator

We first consider Eq. (1.5) with $d = 1$ and

$$V(Q) = \frac{1}{2}\omega_0^2 Q^2. \tag{5.1}$$

Physically, with this potential, Eq. (1.5) describes a randomly forced harmonic oscillator with resonance frequency ω_0 and damping parameter λ; the strength of the Gaussian forcing is given by σ. For $\omega_0 = 0$ (i.e., in the absence of a potential), the SDE can be interpreted as a model for the dispersion of an atmospheric pollutant in a one-dimensional turbulent velocity field (see Rodean (1996)). In this case, $\sigma^2/(2\lambda)$ is the turbulent-velocity variance and $1/\lambda$ the velocity relaxation-time. In Figure 1, the marginal distributions for the position and velocity are visualised as a function of t for the first set of parameters used in the numerical experiments ($\omega_0 = 1 = \lambda = \sigma$).

We choose this simple example, for which we know the analytical solution, to verify the correctness of our code and to quantify numerical errors. As the system is linear, the joint pdf of Q and P is Gaussian and is defined by their mean and covariance. Denoting $\boldsymbol{X}(t) = (Q(t), P(t))^\mathsf{T}$ and the initial solution by $\boldsymbol{X}_0 = \boldsymbol{X}(t = 0) = (Q(t = 0), P(t = 0))^\mathsf{T}$, we have

$$\boldsymbol{X}(t) = \exp\big[-\Lambda t\big]\boldsymbol{X}_0 + \int_0^t \exp\big[-\Lambda(t-s)\big]\boldsymbol{\Sigma}\, dW(s) \tag{5.2}$$

with

$$\Lambda \coloneqq \begin{pmatrix} 0 & -1 \\ \omega_0^2 & \lambda \end{pmatrix}, \qquad \boldsymbol{\Sigma} \coloneqq \begin{pmatrix} 0 \\ \sigma \end{pmatrix}.$$

$\boldsymbol{X}(t)$ follows a Gaussian distribution with mean

$$\mathbb{E}\big[\boldsymbol{X}(t)\big] = \exp\big[-\Lambda t\big]\boldsymbol{X}_0 \tag{5.3}$$

and covariance matrix

$$B(t) \coloneqq \int_0^t \exp\big[-\Lambda(t-s)\big]\boldsymbol{\Sigma}\boldsymbol{\Sigma}^\mathsf{T} \exp\big[-\Lambda^\mathsf{T}(t-s)\big]\, ds. \tag{5.4}$$

The mean vector and covariance matrix can be evaluated easily using a computer algebra system.

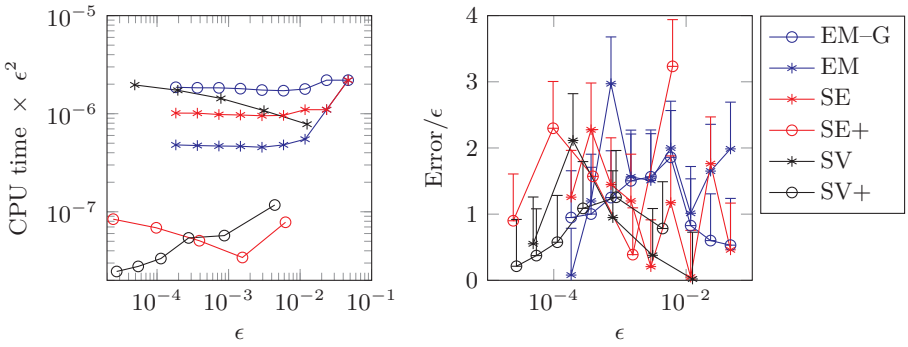

Figure 2: For the harmonic oscillator with parameter set 1. The left-hand plot shows the CPU time for a given value of ϵ; the time is scaled by ϵ^{-2} and this leads to a flat profile in each case. The right-hand plot shows the mean and standard deviation of the error; the errors are divided by ϵ to show both mean and standard deviation are $\mathcal{O}(\epsilon)$.

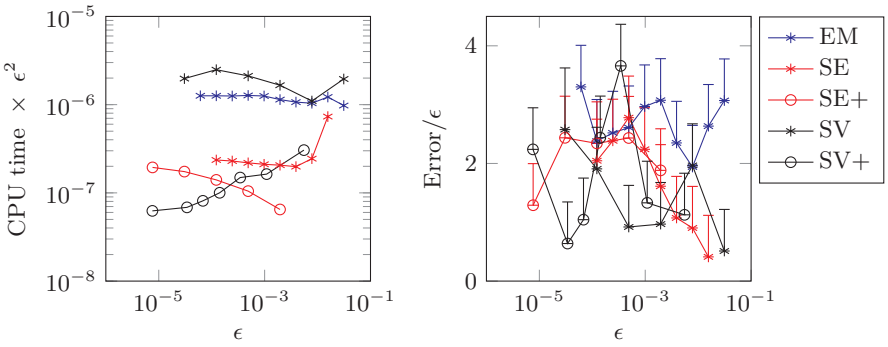

Figure 3: For the harmonic oscillator with parameter set 2.

Numerical results

We compute $\mathbb{E}[\phi(\boldsymbol{X}(1))]$ for $\phi(Q,P) = Q^2 + P^2$ and the following set of parameters:

1. $\omega_0 = 1 = \lambda = \sigma$; and
2. $\omega_0 = 1$, $\lambda = 4$, and $\sigma = 2$.

The initial position and velocity were set to $Q(t=0) = P(t=0) = -1$ in both cases. Errors are computed using the exact value computed from Eqs. (5.3) and (5.4). The exact values are 2.079454259269310 and 1.750429495028081, respectively. The CPU time scaled by ϵ^{-2} and error scaled by ϵ are plotted in Figures 2 and 3 against ϵ. The scaling means we expect the graphs to be flat. We observe for both parameter sets that SV+ is the fastest integrator for small ϵ, though the difference with SE+ is small. Even though SV+ uses a weak fourth-order accurate integrator, the complexity of MLMC cannot be reduced beyond $\mathcal{O}(\epsilon^{-2})$ and is the same as for the other integrators. The improvements come by improving constants, in this case by a large amount in comparison to EM–G. The differences between EM and EM–G results from the cost of generating samples from the three-point and Gaussian distribution respectively and, for this example, the random-number generation represents a significant part of the computation. For the second set of parameter values in Figure 3, the relaxation time is shorter and the noise is larger, and EM is unable to perform as well as for the first set of parameters. This is to be expected and the stability constraints on EM are well known.

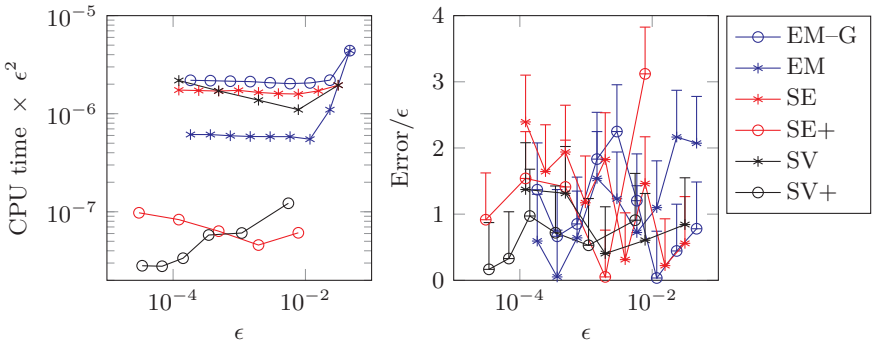

Figure 4: For the double-well potential with parameter set 1.

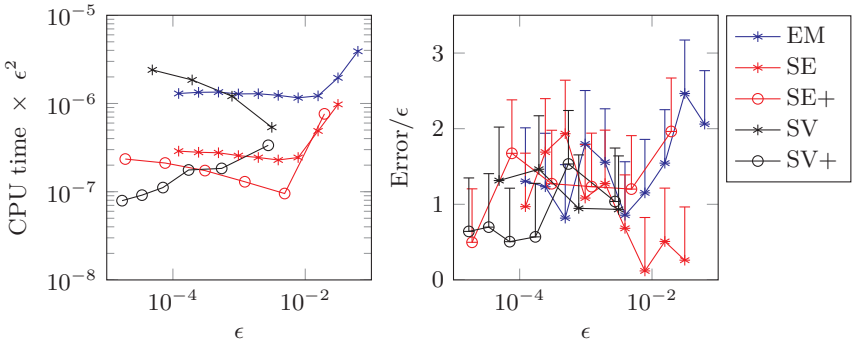

Figure 5: For the double-well potential with parameter set 2.

5.2 Double-well potential

We now change the potential and consider the double-well potential

$$V(Q) = \frac{\omega_0^2}{8Q_{\min}^2}(Q^2 - Q_{\min}^2)^2,$$

where $Q_{\min}$ and ω_0 are parameters. We compute $\mathbb{E}[\phi(\boldsymbol{X}(1))]$ for $\phi(Q,P) = Q^2 + P^2$ and the following set of parameters:

1. $Q_{\min} = \omega_0 = 1 = \lambda = \sigma$; and
2. $Q_{\min} = \omega_0 = 1$, $\lambda = 4$, and $\sigma = 2$.

We take initial data $Q(t=0) = P(t=0) = -1$. Errors are computed relative to a numerically computed value with $\epsilon = 10^{-6}$ given by 2.544985 and 2.02376954. Again, the scaled CPU time and error are plotted against ϵ in Figures 4 and 5. A similar pattern to the harmonic oscillator holds, and SE+ and SV+ are found to be the most efficient integrators, depending on the size of ϵ.

6 Conclusion

Tables 1 and 2 summarise our findings: MLMC gives a significant speed-up over the traditional Monte Carlo computation of averages and, even though the optimal complexity estimate $\mathcal{O}(\epsilon^{-2})$ of MLMC holds for all the integrators under study, there is significant variation between the integrators. Splitting methods are particularly appropriate for the Langevin equations and using the exact OU solution yields a far more stable integrator than Euler–Maruyama, even though

Parameter Set 1	Harmonic oscillator $\epsilon = 1.2 \times 10^{-3}$		Double-well potential $\epsilon = 1.9 \times 10^{-3}$	
	seconds	ratio	seconds	ratio
EM–G–MC	13.7	10× slower	5.93	10× slower
EM–G	1.3	1	5.6×10^{-1}	1
EM	3.2×10^{-1}	4× faster	1.6×10^{-1}	3.5× faster
SE+	1.8×10^{-2}	18× faster	1.3×10^{-2}	43× faster

Table 1: Example CPU times for the EM, EM–G, and SE+ variants of the MLMC method, alongside data (EM–G–MC) for the Monte Carlo method based on Euler–Maruyama with Gaussian increments. The speed-up ratios are given relative to EM–G.

Parameter Set 2	Harmonic oscillator $\epsilon = 1.1 \times 10^{-3}$		Double-well potential $\epsilon = 2.8 \times 10^{-3}$	
	seconds	ratio	seconds	ratio
EM–MC	19.8	6× slower	3.31	6.7× slower
EM–G	3.16	1	4.93×10^{-1}	1
EM	8.34×10^{-1}	3.7× faster	1.4×10^{-1}	3.5× faster
SV+	1.6×10^{-2}	200× faster	4.3×10^{-2}	11× faster

Table 2: Example CPU times for the EM, EM–G, and SV+ variants of the MLMC method, alongside data (EM–G–MC) for the Monte Carlo method based on Euler–Maruyama with Gaussian increments. The speed-up ratios are given relative to EM–G.

both are explicit integrators. In the experiments, the difference in computation time between Euler–Maruyama and the splitting methods is greater for the second parameter sets (compare EM and SE in Figures 2–5), because the dissipation λ is higher and the Euler–Maruyama method suffers from a time-step restriction.

The use of three-point random variables provides a speed-up because Gaussian random variables are more expensive to sample, and also allows exact computation of the coarse level. This method is easy to implement and it works well because the coarse level is the most significant part of the work for these problems. The improvement would be less dramatic in higher dimensions as the number of samples required would increase dramatically and it may be impossible to compute the coarse level exactly.

Overall, the fastest method depends on the size of ϵ. For $\epsilon \approx 10^{-4}$, the fourth-order method given by the extrapolated Störmer–Verlet/exact OU process with exact coarse-level averages is fastest. At intermediate ranges of $\epsilon \approx 10^{-3}$ to 10^{-2}, symplectic Euler is best used in place Störmer–Verlet. (Debrabant & RÃűßler, 2013) demonstrates theoretically that indeed higher-order integrators can reduce the constant (though not the complexity) when coupled with a low-order method for the coarse levels. The extrapolation method is a simple and natural candidate for this theory, as it reduces the order only on the finest level and all the coarse levels are computed with lower order.

This paper also introduced an alternative analysis method for MLMC based on modified equations. It provides a convenient approach to MLMC through weak-approximation theory; strong-approximation theory is only needed to relate the original and modified equations and not the numerical methods. This accommodated the use of the three-point approximation and the exact OU solution easily.

A Proof of Lemma 3.3

Proof. This is a standard Gronwall argument with the Ito isometry. The integral equation for the difference $\boldsymbol{X}(t) - \boldsymbol{X}_h(t)$ is

$$\begin{aligned}\boldsymbol{X}(t) - \boldsymbol{X}_h(t) = &\int_0^t \big(\boldsymbol{f}(\boldsymbol{X}(s)) - \tilde{\boldsymbol{f}}(\boldsymbol{X}_h(s))\big)\, ds \\ &+ \int_0^t \Big(G(\boldsymbol{X}(s)) - \widetilde{G}(\boldsymbol{X}_h(s))\Big)\, d\boldsymbol{W}(s),\end{aligned}$$

where $\tilde{\boldsymbol{f}} \coloneqq \boldsymbol{f} + h\boldsymbol{f}_1$ and $\widetilde{G} \coloneqq G + hG_1$. By conditions (i) and (ii),

$$\big\|\boldsymbol{f}(\boldsymbol{X}(s)) - \boldsymbol{f}(\boldsymbol{X}_h(s)) - h\boldsymbol{f}_1(\boldsymbol{X}_h(s))\big\|_{L^2(\Omega,\mathbb{R}^d)} \leq L\big\|\boldsymbol{X}(s) - \boldsymbol{X}_h(s)\big\|_{L^2(\Omega,\mathbb{R}^d)} + C_1 h$$

and similarly for G. Assume $\boldsymbol{X}(0) = \boldsymbol{X}_h(0)$. The Ito isometry and Jensen's inequality give

$$\begin{aligned}\mathbb{E}\Big[\big\|\boldsymbol{X}(t) - \boldsymbol{X}_h(t)\big\|^2\Big] \leq &2t\int_0^t \mathbb{E}\Big[\big\|\boldsymbol{f}(\boldsymbol{X}(s)) - \tilde{\boldsymbol{f}}(\boldsymbol{X}_h(s))\big\|^2\Big]\, ds \\ &+ 2\int_0^t \Big\|G(\boldsymbol{X}(s)) - \widetilde{G}(\boldsymbol{X}_h(s))\Big\|_{\mathrm{F}}^2\, ds \\ \leq &4t\int_0^t L^2\mathbb{E}\Big[\big\|\boldsymbol{X}(s) - \boldsymbol{X}_h(s)\big\|^2\Big] + C_1^2 h^2\, ds \\ &+ 4\int_0^t L^2\mathbb{E}\Big[\big\|\boldsymbol{X}(s) - \boldsymbol{X}_h(s)\big\|^2\Big] + C_1^2 h^2\, ds.\end{aligned}$$

Gronwall's inequality gives $\mathcal{O}(h^2)$ bounds on $\mathbb{E}\big[\|\boldsymbol{X}(t) - \boldsymbol{X}_h(t)\|^2\big]$. A similar argument can be applied to $\boldsymbol{Y}$ and applying the Lipschitz condition on ψ completes the proof. □

References

Abdulle, A., & Blumenthal, A. 2013. Stabilized multilevel Monte Carlo method for stiff stochastic differential equations. *Journal of Computational Physics*, **251**, 445–460.

Anderson, D. F., & Higham, D. J. 2011. Multilevel Monte Carlo for continuous time Markov chains, with applications in biochemical kinetics. *Multiscale Modeling & Simulation*, **10**(1), 146–179.

Barth, A., Schwab, C., & Zollinger, N. 2011. Multi-level Monte Carlo finite element method for elliptic PDEs with stochastic coefficients. *Numerische Mathematik*, **119**(1), 123–161.

Beard, D. A., & Schlick, T. 2000. Inertial stochastic dynamics: I. Long timestep methods for Langevin dynamics. *Journal of Chemical Physics*, **112**(17), 7313–7322.

Bou-Rabee, N., & Owhadi, H. 2010. Long-run accuracy of variational integrators in the stochastic context. *SIAM J. Numer. Anal.*, **48**(1), 278–297.

Brunge, A., Brooks, C. L., & Karplus, M. 1984. Stochastic boundary conditions for molecular dynamics simulations of ST2 water. *Chem. Phys. Lett.*, **105**, 495–400.

Cliffe, K. A., Giles, M. B., Scheichl, R., & Teckentrup, A. L. 2011. Multilevel Monte Carlo methods and applications to elliptic PDEs with random coefficients. *Computing and Visualization in Science*, **14**(1), 3–15.

Debrabant, K., & RÃűßler, A. 2013. *On the Acceleration of the Multi-Level Monte Carlo Method.*

Debussche, A., & Faou, E. 2012. Weak backward error analysis for SDEs. *SIAM J. Numer. Anal.*, **50**(3), 1735–1752.

Dereich, S., & Heidenreich, F. 2011. A multilevel Monte Carlo algorithm for Lévy-driven stochastic differential equations. *Stochastic Processes and their Applications*, **121**, 1565–1587.

Giles, M. B. 2008. Multilevel Monte Carlo path simulation. *Oper. Res.*, **56**(3), 607–617.

Giles, M.B., & Reisinger, C. 2012. Stochastic finite differences and multilevel Monte Carlo for a class of SPDEs in finance. *SIAM Journal of Financial Mathematics*, **3**(1), 572–592.

Giles, M.B., & Szpruch, L. 2013. Multilevel Monte Carlo methods for applications in finance. *Pages 3–47 of:* Gerstner, T., & Kloeden, P. (eds), *Recent Developments in Computational Finance*. World Scientific Press.

Hairer, E., Lubich, C., & Wanner, G. 2010. *Geometric Numerical Integration*. Springer Series in Computational Mathematics, vol. 31. Springer, Heidelberg. Structure-preserving Algorithms for Ordinary Differential Equations, Reprint of the second (2006) edition.

Heinrich, S. 2001. Multilevel Monte Carlo Methods. *Pages 58–67 of: Large-scale scientific computing*. Springer.

Hoel, H., von Schwerin, E., Szepessy, A., & Tempone, R. 2012. Adaptive multilevel Monte Carlo simulation. *Numerical Analysis of Multiscale Computations*, **82**, 217–234.

Hutzenthaler, M., Jentzen, A., & Kloeden, P. E. 2013. Divergence of the multilevel Monte Carlo Euler method for nonlinear stochastic differential equations. *Ann. Appl. Probab.*, **23**(5), 1913–1966.

Kloeden, P. E., & Platen, E. 1992. *Numerical solution of stochastic differential equations*. Applications of Mathematics (New York), vol. 23. Springer-Verlag, Berlin.

Kopec, M. 2013. *Weak backward error analysis for Langevin process*. arXiv:1310.2599.

Leimkuhler, B., & Reich, S. 2004. *Simulating Hamiltonian Dynamics*. Cambridge Monographs on Applied and Computational Mathematics, vol. 14. Cambridge University Press.

Lemaire, V., & Pagès, G. 2013. *Multilevel Richardson–Romberg extrapolation*. Preprint arXiv:1401.1177.

Mishra, S., Schwab, C., & Šukys, J. 2012. Multi-level Monte Carlo finite volume methods for nonlinear systems of conservation laws in multi-dimensions. *Journal of Computational Physics*, **231**(8), 3365–3388.

Rodean, H. C. 1996. Stochastic Lagrangian models of turbulent diffusion. *Meteorological Monographs*, **26**(48), 1–84.

Shardlow, T. 2006. Modified equations for stochastic differential equations. *BIT*, **46**(1), 111–125.

Talay, D., & Tubaro, L. 1990. Expansion of the global error for numerical schemes solving stochastic differential equations. *Stochastic Anal. Appl.*, **8**(4), 483–509.

Wang, W., & Skeel, R. D. 2003. Analysis of a few numerical integration methods for the Langevin equation. *Molecular Physics*, **101**(14), 2149–2156.

Zygalakis, K. C. 2011. On the existence and the applications of modified equations for stochastic differential equations. *SIAM J. Sci. Comput.*, **33**(1), 102–130.