DE GRUYTER DOI 10.1515/mcma-2013-0014 — Monte Carlo Methods Appl. 2014; 20 (1):1–41

Research Article

Håkon Hoel, Erik von Schwerin, Anders Szepessy and Raúl Tempone

Implementation and analysis of an adaptive multilevel Monte Carlo algorithm

Abstract: We present an adaptive multilevel Monte Carlo (MLMC) method for weak approximations of solutions to Itô stochastic differential equations (SDE). The work [11] proposed and analyzed an MLMC method based on a hierarchy of uniform time discretizations and control variates to reduce the computational effort required by a single level Euler–Maruyama Monte Carlo method from $\mathcal{O}(\mathrm{TOL}^{-3})$ to $\mathcal{O}(\mathrm{TOL}^{-2}\log(\mathrm{TOL}^{-1})^2)$ for a mean square error of $\mathcal{O}(\mathrm{TOL}^2)$. Later, the work [17] presented an MLMC method using a hierarchy of adaptively refined, non-uniform time discretizations, and, as such, it may be considered a generalization of the uniform time discretization MLMC method. This work improves the adaptive MLMC algorithms presented in [17] and it also provides mathematical analysis of the improved algorithms. In particular, we show that under some assumptions our adaptive MLMC algorithms are asymptotically accurate and essentially have the correct complexity but with improved control of the complexity constant factor in the asymptotic analysis. Numerical tests include one case with singular drift and one with stopped diffusion, where the complexity of a uniform single level method is $\mathcal{O}(\mathrm{TOL}^{-4})$. For both these cases the results confirm the theory, exhibiting savings in the computational cost for achieving the accuracy $\mathcal{O}(\mathrm{TOL})$ from $\mathcal{O}(\mathrm{TOL}^{-3})$ for the adaptive single level algorithm to essentially $\mathcal{O}(\mathrm{TOL}^{-2}\log(\mathrm{TOL}^{-1})^2)$ for the adaptive MLMC algorithm.

Keywords: Computational finance, Monte Carlo, multilevel, adaptivity, weak approximation, error control, Euler–Maruyama method, a posteriori error estimates, backward dual functions, adjoints

MSC 2010: 65C30, 65Y20, 65L50, 65H35, 60H35, 60H10

Håkon Hoel, Raúl Tempone: Applied Mathematics and Computational Sciences, KAUST, Thuwal, Saudi Arabia, e-mail: hakon.hoel@kaust.edu.sa, raul.tempone@kaust.edu.sa
Erik von Schwerin: CSQI-MATHICSE, École Polytechnique Fédérale de Lausanne, Switzerland, e-mail: erik.vonschwerin@epfl.ch
Anders Szepessy: Department of Mathematics, Royal Institute of Technology (KTH), Stockholm, Sweden, e-mail: szepessy@kth.se

1 Introduction

This work develops multilevel adaptive algorithms for weak approximation of Itô stochastic differential equations (SDEs)

$$dX(t) = a(t, X(t))dt + b(t, X(t))dW(t), \quad 0 < t < T, \tag{1.1}$$

where $X(t;\omega)$ is a stochastic process in $\mathbb{R}^d$, with randomness generated by a k-dimensional Wiener process with independent components $W(t;\omega)$, cf. [20, 28], and $a(t,x) \in \mathbb{R}^d$ and $b(t,x) \in \mathbb{R}^{d\times k}$ are the drift and diffusion fluxes. For any given sufficiently well behaved function $g : \mathbb{R}^d \to \mathbb{R}$ our goal is to approximate the expected value $\mathrm{E}[g(X(T))]$ by adaptive multilevel Monte Carlo (MLMC) methods. A typical application is to compute option prices in mathematical finance, cf. [13, 19], and other related models based on stochastic dynamics are used for example in molecular dynamics simulations at constant temperature [5], for stochastic climate prediction [23], and for wave propagation in random media [1].

The computational complexity of a Monte Carlo method is determined by the number of sample realizations approximating $g(X(T))$ and their average cost. When a standard Monte Carlo method based on a uniform time stepping scheme of weak order one is used to compute $\mathrm{E}[g(X(T))]$ to an accuracy TOL with high probability, the cost is asymptotically proportional to TOL^{-3}, provided that the functions a, b, and g are sufficiently regular. A Monte Carlo method cannot do better than a cost proportional to TOL^{-2}, since this is the total cost when each realization of $g(X(T))$ is generated exactly at a unit cost. The goal of this work is to combine two techniques for improving the standard Monte Carlo method: the first is to use adaptive time stepping which retains the single level complexity $\mathcal{O}(\mathrm{TOL}^{-3})$ for a wider set of problems than a uniform time stepping does, and can reduce the proportionality constant for other problems with widely varying scales. The second is the MLMC method, which in many cases can reduce the complexity to nearly the optimal $\mathcal{O}(\mathrm{TOL}^{-2})$ when based on the Euler–Maruyama scheme, and which can achieve the optimal rate using the Milstein scheme.

In the context of weak approximation of SDEs, the MLMC method based on uniform time stepping was introduced by Giles in [11], and around ten years prior to Giles' method, a similar MLMC idea was presented for applications in the context of parametric integration, cf. [15, 16]. Giles' MLMC method, which is an extension of a two-level control variate technique, cf. [21], reduces the complexity of weak approximations of SDEs by a control variate type variance reduction. The variance reduction is obtained using subtly correlated numerical realizations of the SDE (1.1) on hierarchies of uniform time meshes of size

$$\Delta t_\ell = C^{-\ell}\Delta t_0, \quad C \in \{2, 3, \dots\} \text{ and } \ell \in \{0, 1, \dots, L\}. \tag{1.2}$$

That is, the MLMC method approximates $\mathrm{E}[g(X(T))]$ by the multilevel estimator

$$\mathcal{A}_{\mathcal{ML}}(g(\overline{X}(T)); M_0) = \sum_{i=1}^{M_0} \frac{g(\overline{X}_0(T;\omega_{i,0}))}{M_0} + \sum_{\ell=1}^{L}\sum_{i=1}^{M_\ell} \frac{g(\overline{X}_\ell(T;\omega_{i,\ell})) - g(\overline{X}_{\ell-1}(T;\omega_{i,\ell}))}{M_\ell}, \tag{1.3}$$

with $\overline{X}_\ell(T;\omega)$ denoting a numerical solution realization generated on a mesh with uniform step size Δt_ℓ. The multilevel estimator is a sum of $L+1$ sample averages computed from mutually independent sample sets on the given mesh levels with M_ℓ respective, independent realizations. Furthermore, the number of realizations on the higher leveles, $\{M_\ell\}_{\ell=1}^{L}$, have a fixed relation to the number of realizations on the coarsest mesh, M_0, which is the only free parameter in (1.3), when the number of levels L is fixed. To reduce the variance in the estimator (1.3), the realization pairs $\overline{X}_\ell(T;\omega_{i,\ell})$ and $\overline{X}_{\ell-1}(T;\omega_{i,\ell})$ of the summands $g(\overline{X}_\ell(T;\omega_{i,\ell})) - g(\overline{X}_{\ell-1}(T;\omega_{i,\ell}))$ for each level $\ell > 0$ are generated from the same Brownian path, $W(t;\omega_{i,\ell})$, but realized on *different* temporal grids with uniform time steps, Δt_ℓ and $\Delta t_{\ell-1}$, respectively. The efficiency of the multilevel estimator stems from an a priori known order of strong convergence for the numerical method employed on each level of the hierarchy.

Supposing $\mathrm{TOL} > 0$ is the desired accuracy in the approximation of $\mathrm{E}[g(X(T))]$, the main result of Giles' work [11] is that the computational cost needed to achieve the Mean Square Error (MSE)

$$\mathrm{E}[(\mathcal{A}_{\mathcal{ML}}(g(\overline{X}(T)); M_0) - \mathrm{E}[g(X(T))])^2] = \mathcal{O}(\mathrm{TOL}^2), \tag{1.4}$$

when generating numerical realizations $\overline{X}_\ell(T;\omega)$ using the first order accurate Forward Euler method, can be reduced from $\mathcal{O}(\mathrm{TOL}^{-3})$ with the standard Monte Carlo method to $\mathcal{O}((\mathrm{TOL}^{-1}\log(\mathrm{TOL}^{-1}))^2)$ using the MLMC method. Furthermore, whenever the function g is Lipschitz and for scalar Itô SDE, the computational cost can be further reduced to $\mathcal{O}(\mathrm{TOL}^{-2})$ using the first order strong convergence Milstein method. In addition, the work [10] shows how to apply the Milstein method for several scalar SDE cases where the Lipschitz condition is not fulfilled and still obtain the cost $\mathcal{O}(\mathrm{TOL}^{-2})$.

Building on the work on adaptive methods for weak approximation of SDE presented in [25, 29] and Giles' work on uniform time stepping MLMC methods [11], the contribution of the present paper is the development and analysis of two novel MLMC algorithms with adaptive, non-uniform time stepping: one algorithm that uses adaptive mesh refinements to construct a path dependent mesh for each realization and another algorithm that constructs the meshes adaptively based on sample averaged error densities and then uses the same mesh for all realizations on a given mesh level in the hierarchy. The former algorithm is referred to as the *stochastic* time stepping algorithm and the latter as the *deterministic* time stepping algorithm. Adaptivity is useful for problems lacking regularity since adaptive mesh refinement algorithms resolve singular points better than uniform mesh algorithms by construction, and may consequently also have considerably lower computational complexity, cf. [26]. The idea of extending the MLMC method [11] to hierarchies of adaptively refined, non-uniform time discretizations that are generated by the adaptive algorithm introduced in [8, 25, 26] was first introduced and tested computationally by the authors in [17].

The numerical method for SDE considered in this paper is the Euler–Maruyama method with non-uniform time stepping which we now recall for the reader's convenience. Let $0 = t_0 < t_1 < \cdots < t_N = T$ denote a given time discretization, without reference to its place in the hierarchies, and $\{0 = W(t_0;\omega), W(t_1;\omega), \dots, W(t_N;\omega)\}$ denote a realization of the Wiener process on that discretization. Then the Euler–Maruyama approximation to the true solution of (1.1) is given by the scheme

$$\begin{aligned}\overline{X}(t_0;\omega) &= X(0),\\ \overline{X}(t_{n+1};\omega) &= a(\overline{X}(t_n;\omega),t_n)(t_{n+1}-t_n) + b(\overline{X}(t_n;\omega),t_n)(W(t_{n+1};\omega) - W(t_n;\omega)),\end{aligned} \tag{1.5}$$

iterated for $n = 1, 2, \dots$. In the setting of adaptive mesh refinement there is no given notion of mesh size, so the hierarchy of meshes for the multilevel estimator (1.3) cannot be described as for the uniform time stepping (1.2). Instead, we generate a hierarchy of meshes by successively increasing the accuracy in our computations, introducing the time discretization error tolerance levels[1]

$$\mathrm{TOL}_{\mathrm{T},\ell} = 2^{\ell-L}\mathrm{TOL}_{\mathrm{T}}, \quad \ell \in \{0,1,\dots,L\}, \tag{1.6}$$

and (by adaptive refinements based on error indicators) determining the corresponding meshes so that for each level $\ell \in \{0,1,\dots,L\}$,

$$|\mathrm{E}[g(X(T))] - \mathrm{E}[g(\overline{X}_\ell(T))]| \lesssim \mathrm{TOL}_{\mathrm{T},\ell}.$$

In Section 4, we prove that this procedure results in an adaptive MLMC algorithm fulfilling

$$|\mathcal{A}_{\mathcal{ML}}(g(\overline{X}(T)); M_0) - \mathrm{E}[g(X(T))]| \le \mathrm{TOL}, \tag{1.7}$$

with probability close to one, and that the computational cost for obtaining this error estimate (1.4) is essentially $\mathcal{O}(\mathrm{TOL}^{-2}\log(\mathrm{TOL}^{-1})^2)$, cf. Theorem 4.1 and 4.2, respectively. Analogous theoretical results also hold for the adaptive algorithm with deterministic stepping, but, for the sake of brevity, they are not included here, see [25] for more information on this setting.

This work also includes three numerical examples, the most relevant ones being one with a drift singularity and one with a stopped diffusion. For both of these examples the observed computational work of multilevel Monte Carlo based on adaptive time stepping is approximately $\mathcal{O}(\mathrm{TOL}^{-2}\log(\mathrm{TOL}^{-1})^2)$, that is close to the optimal complexity and more efficient than the single level version of the adaptive algorithm. The rest of this paper is organized as follows. Section 1.1 introduces the notion of error density and error indicators, and recalls useful results for single level adaptive forward Euler Monte Carlo methods. Section 2 describes the adaptive multilevel Monte Carlo algorithms. Section 3 presents numerical examples and Section 4 proves accuracy and complexity results for the adaptive MLMC algorithm.

1.1 A single level posteriori error expansion

In this subsection we give a short description the adaptive numerical method we will use for SDE, recalling theoretical results and stating required regularity condition for the method.

Assume that the process X satisfies (1.1) and its corresponding numerical solution $\overline{X}$ is given by (1.5), then the error expansions in [29, Theorem 1.2 and 2.2] have the form

$$\mathrm{E}[g(X(T)) - g(\overline{X}(T))] = \mathrm{E}\left[\sum_{n=0}^{N-1} \rho_n \Delta t_n^2\right] + \text{higher order terms}, \tag{1.8}$$

where $\rho_n \Delta t_n^2$ are computable error indicators and ρ_n measures the density of the global error in (1.8). Typically, an adaptive algorithm does the two following things iteratively:

(1) if the error indicators satisfy an accuracy condition, then stop; otherwise
(2) the algorithm chooses where to refine the mesh based on the error indicators and return to step (1).

[1] For error control, the tolerance is split into a statistical error tolerance and a time discretization error tolerance, that is, $\mathrm{TOL} = \mathrm{TOL}_\mathrm{S} + \mathrm{TOL}_\mathrm{T}$, cf. Section 2.

In addition to estimating the global error $\mathrm{E}[g(X(T)) - g(\overline{X}(T))]$ in the sense of equation (1.8), the error indicators $\rho_n \Delta t_n^2$ indicate which mesh intervals that should be refined to reach the optimal mesh; a result that follows from the almost sure convergence of the density ρ_n as $\mathrm{TOL_T} \downarrow 0$, cf. [26, Section 4].

Given an initial time discretization $\Delta t[0](t)$, the stochastic time stepping algorithm refines the initial mesh until[2]

$$|\rho(t,\omega)|(\Delta t(t))^2 < \text{constant}. \tag{1.9}$$

The final mesh refinment $\Delta t(t)$ is obtained by repeated halving of mesh intervals and thus takes the form

$$\Delta t(t) = \Delta t[0](t)/2^n \quad \text{for some natural number } n = n(t,\omega).$$

The criterion (1.9) uses an approximate error density function ρ, satisfying for $t \in [0,T]$ and all outcomes ω the uniform upper and lower bounds

$$\rho_{\mathrm{low}}(\mathrm{TOL_T}) \le |\rho(t,\omega)| \le \rho_{\mathrm{up}}(\mathrm{TOL_T}). \tag{1.10}$$

In this construction the positive functions ρ_{low} and ρ_{up} are chosen so that the limits

$$\rho_{\mathrm{up}}(\mathrm{TOL_T}) \to +\infty, \quad \rho_{\mathrm{low}}(\mathrm{TOL_T}) \to 0 \quad \text{and} \quad \frac{\mathrm{TOL_T}}{\rho_{\mathrm{low}}(\mathrm{TOL_T})} \to 0$$

hold as $\mathrm{TOL_T} \downarrow 0$.

For each realization, successive subdivisions of the time steps will asymptotically yield the smallest mesh, in terms of grid points, satisfying (1.9). Furthermore, the Wiener increments ΔW generated on the refined mesh by Brownian bridge interpolation, cf. [20], will have the correct distribution with the necessary independence. At this point we note that adaptive time stepping for SDE is a subtle construction that may lead to wrong results if implemented incorrectly, cf. [9].

Remark 1.1. Although the time and Wiener increments adaptively generated to satisfy (1.9)–(1.10) are not adapted to the natural Wiener filtration, it is verified in [29] that the adaptive method indeed converges to the correct limit, equaling the limit of the Euler–Maruyama method with adapted time steps.

Remark 1.2. The work [29] includes an additional assumption, namely that the sensitivity of the error density to values of the Wiener process can be bounded by a deterministic function of $\mathrm{TOL_T}$. This assumption can be removed by estimating the sensitivity of the error density to values of the Wiener process directly in terms of polynomials of the Wiener increments and then following essentially the same steps of the analysis given in [29, Section 3], taking into account that an accepted sequence of refinements remains the same under perturbations of the Wiener increments if all the signs of the refinement inequalities (1.9) remain unchanged for all time steps during the finite sequence of refinements. This line of analysis yields the same estimates for strong and weak convergence as stated in [29].

The regularity conditions presented in the following lemma is a subset of the conditions required in the work [27] for developing an adaptive weak approximation method in the more general setting of jump diffusions.

Lemma 1.3 (Regularity [27, Lemma 2.1]). (a) *Assume that the following regularity conditions hold:*

(1) *The functions $a(t,x)$ and $b(t,x)$ are continuous in (t,x) and are twice continuously differentiable with respect to x.*
(2) *The partial derivatives of first and second order with respect to x of the functions a and b are uniformly bounded.*
(3) *The function g is twice continuously differentiable, and together with its partial derivatives of first and second order it is uniformly bounded.*

Then the cost to go *function, defined by*

$$u(t,x) = \mathrm{E}[g(X(T)) \mid X(t) = x], \tag{1.11}$$

[2] The precise expressions including the constants are given in (2.7) and (2.20) below.

satisfies the Kolmogorov equation

$$\partial_t u(t,x) + a_k \partial_k u(t,x) + d_{kn}\partial_{kn} u(t,x) = 0, \quad u(T,\cdot) = g, \tag{1.12}$$

where we use the Einstein summation convention[3] *and* $d_{kn} := b_k^l b_n^l/2$.

(b) *Assume further that the following extra regularity conditions are satisfied:*
  (1) *The functions* $\partial_\beta a(t,\cdot)$ *and* $\partial_\beta b(t,\cdot)$ *are bounded uniformly in* t *for multi-indices* β *with* $1 \le |\beta| \le 8$.
  (2) *The functions* $a(\cdot,x)$, $b(\cdot,x)$ *have continuous and uniformly bounded first order time derivatives.*
  (3) *The function* g *has spatial derivatives* $\partial_\beta g$, *with polynomial growth for* $|\beta| \le 8$.
  Then the function u *has continuous partial derivatives with respect to* x *up to the order* 8, *satisfying the following polynomial growth condition: for all* $i \in \{0,1,2\}$ *and* $\alpha \in \mathbb{N}^d$ *with* $i + |\alpha| \le 8$ *there exist* $p_{\alpha,i} \in \mathbb{N}$ *and* $C_{\alpha,i} > 0$ *such that*

$$\max_{0\le t\le T} |\partial_t^i \partial_\alpha u(t,x)| \le C_{\alpha,i}(1+|x|^{p_{\alpha,i}}) \quad \text{for all } x \in \mathbb{R}^d.$$

The strong convergence result we present next was stated and proved in [29, Lemma 3.1]. The convergence result is helpful for proving the existence of a stochastic time error expansion and for bounding the statistical error of the weak approximation.

Lemma 1.4 (Strong convergence). *For* X, *the solution of* (1.1), *suppose that* a, b, *and* g *satisfy the assumptions in Lemma 1.3, that* $\overline{X}$ *is constructed by the forward Euler method based on the stochastic time stepping algorithm described in Section 2 with step size* Δt_n *satisfying* (1.9)–(1.10), *and that the corresponding* ΔW_n *are generated by Brownian bridges. Then*

$$\sup_{0\le t\le T} \mathrm{E}[|X(t)-\overline{X}(t)|^2] = \mathcal{O}(\Delta t_{\mathrm{sup}}) = \mathcal{O}\left(\frac{\mathrm{TOL_T}}{\rho_{\mathrm{low}}(\mathrm{TOL_T})}\right) \to 0 \tag{1.13}$$

as $\mathrm{TOL_T} \downarrow 0$, *where* $\Delta t_{\mathrm{sup}} \equiv \sup_{n,\omega} \Delta t_n(\omega)$.

A theorem proving the existence of an error expansion for the more general setting of jump diffusions was given in the work [27]. We recall that theorem here, in a form adapted to our setting.

Theorem 1.5 (Single level stochastic time stepping error expansion [27, Theorem 3.1]). *Given the assumptions in Lemma 1.4 and a deterministic initial value* $X(0)$, *the time discretization error in* (1.8) *may be expressed by an expansion based on the drift and diffusion fluxes and the discrete dual functions* φ, φ', *and* φ'' *given in* (1.16)–(1.21). *The expansion has the following computable leading order terms:*

$$|\mathrm{E}[g(X(T))] - \mathrm{E}[g(\overline{X}(T))]| = \mathrm{E}\left[\sum_{n=0}^{N-1} \tilde{\rho}(t_n,\omega)(\Delta t_n)^2\right] + \mathcal{O}\left(\left(\frac{\mathrm{TOL_T}}{\rho_{\mathrm{low}}(\mathrm{TOL_T})}\right)^{1/2}\left(\frac{\rho_{\mathrm{up}}(\mathrm{TOL_T})}{\rho_{\mathrm{low}}(\mathrm{TOL_T})}\right)^{\epsilon}\right)\mathrm{E}\left[\sum_{n=0}^{N-1}(\Delta t_n)^2\right] \tag{1.14}$$

for any $\epsilon > 0$ *and where*

$$\begin{aligned} \tilde{\rho}(t_n,\omega) \equiv \frac{1}{2}\Big((\partial_t a_k + \partial_j a_k a_j + \partial_{ij} a_k d_{ij})\varphi_k(t_{n+1}) \\ + (\partial_t d_{km} + \partial_j d_{km} a_j + \partial_{ij} d_{km} d_{ij} + 2\partial_j a_k d_{jm})\varphi'_{km}(t_{n+1}) \\ + (2\partial_j d_{km} d_{jr})\varphi''_{kmr}(t_{n+1})\Big) \end{aligned} \tag{1.15}$$

and the terms in the sum of (1.15) *are evaluated at the a posteriori known points* $(t_n, \overline{X}(t_n))$, *i.e.,*

$$\partial_\alpha a \equiv \partial_\alpha a(t_n, \overline{X}(t_n)), \quad \partial_\alpha b \equiv \partial_\alpha b(t_n, \overline{X}(t_n)), \quad \partial_\alpha d \equiv \partial_\alpha d(t_n, \overline{X}(t_n)).$$

Here $\varphi \in \mathbb{R}^d$ *is the solution of the discrete dual backward problem*

$$\begin{aligned} \varphi_i(t_n) &= \partial_i c_j(t_n, \overline{X}(t_n))\varphi_j(t_{n+1}), \quad t_n < T, \\ \varphi_i(T) &= \partial_i g(\overline{X}(T)), \end{aligned} \tag{1.16}$$

[3] When an index variable appears twice in a single term, this means that a summation over all possible values of the index takes place. For example, $a_k\partial_k u(t,x) = \sum_{k=1}^d a_k \partial_k u(t,x)$, where d is the space dimension of the SDE ($a, x \in \mathbb{R}^d$).

with

$$c_i(t_n, x) \equiv x_i + \Delta t_n a_i(t_n, x) + \Delta W_n^{\ell} b_i^{\ell}(t_n, x) \tag{1.17}$$

and its respective first and second variation

$$\varphi'_{ij} \equiv \partial_{x_j(t_n)} \varphi_i(t_n) \equiv \frac{\partial \varphi_i(t_n; \overline{X}(t_n) = x)}{\partial x_j}, \tag{1.18}$$

$$\varphi''_{ikm}(t_n) \equiv \partial_{x_m(t_n)} \varphi'_{ik}(t_n) \equiv \frac{\partial \varphi'_{ik}(t_n; \overline{X}(t_n) = x)}{\partial x_m}, \tag{1.19}$$

respectively satisfying

$$\begin{aligned}
&\varphi'_{ik}(t_n) = \partial_i c_j(t_n, \overline{X}(t_n)) \partial_k c_p(t_n, \overline{X}(t_n)) \varphi'_{jp}(t_{n+1}) + \partial_{ik} c_j(t_n, \overline{X}(t_n)) \varphi_j(t_{n+1}), \quad t_n < T, \\
&\varphi'_{ik}(T) = \partial_{ik} g(\overline{X}(T)),
\end{aligned} \tag{1.20}$$

and

$$\begin{aligned}
\varphi''_{ikm}(t_n) = {} & \partial_i c_j(t_n, \overline{X}(t_n)) \partial_k c_p(t_n, \overline{X}(t_n)) \partial_m c_r(t_n, \overline{X}(t_n)) \varphi''_{jpr}(t_{n+1}) \\
& + \partial_{im} c_j(t_n, \overline{X}(t_n)) \partial_k c_p(t_n, \overline{X}(t_n)) \varphi'_{jp}(t_{n+1}) + \partial_i c_j(t_n, \overline{X}(t_n)) \partial_{km} c_p(t_n, \overline{X}(t_n)) \varphi'_{jp}(t_{n+1}) \\
& + \partial_{ik} c_j(t_n, \overline{X}(t_n)) \partial_m c_p(t_n, \overline{X}(t_n)) \varphi'_{jp}(t_{n+1}) + \partial_{ikm} c_j(t_n, \overline{X}(t_n)) \varphi_j(t_{n+1}), \quad t_n < T, \\
\varphi''_{ikm}(T) = {} & \partial_{ikm} g(\overline{X}(T)).
\end{aligned} \tag{1.21}$$

Observe that the constant in $\mathcal{O}$ that appears in (1.14) may not be uniform with respect to the value ϵ. Thus, in practice one chooses $\epsilon = \epsilon(\mathrm{TOL})$ to minimise the contribution of the remainder term to the error expansion (1.14).

At the end of this section, we describe how the error density $\tilde{\rho}(t_n, \omega)$ in (1.15) is modified so that the bounds (1.10) hold and $\Delta t_{\mathrm{sup}} \to 0$ as $\mathrm{TOL_T} \downarrow 0$. The latter criterion is needed to ensure that the adaptive method converges strongly, cf. Lemma 1.4. For $t \in [t_n, t_{n+1})$ and $n = 1, \dots, N$, consider the piecewise constant function

$$\rho(t) \equiv \mathrm{sign}(\tilde{\rho}(t_n)) \min(\max(|\tilde{\rho}(t_n)|, \rho_{\mathrm{low}}(\mathrm{TOL_T})), \rho_{\mathrm{up}}(\mathrm{TOL_T})), \tag{1.22}$$

where

$$\begin{aligned}
\rho_{\mathrm{low}}(\mathrm{TOL_T}) &= \mathrm{TOL_T}^{\bar{\gamma}}, \quad 0 < \bar{\gamma} < \frac{\alpha}{\alpha + 2}, \ 0 < \alpha < \frac{1}{2}, \\
\rho_{\mathrm{up}}(\mathrm{TOL_T}) &= \mathrm{TOL_T}^{-r}, \quad r > 0,
\end{aligned} \tag{1.23}$$

and $\mathrm{sign}(x) := 1$ for $x \geq 0$ and -1 for $x < 0$. The error density ρ defined by (1.22) is used in mesh refinement, cf. (2.19) and (2.20) for the stochastic time stepping algorithm, and (2.6) and (2.7) for the deterministic (path independent) time stepping algorithm. From now on, with a slight abuse of notation, let $\rho(t_n) = \rho_n$ denote the modified density (1.22).

Following the error expansion in Theorem 1.5, the time discretization error is approximated by

$$|\mathcal{E}_T| = |\mathrm{E}[g(X(T)) - g(\overline{X}(T))]| \lesssim \mathrm{E}\Bigg[\sum_{n=0}^{N-1} r(n)\Bigg] \tag{1.24}$$

using the error indicator, $r(n)$, defined by

$$r(n) \equiv |\rho(t_n)| \Delta t_n^2 \tag{1.25}$$

with the modified error density defined by (1.22). According to [25, Corollary 4.3 and Theorem 4.5], the error density converges almost surely to a limit density we denote $\hat{\rho}$, i.e., $\rho \to \hat{\rho}$ as $\mathrm{TOL_T} \downarrow 0$.

Remark 1.6 (More general expected values). Suppose that $h : [0, T] \times \mathbb{R}^d \to \mathbb{R}$ is sufficiently smooth. Then the error estimates in Theorem 1.5 includes estimates of expected values of the form

$$\mathrm{E}\Bigg[\int_0^T h(t, X(t)) dt + g(X(T))\Bigg].$$

This follows from introducing the additional variable $X^{(d+1)}(t)$ and the equation $dX^{(d+1)}(t) = h(t, X(t))dt$ to the SDE (1.1) and eliminating the additional variables in X and φ, so that equation (1.16) is extended to

$$\begin{aligned}&\varphi_i(t_n) = \partial_i c_j(t_n, \overline{X}(t_n))\varphi_j(t_{n+1}) + \partial_i h(t_n, \overline{X}(t_n))\Delta t_n, \quad t_n < T,\\&\varphi_i(T) = \partial_i g(\overline{X}(T)),\end{aligned}$$

equation (1.20) is extended to

$$\begin{aligned}&\varphi'_{ik}(t_n) = \partial_i c_j(t_n, \overline{X}(t_n))\partial_k c_p(t_n, \overline{X}(t_n))\varphi'_{jp}(t_{n+1}) + \partial_{ik} c_j(t_n, \overline{X}(t_n))\varphi_j(t_{n+1}) + \partial_{ik} h(t_n, \overline{X}(t_n))\Delta t_n, \quad t_n < T,\\&\varphi'_{ik}(T) = \partial_{ik} g(\overline{X}(T)),\end{aligned}$$

and equation (1.21) is extended in a similar fashion.

2 Adaptive algorithms and multilevel variance reduction

In this section we describe two versions of the adaptive MLMC algorithm. In Section 2.1, we present the deterministic (path independent) time stepping adaptive MLMC algorithm. This algorithm is designed for SDEs with singularities which occur essentially at deterministic times. For this class of problems the same refined mesh may be used to efficiently improve the accuracy of all realizations at a given accuracy threshold. An example from this class of problems, which we present in more detail in Section 3.2, is the drift singularity

$$dX(t) = \begin{cases} X(t)dW(t), & t \in [0, \alpha],\\ \frac{X(t)}{2\sqrt{t-\alpha}}dt + X(t)dW(t), \quad \alpha \in (0, T), & t \in (\alpha, T].\end{cases}$$

The deterministic time stepping adaptive MLMC algorithm constructs a mesh hierarchy by adaptive refinements based on comparatively small sample sets and then performs a greater number of realizations on the constructed mesh hierarchy to control the statistical error.

The second algorithm, which we present in Section 2.2, is the stochastic (path dependent) time stepping adaptive MLMC algorithm. This algorithm is designed for SDE problems where the optimal mesh refinement depends strongly on the realization, or path, considered. The stopped diffusion SDE

$$dX(t) = 1_{X(t)<2}\Big(\frac{11}{36}X(t)dt + \frac{1}{6}X(t)dW(t)\Big) \quad \text{and} \quad X(0) = 1.6$$

is an example of such a problem where the mesh refinement of a numerical realization $\overline{X}(t; \omega)$ is most important when the realization is close to the stopping barrier $x = 2$. See Section 3.3 for more on this stopped diffusion problem. For the stochastic time stepping adaptive MLMC algorithm, meshes are adaptively refined for each individual realization of the underlying Wiener process.

2.1 Path independent time stepping

We recall that for a given SDE (1.1), function $g : \mathbb{R}^d \to \mathbb{R}$, and tolerance $\mathrm{TOL} > 0$, our goal is to construct an adaptive MLMC algorithm for which the event

$$|\mathcal{A}_{\mathcal{ML}}(g(\overline{X}(T)); M_0) - \mathrm{E}[g(X(T))]| \leq \mathrm{TOL}$$

holds with probability close to one for the multilevel estimator $\mathcal{A}_{\mathcal{ML}}(g(\overline{X}(T)); M_0)$ that is defined by (1.3). We approach this goal by splitting the above approximation error as follows:

$$|\mathcal{A}_{\mathcal{ML}}(g(\overline{X}(T)); M_0) - \mathrm{E}[g(X_T)]| \leq \underbrace{|\mathrm{E}[g(\overline{X}_L(T)) - g(X(T))]|}_{=: \mathcal{E}_T} + \underbrace{|\mathcal{A}_{\mathcal{ML}}(g(\overline{X}(T)); M_0) - \mathrm{E}[g(\overline{X}_L(T))]|}_{=: \mathcal{E}_S},$$

and controlling the total error by requiring that the time discretization error fulfills $\mathcal{E}_T \leq \mathrm{TOL_T}$, asymptotically, and that the statistical error fulfills $\mathcal{E}_S \leq \mathrm{TOL_S}$, with high probability. Here, the tolerance also has been split into a time discretization error tolerance and a statistical error tolerance,

$$\mathrm{TOL} = \mathrm{TOL}_\mathrm{T} + \mathrm{TOL}_\mathrm{S}.$$

The computations then naturally divides into two phases. The first phase, consisting of Algorithm 1 and Algorithm 2, constructs a hierarchy of meshes to control the time discretization error $\mathcal{E}_T$. The second phase, consisting of Algorithm 3, Algorithm 4 and Algorithm 5, computes a sufficiently large number of Euler–Maruyama realizations (1.5) on the constructed hierarchy of grids to ensure that $\mathcal{E}_S \le \mathrm{TOL}_\mathrm{S}$, with probability close to one.

2.1.1 Generating the mesh hierarchy

We start by generating a hierarchy of meshes $\{\Delta t^{\{\ell\}}\}_{\ell=0}^{L}$ for numerical approximation of the SDE (1.1), with the ℓth mesh given by

$$\Delta t^{\{\ell\}} = (0 = t_0^{\{\ell\}}, t_1^{\{\ell\}}, \dots, t_{N_\ell}^{\{\ell\}} = T) \quad \text{and} \quad \Delta t_n^{\{\ell\}} := t_{n+1}^{\{\ell\}} - t_n^{\{\ell\}}.$$

The meshes are adaptively refined from a given initial, usually but not necessarily, uniform mesh $\Delta t^{\{-1\}}$ in a sequential manner such that $\Delta t^{\{\ell-1\}} \subset \Delta t^{\{\ell\}}$ for all $\ell \in \{0, 1, \dots, L\}$. On level ℓ the mesh is constructed with the aim that the time discretization error in the approximation of $\mathrm{E}[g(\overline{X}_\ell(T))]$ fulfills

$$|\mathrm{E}[g(\overline{X}_\ell(T)) - g(X(T))]| < 2^{L-\ell}\mathrm{TOL}_\mathrm{T} =: \mathrm{TOL}_{\mathrm{T},\ell}, \tag{2.1}$$

where $\overline{X}_\ell(T)$ denotes an Euler–Maruyama approximation of the SDE (1.1) on the mesh $\Delta t^{\{\ell\}}$. The number of mesh levels L is chosen so that the largest tolerance

$$\mathrm{TOL}_{\mathrm{T},0} = 2^L \mathrm{TOL}_\mathrm{T} \tag{2.2}$$

is much larger than TOL_T and results in a quite coarse mesh on level 0. To be more precise, with a rough estimate of the magnitude of $\mathrm{E}[g(X(T))]$ taken into account we prescribe an upper bound[4] $\mathrm{TOL}_{\mathrm{T,Max}}$ for $\mathrm{TOL}_{\mathrm{T},0}$ and determine L by the equation

$$L = \lfloor \log_2(\mathrm{TOL}_{\mathrm{T,Max}}/\mathrm{TOL}_\mathrm{T}) \rfloor. \tag{2.3}$$

For the construction of a time step refinement criterion we introduce the following notation for the mean number of time steps of the accepted mesh on level ℓ:

$$\mathcal{N}_\ell := \mathrm{E}\Bigg[\int_0^T \frac{1}{\Delta t^{\{\ell\}}(\tau)} d\tau\Bigg], \tag{2.4}$$

and $\Delta t^{\{\ell\}}(\cdot) : [0, T] \to \mathbb{R}_+$ denotes the step function

$$\Delta t^{\{\ell\}}(\tau) := \Delta t_{n(\tau)}^{\{\ell\}}, \quad \text{where } n(\tau) := \{m \in \{0, 1, \dots, N_\ell - 1\} \mid t_m^{\{\ell\}} \le \tau < t_{m+1}^{\{\ell\}}\}.$$

Furthermore, for a set of M independent samples, we let

$$\mathcal{A}(f; M) := \frac{1}{M}\sum_{i=1}^{M} f(\omega_i) \quad \text{and} \quad \mathcal{V}(f; M) := \frac{1}{M-1}\sum_{i=1}^{M}(f(\omega_i) - \mathcal{A}(f; M))^2 \tag{2.5}$$

denote the sample average operator and the sample variance operator, respectively.

The inputs in Algorithm 1 are: initial mesh $\Delta t^{\{-1\}}$, initial number of sample realizations M_{-1}, time discretization error tolerance TOL_T, grid levels L, initial estimate of the number of time steps on the accepted coarse mesh $\overline{\mathcal{N}}_0$ (i.e., $\overline{\mathcal{N}}_0 \approx \mathcal{N}_0$), and the three parameters C_R, C_S, and R which are all used in the refinement and stopping conditions (2.7), (2.6), and (2.10), respectively. We choose the initial estimated number of time steps $\overline{\mathcal{N}}_0$ as a small integer not smaller than the number of steps in $\Delta t^{\{-1\}}$.

[4] For example take $\mathrm{TOL}_{\mathrm{T,Max}}$ as half the estimated value of $\mathrm{E}[g(X(T))]$.

On a given level ℓ, the output mesh $\Delta t^{\{\ell\}}$ is computed by first setting $\Delta t^{\{\ell\}} = \Delta t^{\{\ell-1\}}$, $M_\ell = M_{\ell-1}$, and $\overline{\mathcal{N}}_\ell = 2\overline{\mathcal{N}}_{\ell-1}$ ($\overline{\mathcal{N}}_\ell$ is an estimate of the generally unknown value $\mathcal{N}_\ell$ defined in (2.4)). Thereafter, M_ℓ realizations of $g(\overline{X}_\ell(T))$ are generated on the mesh $\Delta t^{\{\ell\}}$ and the sampled error indicators $r_{[\ell]}(n)$, as defined in equation (1.25), are computed for all the time steps of the mesh on each of the M_ℓ generated realizations. With N_ℓ denoting the the number of timesteps in the present mesh $\Delta t^{\{\ell\}}$, the mesh is accepted if the stopping condition

$$\max_{1\le n\le N_\ell} \mathcal{A}(r_{[\ell]}(n); M_\ell) < C_{\mathrm{S}} \frac{\mathrm{TOL}_\ell}{\overline{\mathcal{N}}_\ell} \tag{2.6}$$

is fulfilled. Otherwise, the n-th time step is refined by splitting it into two equal parts if

$$\mathcal{A}(r_{[\ell]}(n); M_\ell) \ge C_{\mathrm{R}} \frac{\mathrm{TOL}_\ell}{\overline{\mathcal{N}}_\ell}. \tag{2.7}$$

Normally, the value for C_{R} would be around 2, and one must take $C_{\mathrm{S}} > C_{\mathrm{R}}$ following the theory developed in [25, 26]. If the mesh is refined, the Wiener increments of each of the M_ℓ realizations of $g(\overline{X}_\ell(T))$ is correspondingly refined by Brownian bridge interpolation, N_ℓ is set to the number of time steps in the refined mesh, the estimated mean number of time steps is updated to $\overline{\mathcal{N}}_\ell = \max\{\overline{\mathcal{N}}_\ell, N_\ell\}$, and the realizations of $g(\overline{X}_\ell(T))$ are recomputed on the refined mesh. This proecdure is repeated until the stopping condition (2.6) is fulfilled.

The adaptive refinements of the computational grid are based on the sample averaged error indicators $\mathcal{A}(r_{[\ell]}(n); M_\ell)$. To estimate the mean error indicators $\mathrm{E}[r_{[\ell]}(n)]$ with sufficient accuracy, we need a mechanism for determining how many samples to use in the sample averages, i.e., M_ℓ. With $\mathtt{E}_{\Delta t^{\{\ell\}}}$ denoting the computed estimate of the time discretization error, i.e.,

$$\mathtt{E}_{\Delta t^{\{\ell\}}} = \sum_{n=1}^{N_\ell} \mathcal{A}(r_{[\ell]}(n); M_\ell), \tag{2.8}$$

a reasonable reliability requirement is

$$\sqrt{\mathrm{Var}(\mathtt{E}_{\Delta t^{\{\ell\}}})} < R\,\mathrm{E}[\mathtt{E}_{\Delta t^{\{\ell\}}}] \tag{2.9}$$

for some suitably chosen $0 < R < 1$. In our numerical examples, for instance, we use $R = 0.2$. The variance of $\mathtt{E}_{\Delta t^{\{\ell\}}}$ is however unknown, but the i.i.d. distribution of the sampled error indicators motivates the approximation

$$\mathrm{Var}(\mathtt{E}_{\Delta t^{\{\ell\}}}) \approx \frac{\mathcal{V}(\sum_{n=1}^{N_\ell} r_{[\ell]}(n); M_\ell)}{M_\ell} \quad \text{for } \ell = 0, 1, \ldots, L.$$

We consequently approximate the reliability requirement (2.9) by

$$\sqrt{\frac{\mathcal{V}(\sum_{n=1}^{N_\ell} r_{[\ell]}(n); M_\ell)}{M_\ell}} < R\,\mathtt{E}_{\Delta t^{\{\ell\}}} \quad \text{for } \ell = 0, 1, \ldots, L, \tag{2.10}$$

where the number of sample realizations M_ℓ used on level ℓ in the grid construction phase is increased by repeated doubling, i.e., $M_\ell = 2\,M_\ell$, until inequality (2.10) is satisfied. As described earlier, the initial batch size at each level is set by $M_\ell = M_{\ell-1}$, where $M_{\ell-1}$ denotes the stopped number of samples at level $\ell - 1$, and for level $\ell = 0$ it turns out to be sufficient to use initial batch size $M_0 = M_{-1}$ with

$$M_{-1} = \mathrm{const} \cdot \mathrm{TOL}_{\mathrm{T}}^{-1}. \tag{2.11}$$

The adaptive algorithm that generates the above described mesh hierarchy for the deterministic time stepping adaptive MLMC algorithm is presented in Algorithm 1–2 in Section 2.3.

2.1.2 Multilevel simulations on a given hierarchy

In the second phase we will describe the algorithms which ensure that our adaptive MLMC estimate of $\mathrm{E}[g(\overline{X}_L(T))]$ fulfills the statistical error bound

$$\mathcal{E}_S = |\mathcal{A}_{\mathcal{ML}}(g(\overline{X}(T)); M_0) - \mathrm{E}[g(\overline{X}_L(T))]| \le \mathrm{TOL}_{\mathrm{S}}, \tag{2.12}$$

with probability close to one. We recall from (1.3) that the multilevel estimator is defined by

$$\mathcal{A}_{\mathcal{ML}}(g(\overline{X}(T)); M_0) = \mathcal{A}(g(\overline{X}_0(T)); M_0) + \sum_{\ell=1}^{L} \mathcal{A}(g(\overline{X}_\ell(T)) - g(\overline{X}_{\ell-1}(T)); M_\ell), \tag{2.13}$$

where the realization pairs $\overline{X}_\ell(T;\omega_{i,\ell})$ and $\overline{X}_{\ell-1}(T;\omega_{i,\ell})$ that are used in $g(\overline{X}_\ell(T;\omega_{i,\ell})) - g(\overline{X}_{\ell-1}(T;\omega_{i,\ell}))$ for each level $\ell > 0$ are generated by the Euler–Maruyama method (1.5) using the same Brownian path $W(t;\omega_{i,\ell})$ on the respective *different* temporal meshes $\Delta t^{\{\ell\}}$ and $\Delta t^{\{\ell-1\}}$ that were computed by Algorithm 1, which is presented in Section 2.3. Furthermore, all Brownian paths $\{W(t;\omega_{i,\ell})\}_{i,\ell}$ are independent, and the number of samples at the coarsest level is set to $M_0 = 2^{L+\lceil C_{\mathcal{ML}} L\rceil+1}$ for a suitable constant $C_{\mathcal{ML}} \in (0,1)$, cf. Remark 4.11, and the number of samples on higher levels is expressed in terms of M_0 by the ratio

$$M_\ell = \frac{M_0}{2^L}\left\lceil 2^L \frac{\rho_{\mathrm{low}}(\mathrm{TOL}_{\mathrm{T},0})\mathrm{TOL}_{\mathrm{T},\ell}}{\rho_{\mathrm{low}}(\mathrm{TOL}_{\mathrm{T},\ell})\mathrm{TOL}_{\mathrm{T},0}}\right\rceil, \quad \ell = 1,\dots,L, \tag{2.14}$$

where ρ_{low} is the lower bound for the error density introduced in (1.23) and $\lceil\cdot\rceil$ denotes rounding upwards to the nearest integer. The enforced lower bound for the sample sets $\{M_\ell\}_{\ell=0}^{L}$ implies that $M_L \to \infty$ as $\mathrm{TOL} \downarrow 0$, and this motivates the approximation of

$$\frac{\mathcal{A}_{\mathcal{ML}}(g(\overline{X}(T)); M_0) - \mathrm{E}[g(\overline{X}_L(T))]}{\sqrt{\mathrm{Var}(\mathcal{A}_{\mathcal{ML}}(g(\overline{X}(T)); M_0))}}$$

by a normal distributed random variable; see Lemma 4.13 in Section 4 for a justification of this approximation for the stochastic time stepping algorithm. Relying on this approximation, the statistical error (2.12) will be controlled by bounding the multilevel estimator variance

$$\sqrt{\mathrm{Var}(\mathcal{A}_{\mathcal{ML}}(g(\overline{X}(T)); M_0))} \le \frac{\mathrm{TOL}_{\mathrm{S}}}{C_C},$$

for a given positive confidence parameter C_{C}. The variance $\mathrm{Var}(\mathcal{A}_{\mathcal{ML}}(g(\overline{X}(T)); M_0))$ is however unknown, so we introduce the following approximation:

$$\mathrm{Var}(\mathcal{A}_{\mathcal{ML}}(g(\overline{X}(T)); M_0)) \approx \underbrace{\frac{\mathcal{V}(g(\overline{X}_0(T)); M_0)}{M_0} + \sum_{\ell=1}^{L} \frac{\mathcal{V}(g(\overline{X}_\ell(T)) - g(\overline{X}_{\ell-1}(T)); M_\ell)}{M_\ell}}_{=:\sigma^2}. \tag{2.15}$$

Our stopping criterion for the Monte Carlo simulations then becomes

$$\sigma < \frac{\mathrm{TOL}_{\mathrm{S}}}{C_{\mathrm{C}}}. \tag{2.16}$$

Until this condition is fulfilled, the number of samples is iteratively doubled ($M_0 = 2M_0$) and the number of samples at the levels $\{M_\ell\}_{\ell=1}^{L}$ are updated according the ratio (2.14), and a new sample estimate $\mathcal{A}_{\mathcal{ML}}(g(\overline{X}(T)); M_0)$ is generated using the multilevel estimator (2.13). Having determined M_0, we lastly generate and return the output estimate $\mathcal{A}_{\mathcal{ML}}(g(\overline{X}(T)); M_0)$.

The probability of controlling the statistical error, i.e., fulfilling the event (2.12) depends on the chosen value for the confidence parameter C_{C}. For example, with $C_{\mathrm{C}} = 1.65$ the event

$$|\mathcal{A}_{\mathcal{ML}}(g(\overline{X}(T)); M_0) - \mathrm{E}[g(\overline{X}_L(T))]| < C_{\mathrm{C}}\,\sigma$$

occurs with probability greater than 0.9, asymptotically as $\mathrm{TOL} \downarrow 0$. See Algorithm 3–5 in Section 2.3 for more details on the MLMC algorithms approximating $\mathrm{E}[g(\overline{X}_L(T))]$ with the deterministic time stepping algorithm. We refer to [2] for a performance study of this type of Monte Carlo sequential stopping rules.

2.2 Stochastic time stepping

In this subsection we describe the stochastic time stepping MLMC algorithm for approximating the expectation $\mathrm{E}[g(X(T))]$. Quite similar to the setting of path independent time steps, the error control of the MLMC estimate $|\mathcal{A}_{\mathcal{ML}}(g(\overline{X}(T)); M_0) - \mathrm{E}[g(X(T))]|$ is in this setting based on constructing numerical realizations $\overline{X}_\ell(t)$

on *stochastic* adaptively refined meshes $\Delta t^{\{\ell\}}$ so that the time discretization errors

$$|\mathrm{E}[g(\overline{X}_\ell(T)) - g(X(T))]| \leq \mathrm{TOL}_{\mathrm{T},\ell} \quad \text{for } \ell = 0, 1, \dots, L \tag{2.17}$$

are asymptotically fulfilled, and by determining the number of samples M_0 to ensure that the statistical error

$$|\mathcal{A}_{\mathcal{ML}}(g(\overline{X}(T)); M_0) - \mathrm{E}[g(\overline{X}_L(T))]| \leq \mathrm{TOL}_\mathrm{S} \tag{2.18}$$

is fulfilled with a given confidence.

The control of the statistical error (2.18) is very similar to that in the setting of path independent time steps:

(1) Set the initial number of samples used in the MLMC estimator (2.13) to $M_0 = 2^{L+\lceil C_{\mathcal{ML}} L\rceil+1}$ with $C_{\mathcal{ML}} \in (0, 1)$, cf. Remark 4.11.
(2) Configure the number of samples M_ℓ on higher levels in terms of M_0 by the ratio (2.14).
(3) Generate realizations $\{\overline{X}_\ell(T)\}$ for the multilevel estimator $\mathcal{A}_{\mathcal{ML}}(g(\overline{X}(T)); M_0)$ and compute the sample variance σ^2 as defined in (2.15).
(4) If the stopping condition (2.16) is fulfilled, generate a last output estimate $\mathcal{A}_{\mathcal{ML}}(g(\overline{X}(T)); M_0)$ and break. Otherwise, set $M_0 = 2M_0$, update the algorithm parameter estimating the mean number of time steps on each grid level,[5] and return to step (2).

For the ℓ-th sample average summand of the multilevel estimator $\mathcal{A}_{\mathcal{ML}}(g(\overline{X}(T)); M_0)$, i.e., $\mathcal{A}(g(\overline{X}_0(T); M_0)$ if $\ell = 0$ and $\mathcal{A}(g(\overline{X}_\ell(T) - g(\overline{X}_{\ell-1}(T)); M_\ell)$ if $\ell > 0$, the algorithm generates M_ℓ Euler–Maruyama realization pairs,[6] $(\overline{X}_{\ell-1}(T), \overline{X}_\ell(T))$ according to (1.5) with the time discretization errors respectively bounded by $\mathrm{TOL}_{\mathrm{T},\ell-1}$ and $\mathrm{TOL}_{\mathrm{T},\ell}$ in the sense (2.17). The realization pairs are constructed by stochastic adaptive refinements of a given initial mesh $\Delta t^{\{-1\}}$. The realizations in a realization pair $(\overline{X}_{\ell-1}(T), \overline{X}_\ell(T))$ are respectively generated on the adaptively refined meshes $\Delta t^{\{\ell-1\}}$ and $\Delta t^{\{\ell\}}$. These meshes are determined by iteratively refining an initial mesh $\Delta t^{\{-1\}}$. First, $\Delta t^{\{-1\}}$ is adaptively refined to a mesh $\Delta t^{\{0\}}$ on which $|\mathrm{E}[g(\overline{X}_0(T)) - g(X(T))]| \lesssim \mathrm{TOL}_{\mathrm{T},0}$ is fulfilled. Thereafter, $\Delta t^{\{0\}}$ is adaptively refined to a mesh $\Delta t^{\{1\}}$ on which $|\mathrm{E}[g(\overline{X}_1(T)) - g(X(T))]| \lesssim \mathrm{TOL}_{\mathrm{T},1}$ is fulfilled. This iterative refinement procedure continues until the mesh $\Delta t^{\{\ell-2\}}$ is adaptively refined to generate the first output mesh $\Delta t^{\{\ell-1\}}$ and, lastly, $\Delta t^{\{\ell-1\}}$ is adaptively refined to generate the second output mesh $\Delta t^{\{\ell\}}$.

The iterative adaptive mesh refinement procedure in Algorithm 7, Section 2.3, ensures that a mesh $\Delta t^{\{\ell\}}$ for the fine realization in a pair $(\overline{X}_{\ell-1}(T), \overline{X}_\ell(T))$ is determined in the same way as a mesh $\Delta t^{\{\ell\}}$ for the coarse realization in pair $(\overline{X}_\ell(T), \overline{X}_{\ell+1}(T))$, and consequently that $\mathrm{E}[g(\overline{X}_\ell(T))]$ when computed from the finer realization in a pair $(\overline{X}_{\ell-1}(T), \overline{X}_\ell(T))$ is equal to $\mathrm{E}[g(\overline{X}_\ell(T))]$ when computed from the coarse realization in a pair $(\overline{X}_\ell(T), \overline{X}_{\ell+1}(T))$. This construction is one way to guarantee that the consistency condition

$$\mathrm{E}[\mathcal{A}_{\mathcal{ML}}(g(\overline{X}(T)); M_0)] = \mathrm{E}[g(\overline{X}_L(T))]$$

for the multilevel estimator is fulfilled.

Let us next take a closer look at the mesh refinement. Due to the stochastic nature of SDEs, each realization pair $(\overline{X}_{\ell-1}(T), \overline{X}_\ell(T))$ may refine the initial mesh Δt_{-1} differently. In particular, meshes corresponding to different realizations on a given level ℓ may differ. To describe the mesh refinement, taking this feature into account, we introduce some notation. Since statistics on the number time steps in a mesh is important for the mesh refinement algorithm, we introduce the following notation the number of time steps in a mesh realization $\Delta t^{\{\ell\}}(\omega)$:

$$N_\ell(\omega) := \int_0^T \frac{1}{\Delta t^{\{\ell\}}(\tau; \omega)} d\tau.$$

Furthermore, write $\mathcal{N}_\ell := \mathrm{E}[N_\ell]$ for the mean number of time steps on mesh level ℓ and let $\overline{\mathcal{N}}_\ell$ represent the algorithm parameter approximating $\mathcal{N}_\ell$. See Algorithm 8 in Section 2.3 for details on the approximation technique and the update of $\overline{\mathcal{N}}_\ell$ through the iterations.

[5] See Algorithm 8 for details on the parameter update.

[6] Observe that for the level $\ell = 0$ only the realizations of $\overline{X}_0(T)$ are generated.

The mesh refinement condition (1.25) is based on the error indicator $r_{[\ell]}$ and works in a similar fashion as for the single level method: Refinement of a mesh $\Delta t^{\{\ell\}}$ is stopped when

$$\max_{1\leq n\leq N_\ell} r_{[\ell]}(n) < C_\mathrm{S} \frac{\mathrm{TOL}_{\mathrm{T},\ell}}{\overline{\mathcal{N}}_\ell}, \tag{2.19}$$

but as long as inequality (2.19) is violated, the n^th time step of $\Delta t^{\{\ell\}}$ is refined if

$$r_{[\ell]}(n) \geq C_\mathrm{R} \frac{\mathrm{TOL}_{\mathrm{T},\ell}}{\overline{\mathcal{N}}_\ell}. \tag{2.20}$$

Normally, the value for C_R would be around 2, and $C_\mathrm{S} > C_\mathrm{R}$ following the theory developed in [25, 26].

A detailed description of the adaptive MLMC algorithm is given in Algorithm 6 with subroutines Algorithm 7–9 in Section 2.3.

The inputs in Algorithm 6 are: TOL_S, TOL_T, an initial number of sample realizations M_0, L, $\Delta t^{\{-1\}}$, initial guesses for the mean number of time steps $\{\overline{\mathcal{N}}_\ell\}_{\ell=0}^L$ in the hierarchy of accepted adaptively refined meshes, and the three parameters C_R, C_C, and C_S used in the refinement condition (2.20) and stopping conditions (2.16) and (2.19), respectively. In this algorithm the initial estimate of the mean number of time steps are chosen as $\overline{\mathcal{N}}_\ell = c\mathrm{TOL}_{\mathrm{T},\ell}{}^{-1}$, for $\ell = 0, \dots, L$ and a constant c such that $\overline{\mathcal{N}}_0$ is a small integer; in the numerical examples in Section 3, the constant was chosen so that $\overline{\mathcal{N}}_0 \approx 10$ as input.

2.3 Algorithm listings

Algorithm 1: Adaptive generation of a mesh hierarchy.

```
Input  : $\mathrm{TOL}_\mathrm{T}$, $M_{-1}$, $\Delta t^{\{-1\}}$, $L$, $\overline{\mathcal{N}}_0$, $C_\mathrm{R}$, $C_\mathrm{S}$, $R$
Output: $\{\Delta t^{\{\ell\}}\}_{\ell=0}^L$, $M_L$
for $\ell = 0, 1, \dots, L$ do
    Set keep_sampling = TRUE, keep_refining = TRUE,
    $\Delta t^{\{\ell\}} = \Delta t^{\{\ell-1\}}$, $M_\ell = M_{\ell-1}$, and $\mathrm{TOL}_{\mathrm{T},\ell} = 2^{L-\ell}\mathrm{TOL}_\mathrm{T}$.
    while keep_sampling or keep_refining do
        Set keep_sampling = FALSE, keep_refining = FALSE
        Compute $r_{[\ell]}$, $\mathtt{E}_{\Delta t^{\{\ell\}}}$, and $\mathcal{V}(\sum_{n=1}^{N_\ell} r_{[\ell]}(n); M_\ell)$ by calling Algorithm 2: Euler($M_\ell$, $\Delta t^{\{\ell\}}$)
        if $\mathcal{V}(\sum_{n=1}^{N_\ell} r_{[\ell]}(n); M_\ell)$ and $\mathtt{E}_{\Delta t^{\{\}}}$ violate (2.10) then
            Set keep_sampling = TRUE
            Update the number of samples by
            $M_\ell = 2M_\ell$
        else
            if $r_{[\ell]}$ violates (2.6) then
                Set keep_refining = TRUE
                Refine $\Delta t^{\{\ell\}}$ by
                forall intervals $n = 1, 2, \dots, N_\ell$ do
                    if $r_{[\ell]}(n)$ satisfies (2.7) then
                        divide the interval $n$ into two equal parts
                    end
                end
                Update $N_\ell$ and set $\overline{\mathcal{N}}_\ell = \max\{\overline{\mathcal{N}}_\ell, N_\ell\}$.
            end
        end
    end
    Set $\overline{\mathcal{N}}_{\ell+1} = 2\,\overline{\mathcal{N}}_\ell$
end
```

Algorithm 2: Euler.

```
Input  : $M_\ell$, $\Delta t^{\{\ell\}}$
Output: $r_{[\ell]}$, $\mathtt{E}_{\Delta t^{\{\ell\}}}$, $\mathcal{V}(\sum_{n=1}^{N_\ell} r_{[\ell]}(n); M_\ell)$

Compute $M_\ell$ new realizations of $\overline{X}_\ell$ on $\Delta t^{\{\ell\}}$ by Euler–Maruyama method (1.5) and use them to
compute the error indicators $r_{[\ell]}(n)$ on $\Delta t^{\{\ell\}}$ by equation (1.25), $\mathtt{E}_{\Delta t^{\{\ell\}}}$ by equation (2.8), and
$\mathcal{V}(\sum_{n=1}^{N_\ell} r_{[\ell]}(n); M_\ell)$ by equation (2.5).
```

Algorithm 3: Multilevel Monte Carlo on a mesh hierarchy.

```
Input  : $\mathrm{TOL_S}$, $M_0$, $L$, $\{\Delta t^{\{\ell\}}\}_{\ell=0}^L$, $C_\mathrm{C}$
Output: $\mu = \mathcal{A}_{\mathcal{ML}}(g(\overline{X}(T)); M_0)$

Compute $\sigma^2$ by Algorithm 4: MLMC Estimator($M_0, L, \{\Delta t^{\{\ell\}}\}_{\ell=0}^L$).
while $\sigma^2$ violates (2.16) do
    Update the number of samples by $M_0 = 2M_0$.
    Update $\sigma^2$ by Algorithm 4: MLMC Estimator($M_0, L, \{\Delta t^{\{\ell\}}\}_{\ell=0}^L$).
end
Generate the output $\mu = \mathcal{A}_{\mathcal{ML}}(g(\overline{X}(T)); M_0)$ by calling Algorithm 4: MLMC Estimator($M_0, L, \{\Delta t^{\{\ell\}}\}_{\ell=0}^L$).
```

Algorithm 4: MLMC estimator.

```
Input  : $M_0$, $L$, $\{\Delta t^{\{\ell\}}\}_{\ell=0}^L$
Output: $\mu = \mathcal{A}_{\mathcal{ML}}(g(\overline{X}(T)); M_0)$, $\sigma^2 \approx \mathrm{Var}(\mathcal{A}_{\mathcal{ML}}(g(\overline{X}(T)); M_0))$

for $\ell = 0, 1, \dots, L$ do
    Set $M_\ell$ as in (2.14)
    if $\ell = 0$ then
        Call Algorithm 5: Euler($M_0, \{\Delta t^{\{0\}}\}$).
        Set $\mu = \mathcal{A}(g(\overline{X}_0(T)); M_\ell)$ and $\sigma^2 = \frac{\mathcal{V}(g(\overline{X}_0(T)); M_0)}{M_0}$.
    else
        Call Algorithm 5: Euler($M_\ell, \{\Delta t^{\{\ell\}}, \Delta t^{\{\ell-1\}}\}$).
        Set $\mu = \mu + \mathcal{A}(g(\overline{X}_\ell(T)) - g(\overline{X}_{\ell-1}(T)); M_\ell)$ and $\sigma^2 = \sigma^2 + \frac{\mathcal{V}(g(\overline{X}_\ell(T)) - g(\overline{X}_{\ell-1}(T)); M_\ell)}{M_\ell}$.
    end
end
```

Algorithm 5: Euler.

```
Input  : $M$, $\{\Delta t^{\{\ell\}}\}_{\ell=l_0,l_1}$
Output: $\mathcal{V}(g(\overline{X}_0(T)); M)$, $\mathcal{A}(g(\overline{X}_0(T)); M)$ if $l_0 = l_1 = 0$ or
$\mathcal{V}(g(\overline{X}_{\ell_1}(T)) - g(\overline{X}_{\ell_0}(T)); M)$, $\mathcal{A}(g(\overline{X}_{\ell_1}(T)) - g(\overline{X}_{\ell_0}(T); M)$ if $l_0 \neq l_1$

Simulate $M$ new outcomes of the Wiener process $W$ on $\Delta t^{\{\ell_1\}} \supseteq \Delta t^{\{\ell_0\}}$.
if $l_0 = l_1 = 0$ then
    Compute the corresponding realizations of $\overline{X}_0$ on $\Delta t^{\{0\}}$ and use them to compute $\mathcal{A}(g(\overline{X}_0(T)); M)$
    and $\mathcal{V}(g(\overline{X}_0(T)); M)$ by (2.5).
else
    Compute the corresponding realizations of $\overline{X}_{\ell_1}$ and $\overline{X}_{\ell_0}$ on $\Delta t^{\{\ell_1\}}$ and $\Delta t^{\{\ell_0\}}$ and use them to
    compute $\mathcal{A}(g(\overline{X}_{\ell_1}(T)) - g(\overline{X}_{\ell_0}(T)); M)$ and $\mathcal{V}(g(\overline{X}_{\ell_1}(T)) - g(\overline{X}_{\ell_0}(T)); M)$ by (2.5).
end
```

Algorithm 6: Multilevel Monte Carlo with stochastic time stepping.

```
Input  : $\mathrm{TOL}_\mathrm{S}$, $\mathrm{TOL}_\mathrm{T}$, $M_0$, $\Delta t^{\{-1\}}$, $L$, $\{\overline{\mathcal{N}}_\ell\}_{\ell=0}^L$, $C_\mathrm{R}$, $C_\mathrm{S}$, $C_\mathrm{C}$
Output: $\mu = \mathcal{A}_{\mathcal{ML}}(g(\overline{X}(T)); M_0)$

Compute $\sigma^2$ and $\{\overline{\mathcal{N}}_\ell\}_{\ell=0}^L$ by calling Algorithm 7: PMLMC ($\mathrm{TOL}_\mathrm{T}, M_0, \Delta t^{\{-1\}}, L, \{\overline{\mathcal{N}}_\ell\}_{\ell=0}^L, C_\mathrm{R}, C_\mathrm{S}$).
while $\sigma^2$ violates (2.16) do
    Update the number of samples by $M_0 = 2M_0$.
    Update $\sigma^2$ and $\{\overline{\mathcal{N}}_\ell\}_{\ell=0}^L$ by Algorithm 7: PMLMC($\mathrm{TOL}_\mathrm{T}, M_0, \Delta t^{\{-1\}}, L, \{\overline{\mathcal{N}}_\ell\}_{\ell=0}^L, C_\mathrm{R}, C_\mathrm{S}$).
end
Generate the output $\mu = \mathcal{A}_{\mathcal{ML}}(g(\overline{X}(T)); M_0)$ by calling Algorithm 7: PMLMC($\mathrm{TOL}_\mathrm{T}, M_0, \Delta t^{\{-1\}}, L, \{\overline{\mathcal{N}}_\ell\}_{\ell=0}^L, C_\mathrm{R}, C_\mathrm{S}$).
```

Algorithm 7: Pathwise multilevel Monte Carlo estimator (**PMLMC**).

```
Input  : $\mathrm{TOL}_\mathrm{T}$, $M_0$, $\Delta t^{\{-1\}}$, $L$, $\{\overline{\mathcal{N}}_\ell\}_{\ell=0}^L$, $C_\mathrm{R}$, $C_\mathrm{S}$
Output: $\mu = \mathcal{A}_{\mathcal{ML}}(g(\overline{X}(T)); M_0)$, $\sigma^2 \approx \mathrm{Var}(\mathcal{A}_{\mathcal{ML}}(g(\overline{X}(T)); M_0))$, $\{\overline{\mathcal{N}}_\ell\}_{\ell=0}^L$

Compute $M_0$ samples of $g(\overline{X}_0(T))$ and the number of time steps used, $\{N_{0,m}\}_{m=1}^{M_0}$, by generating Wiener increments $\{\Delta W_{-1,m}\}_{m=1}^{M_0}$ on the mesh $\Delta t^{\{-1\}}$ (independently for each realization $m$) and calling Algorithm 9: ATSSE($\Delta t^{\{-1\}}, \Delta W_{-1,m}, \mathrm{TOL}_\mathrm{T} 2^L, \overline{\mathcal{N}}_0, C_\mathrm{R}, C_\mathrm{S}$).
Set $\mu = \mathcal{A}(g(\overline{X}_0(T)); M_0)$ and $\sigma^2 = \frac{\mathcal{V}(g(\overline{X}_0(T)); M_0)}{M_0}$.
Compute the average number of time steps $\mathcal{A}(N_0; M_0)$.
for $\ell = 1, \dots, L$ do
    Set $M_\ell$ as in (2.14) and compute $M_\ell$ new realizations of $g(\overline{X}_{\ell-1}(T))$, their corresponding number of time steps, $\{N_{\ell-1,m}\}_{m=1}^{M_\ell}$, and Wiener increments, $\{\Delta W_{\ell-1,m}\}_{m=1}^{M_\ell}$, by generating Wiener steps $\{\Delta W_{-1,m}\}_{m=1}^{M_0}$ on the mesh $\Delta t^{\{-1\}}$ (independently for each realization $m$) and using the loop
    for $\hat{\ell} = 0, \dots, \ell - 1$ do
        compute $\Delta t^{\{\hat{\ell},m\}}$, $\Delta W_{\hat{\ell},m}$ by calling Algorithm 9: ATSSE($\Delta t^{\{\hat{\ell}-1,m\}}, \Delta W_{\hat{\ell}-1,m}, \mathrm{TOL}_\mathrm{T} 2^{L-\hat{\ell}}, \overline{\mathcal{N}}_{\hat{\ell}}, C_\mathrm{R}, C_\mathrm{S}$).
    end
    Compute the corresponding $M_\ell$ realizations of $g(\overline{X}_\ell(T))$ and their number of time steps, $\{N_{\ell,m}\}_{m=1}^{M_\ell}$, by calling Algorithm 9: ATSSE($\Delta t^{\{\ell-1,m\}}, \Delta W_{\ell-1,m}, \mathrm{TOL}_\mathrm{T} 2^{L-\ell}, \overline{\mathcal{N}}_\ell, C_\mathrm{R}, C_\mathrm{S}$).
    Set $\mu = \mu + \mathcal{A}(g(\overline{X}_\ell(T)) - g(\overline{X}_{\ell-1}(T)); M_\ell)$ and $\sigma^2 = \sigma^2 + \frac{\mathcal{V}(g(\overline{X}_\ell(T)) - g(\overline{X}_{\ell-1}(T)); M_\ell)}{M_\ell}$.
    Compute average number of time steps $\mathcal{A}(N_{\ell-1}; M_\ell)$ and $\mathcal{A}(N_\ell; M_\ell)$.
end
Update the values of $\{\overline{\mathcal{N}}_\ell\}_{\ell=0}^L$ by calling Algorithm 8: UMNT ($\{M_\ell\}_{\ell=0}^L$, $\{\mathcal{A}(N_\ell; M_\ell)\}_{\ell=0}^L$, $\{\mathcal{A}(N_{\ell-1}; M_\ell)\}_{\ell=1}^L$).
```

Algorithm 8: Update for the mean number of time steps (**UMNT**).

```
Input  : $\{M_\ell\}_{\ell=0}^L$, $\{\mathcal{A}(N_\ell; M_\ell)\}_{\ell=0}^L$, $\{\mathcal{A}(N_{\ell-1}; M_\ell)\}_{\ell=1}^L$
Output: $\{\overline{\mathcal{N}}_\ell\}_{\ell=0}^L$

for $\ell = 0, 1, \dots, L$ do
    if $\ell < L$ then
        Set $\overline{\mathcal{N}}_\ell = \frac{M_\ell \mathcal{A}(N_\ell; M_\ell) + M_{\ell+1} \mathcal{A}(N_\ell; M_{\ell+1})}{M_\ell + M_{\ell+1}}$.
    else
        Set $\overline{\mathcal{N}}_L = \mathcal{A}(N_L; M_L)$.
    end
end
```

Algorithm 9: Adaptive time step stochastic Euler (**ATSSE**).

```
Input   : $\Delta t^{\{\mathrm{in}\}}, \Delta W_{\mathrm{in}}$, TOL, $\overline{\mathcal{N}}_{\mathrm{in}}, C_{\mathrm{R}}, C_{\mathrm{S}}$
Output: $\Delta t^{\{\mathrm{out}\}}, \Delta W_{\mathrm{out}}, N_{\mathrm{out}}, g_{\mathrm{out}}$
Set $k = 0$, $\Delta t^{\{[0]\}} = \Delta t^{\{\mathrm{in}\}}$, $\Delta W_{\underline{[0]}} = \Delta W_{\mathrm{in}}$, $N_{[0]}$ = number of steps in $\Delta t^{\{\mathrm{in}\}}$
while $k < 1$ or ($r_{[k-1]}$; TOL, $\overline{\mathcal{N}}_{\mathrm{in}}, C_{\mathrm{S}}$) violates (2.19) do
    Compute the Euler approximation $\overline{X}_{[k]}$ and the error indicators $r_{[k]}$ on $\Delta t^{\{[k]\}}$ with the known
    Wiener increments $\Delta W_{[k]}$.
    if ($r_{[k]}$; TOL, $\overline{\mathcal{N}}_{\mathrm{in}}, C_{\mathrm{S}}$) violates (2.19) then
        Refine the grid $\Delta t^{\{[k]\}}$ by
        forall intervals $n = 1, 2, \ldots, N_{[k]}$ do
            if ($r_{[[k]]}(n)$; TOL, $\overline{\mathcal{N}}_{\mathrm{in}}, C_{\mathrm{R}}$) satisfies (2.20) then
                divide the interval $n$ into two equal parts
            end
        end
        and store the refined grid in $\Delta t^{\{[k+1]\}}$.
        Compute $\Delta W_{[k+1]}$ from $\Delta W_{[k]}$ using Brownian bridges on $\Delta t^{\{[k+1]\}}$.
        Set $N_{[k+1]}$ = number of steps in $\Delta t^{\{[k+1]\}}$.
    end
    Increase $k$ by 1.
end
Set $\Delta t^{\{\mathrm{out}\}} = \Delta t^{\{[k-1]\}}$, $\Delta W_{\mathrm{out}} = \Delta W_{[k-1]}$, $N_{\mathrm{out}} = N_{[k-1]}$, $g_{\mathrm{out}} = g(\overline{X}_{[k-1]})$.
```

3 Numerical experiments

This section presents numerical results from implementations[7] of the algorithms introduced in Section 2. We have selected problems to indicate the use of the adaptive methods. Specifically, uniform time steps are suitable for Problem 3.1, adaptively refined deterministic time steps are suitable for Problem 3.2, and fully stochastic time steps are suitable for Problem 3.4. In both Problems 3.2 and 3.4 the use of the multilevel adaptive algorithms is much more efficient than the use of the corresponding single level versions of the algorithms, which is in turn much more efficient than using a single level uniform time stepping method. For those problems the complexity is close to that of uniform MLMC, since the observed order of strong convergence remains close to $1/2$ even though the order of weak convergence is reduced using uniform time steps. As it is described in this work, the adaptive algorithm is optimized with respect to the weak error, but an extension of the adaptive algorithm which is instead optimized with respect to the strong error is the subject of ongoing research.

The main complexity results in Theorem 4.2 and Remark 4.3 of Section 4 are asymtotic results for TOL approaching 0, excluding asymptotically negligible terms. The approximate upper bound

$$\text{cost} \leq C(\mathrm{TOL}^{-1}(1 + \log_2(\mathrm{TOL}_{\mathrm{T},0}/\mathrm{TOL}_{\mathrm{T}}))^2 \tag{3.1}$$

for the computational complexity captures the essence of Remark 4.3 while keeping the logarithmic factor in a form that is also consistent with large tolerances where $L = 0$. For the numerical tests in this section we fit

[7] The implementations differ from the listed algorithms and the theoretical analysis in that the computed answer

$$\mu = \mathcal{A}_{\mathcal{ML}}(g(\overline{X}(T)); M_0)$$

was taken from the same batch that satisfied the stopping criterion (2.16) without generating a final batch of independent samples after accepting M_0. Note that while the extra batch simplifies the theoretical analysis the experimental errors in Figure 2 still satisfy the accuracy requirements, and the repetition of the final batch would increase the total work with a factor approximately between $3/2$ and 2.

the parameters c_1 and c_2 in the model

$$\log_2(\text{cost}) = c_1 + \log_2(\mathrm{TOL}^{-c_2}(1 + \log_2(\mathrm{TOL}_{\mathrm{T},0}/\mathrm{TOL}_{\mathrm{T}}))^2) \tag{3.2}$$

to the observed computational costs, where by (3.1) we expect $c_2 \approx 2$.

The computations were performed in `Matlab 7` using the built in pseudo-random number generator `randn` for simulating sampling from the normal distribution. In all examples the error tolerance was split equally,

$$\mathrm{TOL}_{\mathrm{S}} = \mathrm{TOL}_{\mathrm{T}} = \frac{\mathrm{TOL}}{2},$$

even though the proof of Theorem 4.2 indicates that this is not optimal; see Remark 4.17. The bounds on the computed error density in (1.10) were $\rho_{\mathrm{low}} = \mathrm{TOL}^{1/9}$ and $\rho_{\mathrm{up}} = \mathrm{TOL}^{-4}$. The confidence parameter was $C_C = 1.65$ corresponding to a 90% confidence interval of the standard normal random variable. For the parameter in the stopping criteria (2.6) and (2.19) we used $C_{\mathrm{S}} = 5$ in Problems 3.2 and 3.4, and $C_{\mathrm{S}} = 3$ in Problem 3.1 where we expect uniform refinements and all error indicators of the same size. The values of the other parameters are listed in Table 1. The particular values of are not necessarily optimized for the problems at hand, but we include them for the purpose of reproducibility.

	Algorithm 1 and 3	
	GBM, Section 3.1	Singularity, Section 3.2
$\Delta t^{\{-1\}}$	1/2	1/4
$\overline{\mathcal{N}}_0$	$\left\lceil \frac{6}{\mathrm{TOL}_{\mathrm{T},0}} \right\rceil$	$\left\lceil \frac{2}{\mathrm{TOL}_{\mathrm{T},0}} \right\rceil$
$\mathrm{TOL}_{\mathrm{T,Max}}$	0.6	0.32
M_{-1}	$\left\lceil 5 \cdot \frac{0.25}{\mathrm{TOL}} \right\rceil$, $\mathrm{TOL} \le 0.25$	$\left\lceil 5 \cdot \frac{0.16}{\mathrm{TOL}} \right\rceil$, $\mathrm{TOL} \le 0.16$
C_{R}	2	2
C_{S}	3	5
R	0.2	0.2
C_{C}	1.65	1.65

	Algorithm 6	
	GBM, Section 3.1	Barrier, Section 3.3
$\Delta t^{\{-1\}}$	1/2	1/5
$\overline{\mathcal{N}}_0$	$\left\lceil \frac{6}{\mathrm{TOL}_{\mathrm{T},0}} \right\rceil$	$\left\lceil \frac{10}{\mathrm{TOL}_{\mathrm{T},0}} \right\rceil$
$\mathrm{TOL}_{\mathrm{T,Max}}$	0.6	2
M_0	$\left\lceil 5 \cdot \frac{0.25}{\mathrm{TOL}} 2^{L(1-\gamma)} \right\rceil$, $\mathrm{TOL} \le 0.25$	$\left\lceil 5 \cdot \frac{0.2}{\mathrm{TOL}} 2^{L(1-\gamma)} \right\rceil$, $\mathrm{TOL} \le 0.8$
C_{R}	2	2
C_{S}	3	5
C_{C}	1.65	1.65

Table 1. List of parameter values used in the computations in Section 3.1–3.3. Here L and $\mathrm{TOL}_{\mathrm{T},0}$ are functions of $\mathrm{TOL}_{\mathrm{T,Max}}$ and $\mathrm{TOL}_{\mathrm{T}}$ by (2.2) and (2.3). Further, $\gamma = 1/9$ is the parameter in $\rho_{\mathrm{low}} = \mathrm{TOL}^{\gamma}$.

3.1 A linear SDE

Problem 3.1. Consider first the standard geometric Brownian motion

$$\begin{aligned} dX(t) &= rX(t)dt + \sigma X(t)dW(t), \quad t \in (0, T), \\ X(0) &= 1, \end{aligned}$$

using $r = 1$ and $\sigma = 0.5$ with a final time $T = 1$ and $g(x) = x$.

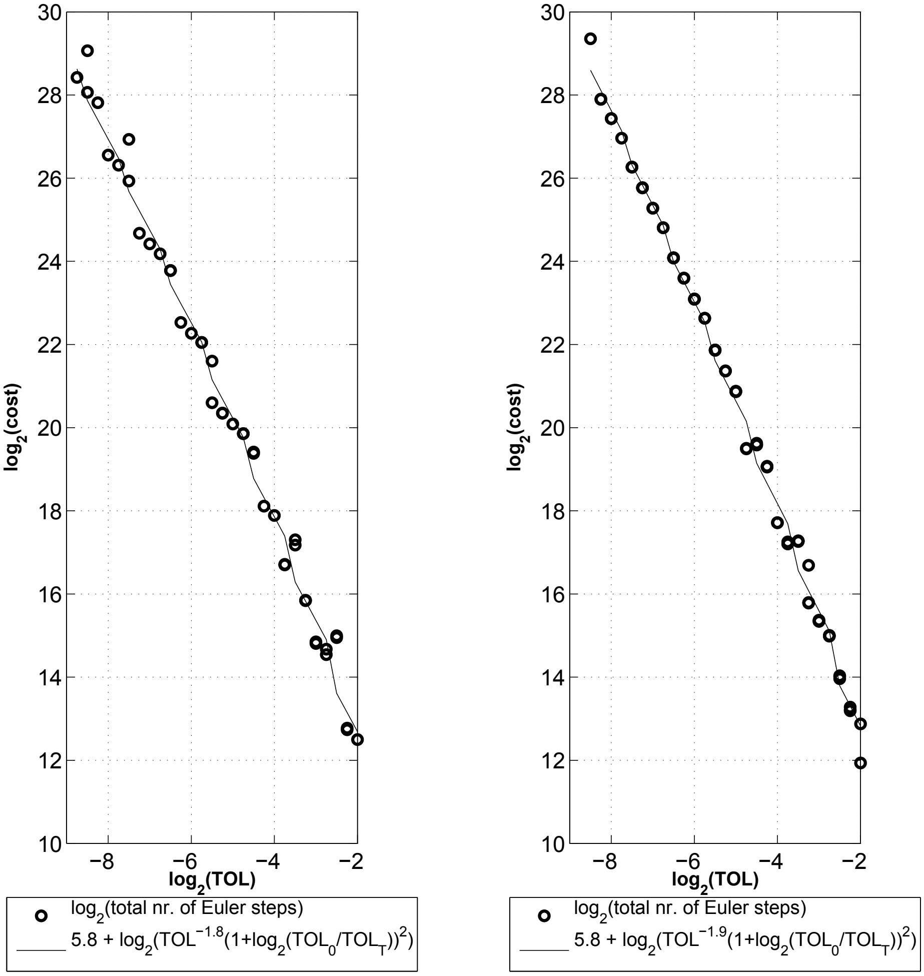

Figure 1. Experimental complexity for both versions of the algorithm applied to the geometrical Brownian motion example of Section 3.1; to the left the version of mesh creation followed by sampling on fixed meshes, in Section 2.1, and to the right the path dependent sampling version in Section 2.2. The computational cost is measured as the total number of Euler time steps taken in all refinement iterations on all levels for all realizations. The graphs show three independent realizations of the underlying Wiener processes for each prescribed tolerance. A least squares fit of the model (3.2) gives $c_2 = 1.8$ and $c_2 = 1.9$ in the two cases respectively; this is slightly better than the prediction of Theorem 4.2 of Section 4.

In this simple example adaptive time stepping is not expected to improve the time discretization error. In fact, the path independent adaptive algorithm produces a hierarchy of uniform grids, and when the fully stochastic adaptive algorithm is applied to this problem all generated meshes are uniform but different realizations of the driving Wiener process may result in different step sizes. The computational cost, measured as the total number of time steps, in all stages in the adaptive refinements, for all realizations of the Euler approximation $\overline{X}$, is shown in Figure 1. For both versions of the algorithm, the computational cost is consistent with the approximate upper bound (3.1) derived from the analyis in Section 4. The work measured this way is very similar in the two versions of the algorithm. However, the version in Section 2.1 is more efficient in this case since it only computes dual solutions in the construction of the mesh hierarchy which is of negligible cost,[8] while the version in Section 2.2 computes both primal and dual for every realization. Since the cost of constructing the mesh hierarchies is asymptotically negligible, and the constructed hierarchies are uniform with geometrically decreasing mesh sizes, the complexity of the adaptive algorithm in Section 2.1 applied to this

[8] See Figure 3 for Problem 3.2.

problem is essentially the same as that of a uniform MLMC algorithm using the same control of the statistical error. The accuracy of both versions of the algorithm is shown in Figure 2.

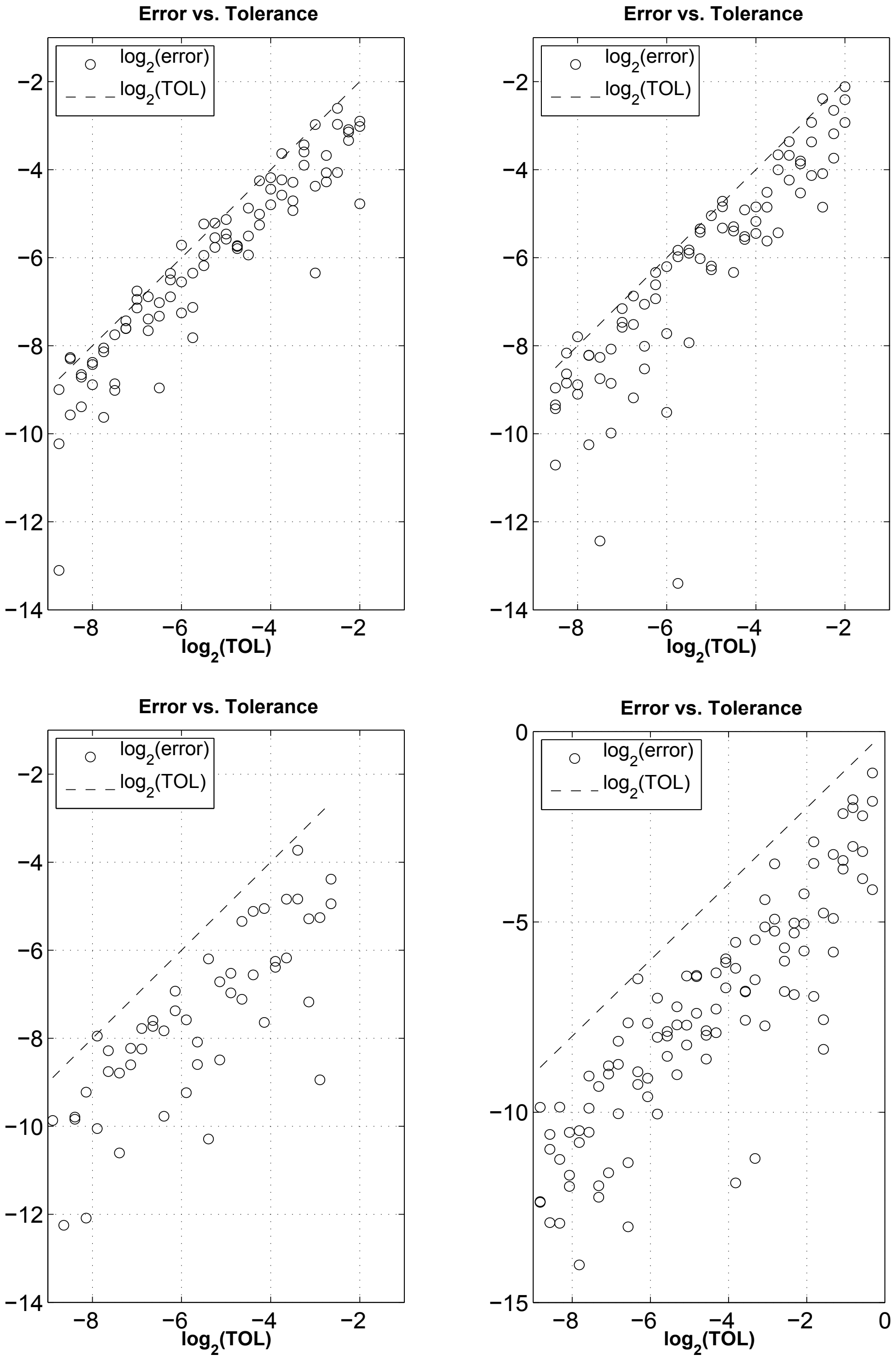

Figure 2. These accuracy tests show the error versus the prescribed tolerance when the adaptive MLMC algorithm is applied to the test examples of Section 3; to the left the version of Section 2.1 applied to the geometric Brownian motion in Section 3.1 (top) and the singularity problem in Section 3.2 (bottom), and to the right the version of Section 2.2 applied to the geometric Brownian motion in Section 3.1 (top) and the stopped diffusion problem in Section 3.3 (bottom).

The work we measure in Figure 1 is greater than the work (4.2) analyzed in Section 4, which is approximately the number of sampled random variables. The comparison made in Table 2 shows the same growth rate as TOL ↓ 0 when the fully stochastic adaptive algorithm is applied to Problem 3.1.

		sampled random variables		**all Euler steps**	
Problem	Version	c_1	c_2	c_1	c_2
GBM	Section 2.1	5.7	1.8	5.8	1.8
GBM	Section 2.2	4.9	1.9	5.8	1.9
Singularity	Section 2.1	10.4	1.9	10.6	1.9
Barrier	Section 2.2	7.3	2.0	8.5	2.2

Table 2. Complexity estimates for the three different problems: the geometric Brownian motion of Section 3.1, the deterministic singularity problem of Section 3.2, and the stopped diffusion problem of Section 3.3. The tabulated values are least square fits of the parameters c_1 and c_2 in the model (3.2) which is approximated by the work estimate defined in (4.2), and by counting the total number of Euler steps performed when solving the primal problem in *all* refinement stages for all levels in the multilevel algorithms.

3.2 Drift singularity, linear SDE

Problem 3.2. Consider for a real constant $\alpha \in (0, T)$ the linear stochastic differential equation

$$dX(t) = \begin{cases} X(t)dW(t), & t \in [0, \alpha], \\ \frac{X(t)}{2\sqrt{t-\alpha}}dt + X(t)dW(t), & t \in (\alpha, T], \end{cases} \tag{3.3}$$

$$X(0) = 1,$$

with the unique solution

$$X(t) = \begin{cases} \exp(W(t) - t/2), & t \in [0, \alpha], \\ \exp(W(t) - t/2)\exp(\sqrt{t-\alpha}), & t \in (\alpha, T]. \end{cases}$$

The goal is to approximate the expected value $\mathrm{E}[X(T)] = \exp(\sqrt{T-\alpha})$.

Here we choose $T = 1$ and $\alpha = T/3$. To avoid evaluating arbitrarily large values of the drift in (3.3) we modify the drift to be

$$a(t, x) = \begin{cases} 0, & t \in [0, \alpha], \\ \frac{x}{2\sqrt{t-\alpha+\mathrm{TOL}^4}}, & t \in (\alpha, T], \end{cases} \tag{3.4}$$

yielding a higher order perturbation $\mathcal{O}(\mathrm{TOL}^2)$ in the computed result and in the size of the optimal time steps. This regularization was applied to maintain consistency with the numerical tests in [25], but it is not strictly necessary given the upper bound, $\rho \leq \rho_{\mathrm{up}}(\mathrm{TOL})$, on the error density in (1.23). Due to the time discontinuity of the drift function and to ensure optimal convergence of the adaptive algorithms, we modify the Euler method to

$$\overline{X}_{n+1} - \overline{X}_n = a(\hat{t}, \overline{X}_n)\,\Delta t_n + \overline{X}_n\,\Delta W_n, \quad n = 0, 1, 2, \dots, \tag{3.5}$$

where we choose the stochastic evaluation time $\hat{t} \in \{t_n, t_{n+1}\}$ so that

$$|a(\hat{t}, \overline{X}_n)| = \max(|a(t_n, \overline{X}_n)|, |a(t_{n+1}, \overline{X}_n)|).$$

Observe that the use of $\hat{t}$ does not change the adapted nature of the Euler method.

Since we now have a singularity in the drift at a deterministic time, the path independent adaptive algorithm described in Section 2.1 is the most suitable, and it is used in this example. The goal here is to verify that the adaptive multilevel algorithms of Section 2 give the same improvement from the single level adaptive algorithm as multilevel Monte Carlo does in the uniform case for regular problems.

The accuracy test in Figure 2 shows good agreement between observed error and prescribed tolerance. As shown in the complexity study in Table 2 and Figure 3 the computational costs grow like

$$\mathrm{TOL}^{-1.9}(1 + \log(\mathrm{TOL}_{\mathrm{T},0}/\mathrm{TOL}_{\mathrm{T}}))^2$$

which is very close to the predicted complexity. The cost of the mesh construction phase of the algorithm is seen to be negligible compared to the total work.

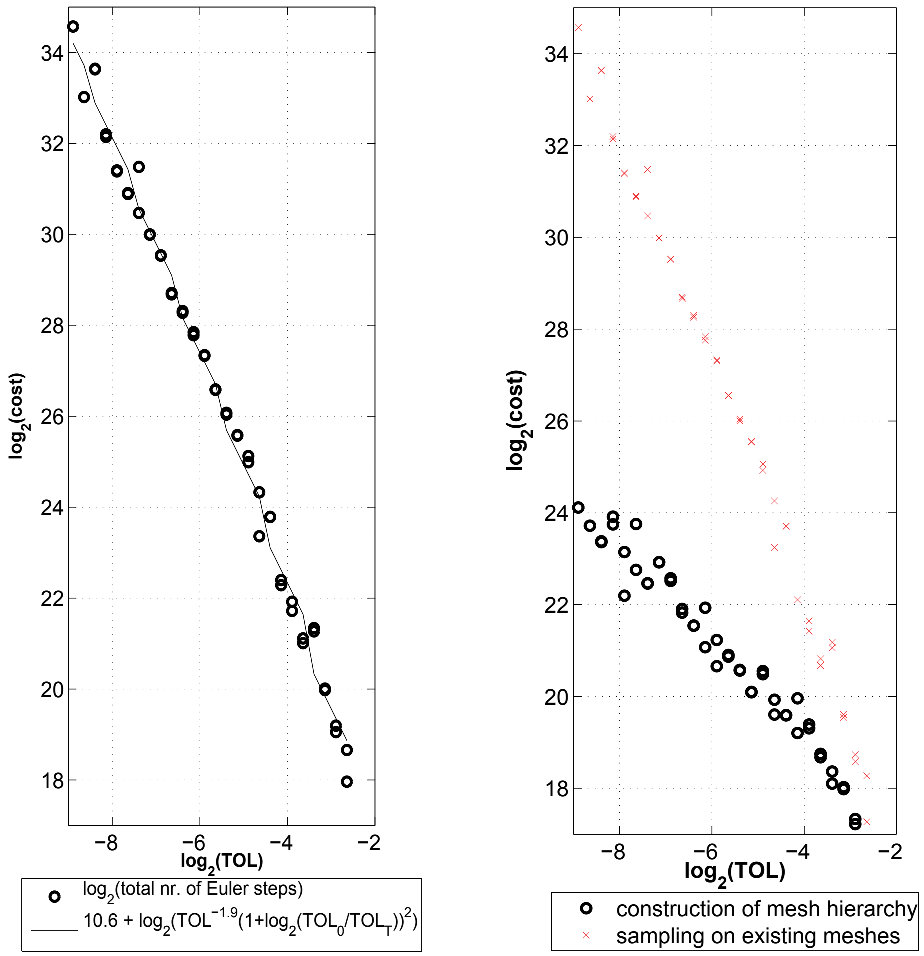

Figure 3. Experimental complexity when the algorithm in Section 2.1 is applied to the drift singularity problem in Section 3.2. To the left is shown the cost of both phases of the algorithm, and to the right the contribution from the generation of the mesh hierarchy and the subsequent sampling to reduce the statistical error; it is clear that the cost of the first phase is negligible compared to the second for small tolerances. The computational cost is measured as the total number of Euler time steps taken in all refinement iterations on all levels for all realizations. The graphs show three independent realizations of the underlying Wiener processes for each prescribed tolerance. A least squares fit of the model (3.2) gives $c_2 = 1.9$.

In this example the weak rate of convergence for the Euler–Maruyama method with uniform time steps is only $1/2$, so the total cost for a single level uniform time stepping algorithm is proportional to TOL^{-4}. The left part of Figure 4 shows that the single level version of the adaptive algorithm improves that complexity to approximately TOL^{-3}, while the multilevel version improves the complexity by nearly one order more. With the regularization (3.4) the observed order of strong convergence of the Euler–Maruyama method with uniform time steps is still $1/2$, so the complexity estimate in [11, Theorem 1] for uniform multilevel simulations applies, and we should get the ideal complexity $(\mathrm{TOL}^{-1}\log(\mathrm{TOL}^{-1}))^2$ for a mean square error of size TOL^2. The right part of Figure 4 shows that this is approximately true for the cost as a function of the maximal observed error over eleven independent realizations of the adaptive runs.

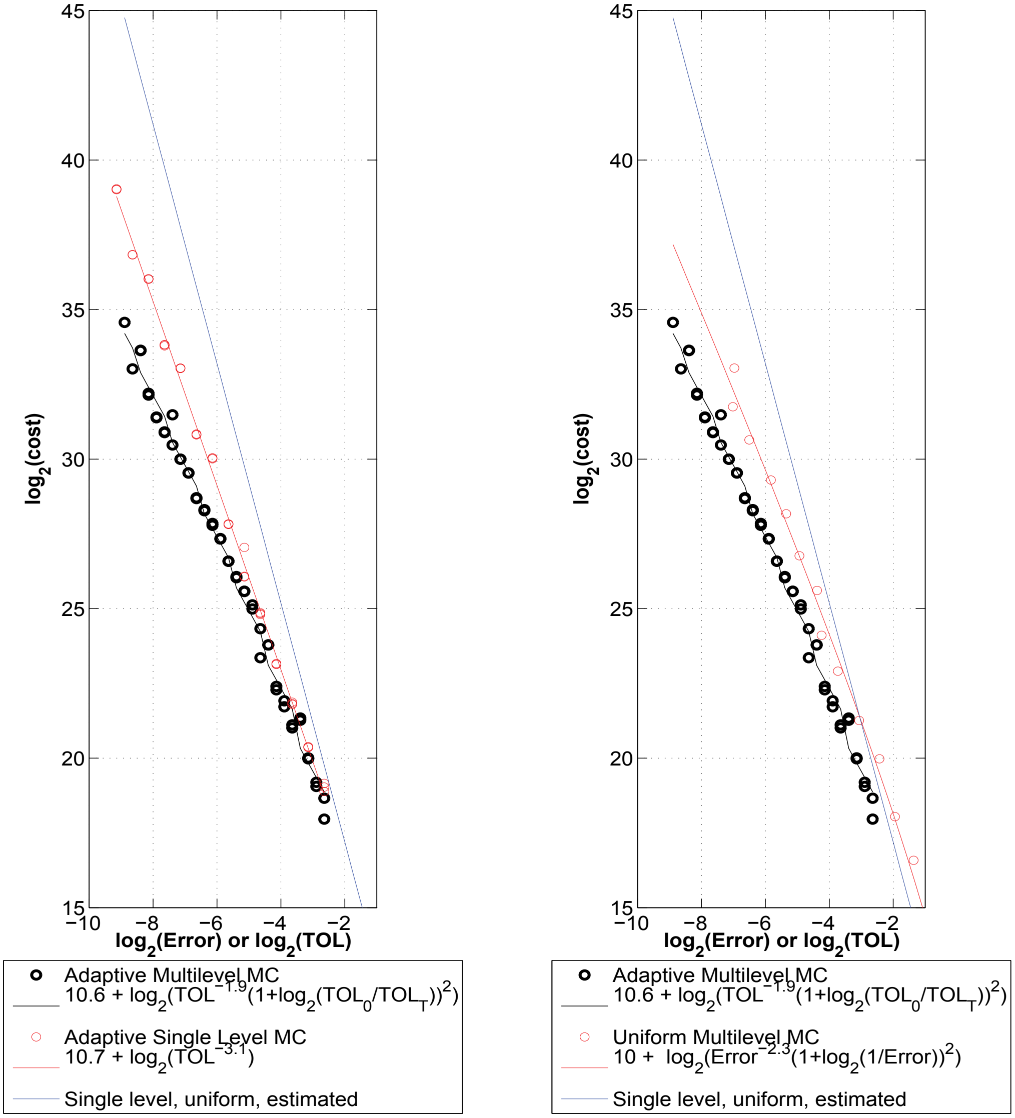

Figure 4. The computational cost of the path independent adaptive algorithm of Section 2.1, applied to the deterministic singularity Problem 3.2, is compared to several alternatives. Left: the multilevel version improves the computational complexity of the single level version of the same adaptive algorithm from approximately proportional to TOL^{-3} to approximately proportional to $\mathrm{TOL}^{-2}(1+\log(\mathrm{TOL}_{\mathrm{T},0}/\mathrm{TOL}_{\mathrm{T}}))^2$. The cost of a standard, uniform time step, Monte Carlo method would be proportional to TOL^{-4}; here the work was estimated from a Central Limit Theorem type confidence interval based on the time discretization errors and sample variances. Right: The cost of the uniform MLMC method is shown as a function of the maximal error, ϵ, over 11 realizations. The observed cost oscillates around a complexity curve that is possibly slightly worse than, but close to, $(\epsilon^{-1}\log(\epsilon^{-1}))^2$, which is expected since the observed strong order of convergence is still $1/2$. For the adaptive algorithm the cost is estimated by the total number of Euler steps taken on all levels in all stages of the adaptive refinement process.

Remark 3.3. In case the location, α, of the singularity in the drift is stochastic, the stochastic time stepping version of the adaptive algorithm in Section 2.2 is the appropriate choice. If we for example consider $\alpha \sim U(0,T)$, independent of the underlying Wiener process, then the stochastic adaptive multilevel Monte Carlo algorithm is applicable even without the a priori TOL-regularization of the drift in (3.4). In this case the uniform multilevel Monte Carlo algorithm cannot be applied without regularization of the drift, since the expected value that is computed by the discrete algorithm is not well defined due to the small probability events of the singularity being arbitrarily close to a grid point from below. In practice when computing with the uniform meshes we may fail to notice that the computation is unreliable since the failures are low probability events.

3.3 Stopped diffusion

Here we compute the solution to a more challenging problem that motivates the use of stochastic time steps that are adaptively refined for each sample path.

The additional difficulty of the problem is that we now wish to compute approximations of an expected value

$$\mathrm{E}[g(X(\tau),\tau)], \tag{3.6}$$

where $X(t)$ solves the SDE (1.1) as before, but where the function $g : D \times [0,T] \to \mathbb{R}$ is evaluated at the first exit time

$$\tau := \inf\{t > 0 : (X(t),t) \notin D \times (0,T)\}$$

from a given open domain $D \times (0,T) \subset \mathbb{R}^d \times (0,T)$. This kind of stopped (or killed) diffusion problems arises for example from barrier option pricing problems in mathematical finance and from boundary value problems in physics.

The main difficulty in the approximation of the stopped diffusion on the boundary ∂D is that a continuous sample path may exit the given domain D even though a discrete approximate solution does not cross the boundary of D. Due to this hitting of the boundary the order of weak convergence of the Euler–Maruyama method is reduced from 1 to $1/2$, in terms of the step size of uniform meshes; see [14]. The problem of simulating stopped diffusion has also been studied in, e.g., [3, 4, 24]. In this subsection we combine the adaptive multilevel algorithm of Section 2.2 with an error estimate derived in [8] that also takes into account the hitting error. This error estimate, and the adaptive algorithm, can be used also when D is multi-dimensional even if the boundary ∂D has corners for example.

The hitting error is accounted for by an extra contribution to the error density in (1.22); this contribution can be expressed in terms of exit probabilities for individual time steps, conditioned on the computed path at the beginning and the end of the time steps, and of the change in the goal function, g, when evaluated at a possible exit point within the time step instead of the actually computed exit $(\overline{X}(\bar{\tau}),\bar{\tau})$. The full expression of the resulting error indicators is given in [8, equation (50)]. Since the differential $\partial_i g(\overline{X}(T),T)$ in the discrete dual backward problem (1.16) does not exist if T is replaced by $\bar{\tau} < T$, this initial value must be alternatively defined; this can be done using difference quotients with restarted computed trajectories as described, both for the discrete dual and for its first and second variations, in [8, equations (20)–(25)]. Note that for this modified error density the proof in [26] of almost sure convergence to a limit density does not apply.

In addition to the modification of the error density a lower bound is introduced on the step size to avoid excessive refinements near the barrier,

$$\Delta t_n \geq \min\left\{\mathrm{TOL}_{\mathrm{T},\ell}{}^{1.5}, \frac{\mathrm{dist}_n \mathrm{dist}_{n+1}/b(\overline{X}(t_n;\omega),t_n)^2}{-3\log(\mathrm{TOL}_{\mathrm{T},\ell})}\right\}, \tag{3.7}$$

where dist_j denotes the distance from $\overline{X}(t_j;\omega)$ to the barrier.

Problem 3.4. For the numerical example we consider the stopped diffusion problem

$$\begin{aligned} dX(t) &= \frac{11}{36}X(t)dt + \frac{1}{6}X(t)dW(t) \quad \text{for } t \in [0,2] \text{ and } X(t) \in (-\infty,2), \\ X(0) &= 1.6. \end{aligned} \tag{3.8}$$

For $g(x,t) = x^3 e^{-t}$ with $x \in \mathbb{R}$, this problem has the exact solution $\mathrm{E}[g(X_\tau,\tau)] = u(X(0),0) = X(0)^3$, where the solution, u, of the Kolmogorov backward equation is $u(x,t) = x^3 e^{-t}$.

We chose an example in one space dimension for simplicity, although it is only in high dimension that Monte Carlo methods are more efficient than deterministic finite difference or finite element methods to solve stopped diffusion problems. The comparison here between the standard Monte Carlo and the multilevel Monte Carlo methods in the simple one-dimensional example indicates that the Multilevel Monte Carlo method will also be more efficient in high-dimensional stopped diffusion problems, where a Monte Carlo method is a good choice. In the case of a scalar SDE, where D is an interval on the real line, the strong order of convergence of the Euler–Maruyama scheme for barrier problems can be close to $1/2$. In fact, it is shown in [12] that $\mathrm{Var}(g(\overline{X}_\ell) - g(\overline{X}_{\ell-1})) = \mathcal{O}(\Delta t^{1-\delta})$, for any $\delta > 0$, using the Euler–Maruyama method with uniform step size Δt on a class of options including some barrier options. In this case [11, Theorem 3.1] tells us that, for any choice of $\delta > 0$, uniform MLMC simulations can be performed at a cost $\mathcal{O}(\mathrm{TOL}^{-2(1+\delta)})$, where the constant may depend on δ, for a mean square error of order TOL^2.

In the remainder of this section we present numerical results on the accuracy and cost of the adaptive multilevel algorithm of Section 2.2, applied to (3.8), with the error estimate modified for the barrier problem, and with the lower bound (3.7) on the step size. The algorithm was applied with a sequence of tolerances with three simulations for each tolerance using different initial states in the pseudo-random number generator. The observed errors are scattered below the corresponding tolerances in Figure 2, showing that the algorithm achieves the prescribed accuracy.

The experimental complexity is illustrated in Figure 5 and Table 2. A least squares fit of the model (3.2) using equal weights on all data points gives $c_2 = 2.0$ when the work is measured by the total number of sampled random variables; this is the measure of work that is estimated by (4.2) in Section 4. When all Euler steps in all refinement stages are included, the least squares fit gives $c_2 = 2.2$. However, the corresponding cost using the single level adaptive algorithm with just one data point per tolerance used grows faster than TOL^{-3} in this example; see Figure 6.

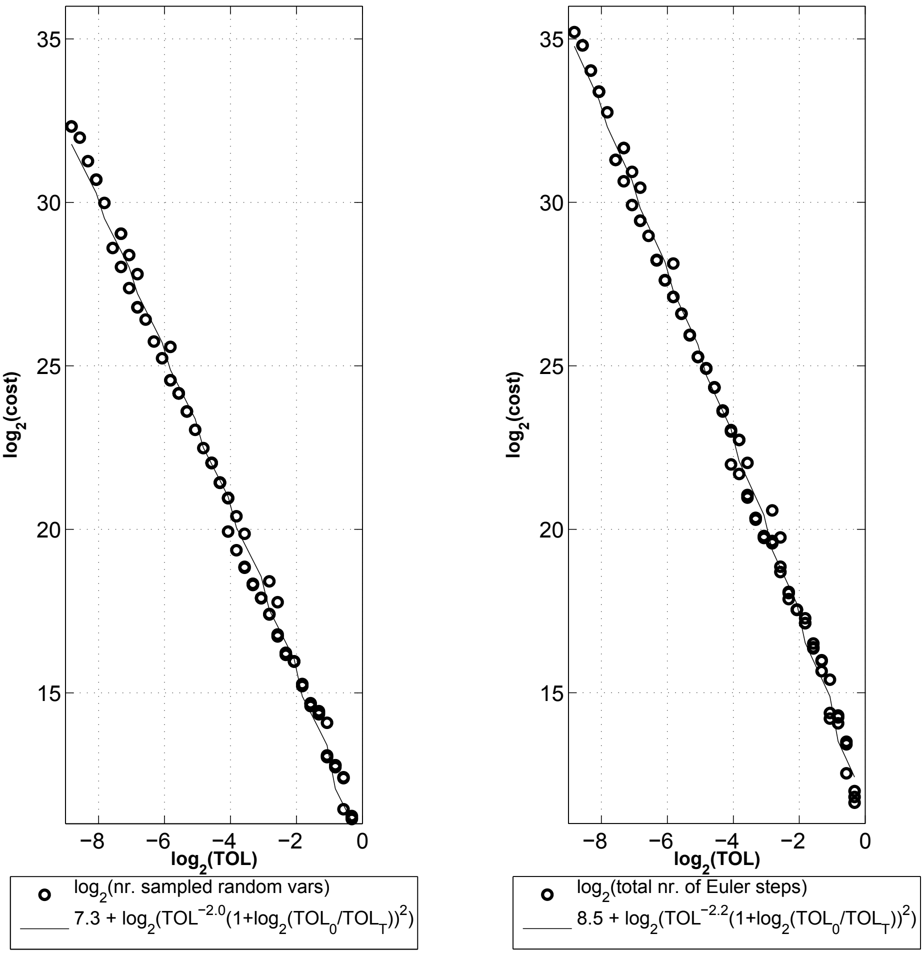

Figure 5. Experimental complexity for the barrier example in Section 3.3. The computational cost of the multilevel adaptive algorithm is shown for varying tolerances using three different initial states in the pseudo-random number algorithm. To the left is shown the work estimate based on the number of sampled random variables, which is the work measure closest to (4.2) used in Section 4; to the right is shown the estimate based on all Euler steps taken in all stages in the adaptive mesh refinement process. A least squares fit of the model (3.2) with equal weight on all observations results in $c_2 = 2.0$ and $c_2 = 2.2$ in the two cases.

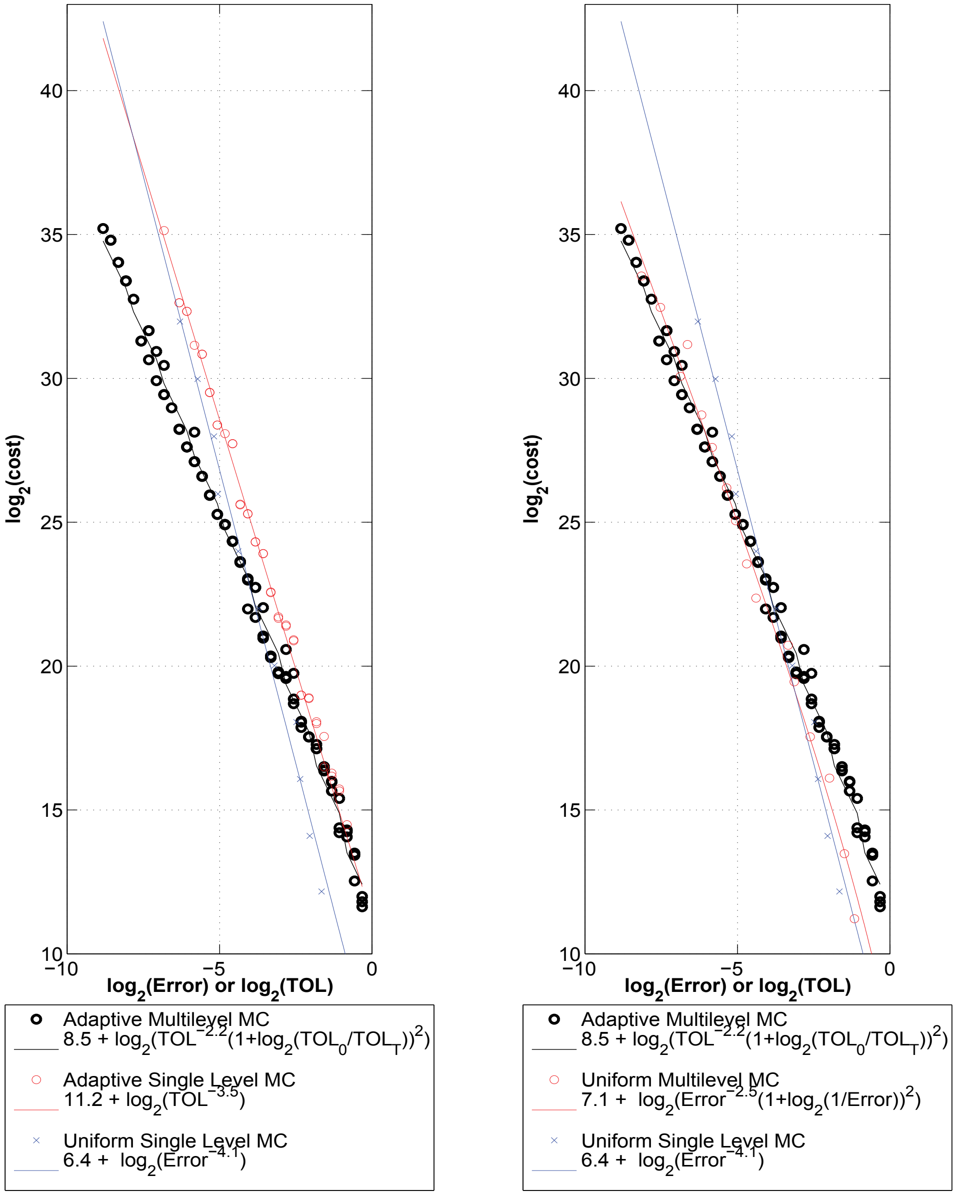

Figure 6. Left: The multilevel version of the path dependent adaptive algorithm of Section 2.2 applied to the barrier Problem 3.4 improves the computational complexity of the single level version of the same adaptive algorithm; a single level method based on uniform time steps has even worse complexity with the computational cost growing like ϵ^{-4}. Right: The cost of the uniform MLMC method is shown as a function of the maximal error, ϵ, over 16 realizations. The observed cost is close to that of adaptive multilevel Monte Carlo, which is expected since the observed observed strong order of convergence is $1/2$, but oscillates around a slightly worse fitted complexity $\epsilon^{-2.5}(1 + \log(\epsilon^{-1}))^2$. The cost is estimated by the total number of Euler steps taken on all levels in all stages of the adaptive refinement process.

In conclusion, the barrier problem (3.8) is not within the scope of Theorem 4.2 since almost sure convergence of the modified error density to a limit density has not been proven yet. Still, the observed convergence of the adaptive MLMC method applied to this problem agrees with the rate in Theorem 4.2. This shows an improved convergence compared to the single level version of the adaptive Monte Carlo algorithm where the cost grows approximately like TOL^{-3}, which in itself is a better order of weak convergence than the one obtained using a single level Monte Carlo method with constant time steps where the cost grows like TOL^{-4}.

4 Theoretical results

In this section we study the asymptotic accuracy and complexity of the stochastic time stepping adaptive MLMC algorithm introduced in Section 2.2. We recall that for a sought accuracy $\mathrm{TOL} > 0$, the goal of the adaptive MLMC algorithm is to construct a Monte Carlo approximation of $\mathrm{E}[g(X(T))]$ that with probability close to one fulfills

$$|\mathrm{E}[g(X(T))] - \mathcal{A}_{\mathcal{ML}}(g(\overline{X}(T)); M_0)| \leq \mathrm{TOL}.$$

Our main result on asymptotic accuracy for the adaptive MLMC algorithm, which is proved in Section 4.2, is

Theorem 4.1 (Multilevel accuracy). *Suppose the assumptions of Lemma 1.3 and* (4.4)–(4.6) *hold and that* $\mathrm{TOL_T} \leq \mathrm{TOL_S}$. *Then the adaptive MLMC algorithm with confidence parameter* $C_C > 0$ *and stochastic time steps* (2.19) *and* (2.20) *satisfies*

$$\liminf_{\mathrm{TOL}\downarrow 0} P(|\mathrm{E}[g(X(T))] - \mathcal{A}_{\mathcal{ML}}(g(\overline{X}(T)); M_0)| \leq \mathrm{TOL}) \geq \int_{-C_C}^{C_C} \frac{e^{-x^2/2}}{\sqrt{2\pi}} dx. \tag{4.1}$$

The motivation for introducing multiple levels in the MC algorithm is to reduce the computational complexity. To study the asymptotic complexity of the adaptive MLMC algorithm we define its work by

$$\mathrm{WORK(TOL)} = \sum_{\ell=0}^{L} \mathrm{E}[M_\ell]\mathrm{E}[N_\ell], \tag{4.2}$$

recalling that M_ℓ denotes the number of realization samples $g(\overline{X}_\ell(T;\omega))$ at level ℓ required to control the statistical error, and N_ℓ denotes the number of adaptive time steps required in the construction of a numerical realization $g(\overline{X}_\ell(T;\omega))$ to control the time discretization error at level ℓ. The function WORK(TOL) is an approximation of the average number of arithmetic operations required in the generation and sampling of $\{g(\overline{X}_\ell(T))\}_{\ell=0}^{L}$ to approximate $\mathrm{E}[g(X(T))]$ for the prescribed confidence C_C and accuracy TOL. The adaptive MLMC algorithm's *real* work, however, is a very complicated expression where products of expectations $\mathrm{E}[M_\ell]\mathrm{E}[N_\ell]$ should be replaced by expectations of products $\mathrm{E}[M_\ell N_\ell]$ and the full cost of the refinement process for each realization should be included. To simplify the analysis here, we have decided to study the asymptotics of the work defined in (4.2), instead of the algorithm's *real* work. Our main complexity theorem follows, but first we recall from [25] that the error density ρ has an almost sure asymptotic limit which we here denote by $\hat{\rho}$, i.e., $\rho \to \hat{\rho}$ as $\mathrm{TOL_T} \downarrow 0$.

Theorem 4.2 (Multilevel computational complexity). *Suppose the assumptions of Lemma 1.3 and* (4.4)–(4.6) *hold and that the lower bound for the error density is on the form* $\rho_{\mathrm{low}}(\mathrm{TOL_T}) = \mathrm{TOL_T^{\bar{\gamma}}}$, *cf.* (1.23), *with* $\bar{\gamma} \to 0$ *and* $L\bar{\gamma} \to \infty$ *as* $\mathrm{TOL} \downarrow 0$. *Then the work for the adaptive MLMC algorithm defined in* (4.2) *fulfills the following bound:*

$$\limsup_{\mathrm{TOL}\downarrow 0} \frac{\mathrm{WORK(TOL)TOL}^2\, \bar{\gamma}}{L\, 2^{\bar{\gamma}L}} \leq \frac{8\, C_C^2\, C_G}{\log(2)\, \mathrm{TOL_{T,Max}}\, C_R} \left(\mathrm{E}\left[\int_0^T \sqrt{|\hat{\rho}(\tau)|} d\tau \right] \right)^2. \tag{4.3}$$

Here, the number of levels $L = \mathcal{O}(\log(\mathrm{TOL}^{-1}))$, C_C *is the confidence parameter,* C_R *and* C_S *are refinement parameters described by* (2.19) *and* (2.20), C_G *is the constant in the second moment bound* (4.39), *where* $\mathrm{TOL_{T,Max}}$ *is the upper bound of the time discretization tolerance at level* $\ell = 0$, *and* $\bar{\gamma}$ *is the lower bound error density exponent;* $\rho_{\mathrm{low}}(\mathrm{TOL_T}) = \mathrm{TOL_T^{\bar{\gamma}}}$, *cf.* (1.23).

Remark 4.3 (Complexity example). Theorem 4.2 implies that if the exponent of the lower error density ρ_{low} is given by $\bar{\gamma}(\mathrm{TOL}) = \log_2(\log_2(L))/L$, then the following complexity bound, notably close to the standard complexity of the uniform time stepping MLMC method, is achieved:

$$\limsup_{\mathrm{TOL}\downarrow 0} \frac{\mathrm{WORK(TOL)\ TOL}^2\ \log_2(\log_2(L))}{L^2\ \log_2(L)} \leq \frac{8\, C_C^2\, C_G}{\log(2)\, \mathrm{TOL_{T,Max}}\, C_R} \left(\mathrm{E}\left[\int_0^T \sqrt{|\hat{\rho}(\tau)|} d\tau \right] \right)^2.$$

To present the proofs of Theorems 4.1 and 4.2 in a gentle fashion, we first prove analogous results for the adaptive SLMC algorithm in Section 4.1. With single level proofs fresh in mind, we move on to the more daunting task of proving Theorems 4.1 and 4.2 in Section 4.2. As already noted, we restrict ourselves here to proving Theorems 4.1 and 4.2 for the stochastic time stepping setting. Stochastic time stepping is however the most general setting, so one can easily prove corresponding results for the deterministic time stepping setting as well.

In addition to Lemma 1.3, the analysis in this section will be derived relying on the following three assumptions.

- Strong approximation convergence rate:[9] For $p = 2$ and 4, we have that

$$\begin{gathered} \mathrm{E}[|g(X(T)) - g(\overline{X}(T))|^p] = \mathcal{O}\bigg(\frac{\mathrm{TOL_T}}{\rho_{\mathrm{low}}(\mathrm{TOL_T})}\bigg)^{p/2}, \\ \mathrm{E}[|g(\overline{X}(T))|^p] = \mathcal{O}(1). \end{gathered} \tag{4.4}$$

- That adaptivity is relevant for the weak approximation problem considered in the sense that the asymptotic error density is nontrivial and we have that

$$\mathrm{E}\bigg[\int_0^T \sqrt{|\hat{\rho}(\tau)|}d\tau\bigg] > 0. \tag{4.5}$$

- For all $s, t \in [0, T]$ the sensitivity of the error density to values of the Wiener process can be bounded as follows:

$$|\partial_{W(t)}\rho(s,\omega)| \le D\rho_{\mathrm{up}}(\mathrm{TOL_T}), \tag{4.6}$$

  for some positive function $D\rho_{\mathrm{up}}$ such that $D\rho_{\mathrm{up}}(\mathrm{TOL_T}) \to +\infty$ as $\mathrm{TOL_T} \downarrow 0$.

4.1 Single level results

The adaptive SLMC algorithm considered in this subsection was first described and analyzed in [29]. The purpose of giving a new analysis here is to construct proofs for the asymptotic accuracy and complexity of the adaptive SLMC algorithm that subsequently are easily extended to proofs for the adaptive MLMC algorithm. In this section's first lemma we show that the adaptive refinement Algorithm 9 stops after a finite number of iterations. This property allows us to later bound the amount of computational work in the single level adaptive algorithm. It also has another important implication: the imposed lower bound on the error density, $\rho_{\mathrm{low}}(\mathrm{TOL_T})$ in (1.10), ensures that the maximum mesh size of the mesh generated by Algorithm 9, $\Delta t_{\mathrm{sup}}(\mathrm{TOL_T})$ introduced in Lemma 1.4, tends to zero as $\mathrm{TOL_T}$ tends to zero. This in turn implies the almost sure convergence of the error density, which is crucial in the proofs of the main results of this section. A similar result also holds for the multilevel case but will not be stated here for the sake of brevity.

Lemma 4.4 (Stopping). *Suppose the adaptive Algorithm 9 applies the mesh refinement strategy* (2.19)–(2.20) *on a set of realizations having the same uniform initial mesh with step size Δt_0. Assume further that the initial estimated average number of time steps, $\overline{\mathcal{N}}_{\mathrm{in}}$, satisfies*

$$\overline{\mathcal{N}}_{\mathrm{in}} < N_{\mathrm{up}} := \frac{T^2\rho_{\mathrm{up}}(\mathrm{TOL_T})}{C_R\,\mathrm{TOL_T}}, \tag{4.7}$$

and that a prescribed accuracy parameter $\mathrm{TOL_T} > 0$ *is given. Then the adaptive refinement in Algorithm 9 stops after a finite number of iterations.*

[9] The work [29] gives conditions under which (4.4) is fulfilled.

Proof. First recall that by (1.10), the error density is bounded from above by $\rho \le \rho_{\mathrm{up}}(\mathrm{TOL_T})$. So given an initial uniform mesh with size Δt_0 and containing N_0 intervals, the uniform mesh size

$$\tilde{\Delta t}(\mathrm{TOL_T}) = \frac{\Delta t_0}{\max\{1, 2^k\}} \quad \text{with} \quad k = \left\lceil \log_2\left(\frac{\rho_{\mathrm{up}}(\mathrm{TOL_T})\, T\, \Delta t_0}{C_{\mathrm{R}}\, \mathrm{TOL_T}} \right) \right\rceil \tag{4.8}$$

satisfies both the stopping condition (2.19) and the non-refinement condition (2.20) for Algorithm 9. When a time step reaches the mesh size $\tilde{\Delta t}(\mathrm{TOL_T})$, it will consequently not be further refined. The number of possible refinements from the initial mesh size Δt_0 to a uniform mesh with step size $\tilde{\Delta t}(\mathrm{TOL_T})$ is bounded by the finite number $N_0 \max\{1, 2^k\}$. The proof is concluded by observing that Algorithm 9 either stops or makes at least one refinement during each iteration. □

The work [26] also proves a similar stopping result, cf. [26, Theorem 3.2], based on the assumption that the initial mesh is sufficiently refined so that the error density does not vary too much between refinement levels. Then, when the single level adaptive algorithm stops, one can prove asymptotic accuracy and efficiency estimates for the resulting weak approximation. In contrast, here we make essentially no assumption on the initial mesh size Δt_0: although the quality of the resulting approximation for the lower levels of the multilevel estimator may be poor, they have no influence in the bias of the multilevel approximation, which is only determined by the finest level, L. Since $L \to \infty$ as $\mathrm{TOL} \downarrow 0$, we can still prove asymptotic accuracy and efficiency estimates. Finally, we observe that assumption (4.7) is fulfilled in all practical cases since one should start the adaptive algorithm with $\overline{\mathbb{N}}_{\mathrm{in}}$ of the order of $\mathrm{TOL_T^{-1}}$, which is much smaller than N_{up}.

The following proofs are inspired by the treatment by Chow and Robbins [6] on the accuracy and complexity of sequential stopping rules for sampling i.i.d. random variables.

We denote the SLMC sample average estimator of $\mathrm{E}[g(X(T))]$ by

$$\mathcal{A}(g(\overline{X}(T)); M) = \sum_{i=1}^{M} \frac{g(\overline{X}(T;\omega_i))}{M},$$

where the realizations of $\overline{X}(T)$ are generated on adaptive meshes and fulfill the weak error bound

$$|\mathrm{E}[g(\overline{X}(T)) - g(X(T))]| \lesssim \mathrm{TOL_T}.$$

Here the total tolerance TOL is split into a time discretization error tolerance and a statistical error tolerance, $\mathrm{TOL} = C_S \mathrm{TOL_T} + \mathrm{TOL_S}$. Remark 4.10 discusses the optimal splitting of TOL further. Let $2^{\mathbb{N}}$ denote the set $\{2^n \mid n \in \mathbb{N}\}$. For the SLMC estimator, the number of samples used in the sample average estimator to control the statistical error $|\mathcal{A}(g(\overline{X}(T)); M) - \mathrm{E}[g(\overline{X}(T))]| \le \mathrm{TOL_S}$ is a stochastic process $M : \mathbb{R}_+ \to 2^{\mathbb{N}}$ defined by

$$M(\mathrm{TOL_S}) := \text{the smallest } k \in 2^{\mathbb{N}+\lceil \log_2(\mathrm{TOL}^{-1})\rceil} \text{ such that } \mathcal{V}(g(\overline{X}(T)); k) < \frac{k\mathrm{TOL_S}^2}{C_C^2}, \tag{4.9}$$

where the sample variance is defined by

$$\mathcal{V}(g(\overline{X}(T)); k) = \sum_{i=1}^{k} \frac{(g(\overline{X}(T;\omega_i)) - \mathcal{A}(g(\overline{X}(T)); k))^2}{k-1}. \tag{4.10}$$

Restricting the initial value of M to the set $2^{\mathbb{N}+\lceil \log_2(\mathrm{TOL}^{-1})\rceil}$ implies that $\lim_{\mathrm{TOL}\downarrow 0} M = \infty$. The asymptotic behavior of M as $\mathrm{TOL} \downarrow 0$ is crucial in our proofs of the asymptotic accuracy and complexity. When proving the asymptotically accuracy result of Proposition 4.6, M should increase sufficiently fast to obtain the sought confidence. For the complexity result of Proposition 4.9, it is on the other hand useful to bound M from above and ensure that it does not grow too fast.

Lemma 4.5. *Suppose the assumptions* (4.4)–(4.6) *hold. Then*

$$\liminf_{\mathrm{TOL}\downarrow 0} \frac{M\mathrm{TOL}_S^2}{\mathrm{Var}(g(\overline{X}(T)))C_C^2} = 1 \quad a.s. \quad and \quad \limsup_{\mathrm{TOL}\downarrow 0} \frac{M\mathrm{TOL}_S^2}{\mathrm{Var}(g(\overline{X}(T)))C_C^2} = 2 \quad a.s. \tag{4.11}$$

Proof. The strong convergence (4.4) for $p = 2$, gives $\lim_{\mathrm{TOL}\downarrow 0} \mathrm{Var}(g(\overline{X}(T))) = \mathrm{Var}(g(X(T)))$, which in particular means that there exists a constant $\widetilde{\mathrm{TOL}} > 0$ such that

$$\frac{\mathrm{Var}(g(X(T)))}{2} < \mathrm{Var}(g(\overline{X}(T))) < 2\mathrm{Var}(g(X(T))) \quad \text{for all TOL} \in (0, \widetilde{\mathrm{TOL}}]. \tag{4.12}$$

The Strong Law of Large Numbers then implies that

$$\lim_{k\to\infty} \mathcal{V}(g(\overline{X}(T)); k) = \mathrm{Var}(g(\overline{X}(T))) \quad \text{a.s.} \quad \text{for all TOL} \in (0, \widetilde{\mathrm{TOL}}]. \tag{4.13}$$

In order to prove results (4.11), introduce the sequence of stochastic processes $y_k : \mathbb{R}_+ \to \mathbb{R}_+$ sub-indexed by $k \in 2^{\mathbb{N}+\lceil \log_2(\mathrm{TOL}^{-1})\rceil}$ and defined by

$$y_k(\mathrm{TOL}) = \frac{\mathcal{V}(g(\overline{X}(T)); k)}{\mathrm{Var}(g(\overline{X}(T)))}. \tag{4.14}$$

Using y_k, definition (4.9) of $M(\mathrm{TOL}_S)$ is equivalent to

$$M(\mathrm{TOL}_S) := \text{the smallest } k \in 2^{\mathbb{N}+\lceil \log_2(\mathrm{TOL}^{-1})\rceil} \text{ such that } y_k(\mathrm{TOL}_S) < \frac{k\mathrm{TOL}_S^2}{\mathrm{Var}(g(\overline{X}(T)))C_C^2}.$$

This stopping condition gives rise to the bounds

$$y_M(\mathrm{TOL}_S) < \frac{M\mathrm{TOL}_S^2}{\mathrm{Var}(g(\overline{X}(T)))C_C^2} \le 2y_{M/2}(\mathrm{TOL}_S). \tag{4.15}$$

Combining (4.13) with definition (4.9), which ensures that $M(\mathrm{TOL}_S) \to \infty$ as TOL $\downarrow 0$, we conclude that

$$\lim_{\mathrm{TOL}\downarrow 0} \mathcal{V}(g(\overline{X}(T)); M(\mathrm{TOL}_S)) = \mathrm{Var}(g(X(T))) > 0 \quad \text{a.s.},$$

which implies that also $\lim_{\mathrm{TOL}\downarrow 0} y_M(\mathrm{TOL}_S) = 1$ a.s. Statement (4.11) then follows by taking limits in (4.15). □

Having obtained asymptotic bounds for M, we are ready to prove the main accuracy result for the adaptive SLMC algorithm.

Proposition 4.6 (Single level accuracy). *Suppose the assumptions of Lemma 1.3 and* (4.4)–(4.6) *hold and that* $\mathrm{TOL_T} \le \mathrm{TOL_S}$. *Then, the adaptive SLMC algorithm with confidence refinement parameter* $C_C > 0$, *and time steps* (2.19) *and* (2.20), *satisfies*

$$\liminf_{\mathrm{TOL}\downarrow 0} P(|\mathrm{E}[g(X(T))] - \mathcal{A}(g(\overline{X}(T)); M)| \le \mathrm{TOL}) \ge \int_{-C_C}^{C_C} \frac{e^{-x^2/2}}{\sqrt{2\pi}}\,dx. \tag{4.16}$$

Proof. For a given $\delta > 0$, we first bound the probability in (4.16) from below as follows:

$$\begin{aligned}
&\liminf_{\mathrm{TOL}\downarrow 0} P\big(|\mathrm{E}[g(X(T))] - \mathcal{A}(g(\overline{X}(T)); M)| \le \mathrm{TOL}\big)\\
&\ge \liminf_{\mathrm{TOL}\downarrow 0} P\big(|\mathrm{E}[g(X(T)) - g(\overline{X}(T))]| + |\mathrm{E}[g(\overline{X}(T))] - \mathcal{A}(g(\overline{X}(T)); M)| \le C_S\mathrm{TOL}_T + \mathrm{TOL}_S\big)\\
&\ge \liminf_{\mathrm{TOL}\downarrow 0} P\big(|\mathrm{E}[g(X(T)) - g(\overline{X}(T))]| \le (C_S+\delta)\mathrm{TOL}_T \text{ and } |\mathrm{E}[g(\overline{X}(T))] - \mathcal{A}(g(\overline{X}(T)); M)| \le (1-\delta)\mathrm{TOL}_S\big)\\
&= \liminf_{\mathrm{TOL}\downarrow 0} P\big(|\mathrm{E}[g(X(T)) - g(\overline{X}(T))]| \le (C_S+\delta)\mathrm{TOL}_T\big) \times P\big(|\mathrm{E}[g(\overline{X}(T))] - \mathcal{A}(g(\overline{X}(T)); M)| \le (1-\delta)\mathrm{TOL}_S\big).
\end{aligned} \tag{4.17}$$

The proof is continued by analyzing the two product terms of the last line of the inequality above separately.

The time discretization error. The assumption that Lemma 1.3 and (4.4) hold implies that

$$\limsup_{\mathrm{TOL}\downarrow 0} \frac{|\mathrm{E}[g(X(T)) - g(\overline{X}(T))]|}{\mathrm{TOL}_T} \le C_S,$$

cf. the proof of [25, Theorem 3.4]. Thereby,

$$\liminf_{\mathrm{TOL}\downarrow 0} P(|\mathrm{E}[g(X(T)) - g(\overline{X}(T))]| \le (C_S+\delta)\mathrm{TOL}_T) = 1.$$

The statistical error. For the above introduced $\delta > 0+$, define the family of sets

$$\Omega_\delta(\mathrm{TOL}_S) = \Big\{k \in 2^{\mathbb{N}+\lceil \log_2(\mathrm{TOL}^{-1})\rceil} \;\Big|\; 1-\delta < \frac{k\mathrm{TOL}_S^2}{\mathrm{Var}(g(\overline{X}(T)))C_C^2} \le 2+\delta\Big\}. \tag{4.18}$$

By the convergence (4.11), we conclude that

$$\lim_{\mathrm{TOL}\downarrow 0} P(M \in \Omega_\delta) = 1.$$

Recall that for the adaptive SLMC algorithm, the number of samples M is determined in the step prior to generating the output $\mathcal{A}(g(\overline{X}(T)); M)$, so that M is independent from $\mathcal{A}(g(\overline{X}(T)); M)$. Using this independence property, Fatou's lemma, and Lindeberg–Feller's version of the Central Limit Theorem, cf. Theorem A.1, yields that

$$\begin{aligned}
&\liminf_{\mathrm{TOL}\downarrow 0} P(|\mathrm{E}[g(\overline{X}(T))] - \mathcal{A}(g(\overline{X}(T)); M)| \le (1-\delta)\mathrm{TOL}_S)\\
&\quad= \liminf_{\mathrm{TOL}\downarrow 0} \sum_{k\in 2^{\mathbb{N}+\lceil \log_2(\mathrm{TOL}^{-1})\rceil}} P(|\mathrm{E}[g(\overline{X}(T))] - \mathcal{A}(g(\overline{X}(T)); k)| \le (1-\delta)\mathrm{TOL}_S)P(M=k)\\
&\quad\ge \liminf_{\mathrm{TOL}\downarrow 0} \sum_{k\in\Omega_\delta} P(|\mathrm{E}[g(\overline{X}(T))] - \mathcal{A}(g(\overline{X}(T)); k)| \le (1-\delta)\mathrm{TOL}_S)P(M=k)\\
&\qquad\quad + \sum_{k\in 2^{\mathbb{N}+\lceil \log_2(\mathrm{TOL}^{-1})\rceil}\setminus\Omega_\delta} \liminf_{\mathrm{TOL}\downarrow 0} P(|\mathrm{E}[g(\overline{X}(T))] - \mathcal{A}(g(\overline{X}(T)); k)| \le (1-\delta)\mathrm{TOL}_S)P(M=k)\\
&\quad\ge \liminf_{\mathrm{TOL}\downarrow 0} \sum_{k\in\Omega_\delta} P\bigg(\sqrt{k}\frac{|\mathrm{E}[g(\overline{X}(T))] - \mathcal{A}(g(\overline{X}(T)); k)|}{\sqrt{\mathrm{Var}(g(\overline{X}(T)))}} \le (1-\delta)^{3/2}C_C\bigg)P(M=k)\\
&\quad\ge \int_{-(1-\delta)^{3/2}C_C}^{(1-\delta)^{3/2}C_C} \frac{e^{-x^2/2}}{\sqrt{2\pi}}\,dx.
\end{aligned} \tag{4.19}$$

The proof is finished by noting that the argument leading to inequality (4.19) is valid for all $\delta > 0$. □

We conclude this subsection with a complexity analysis of the adaptive SLMC algorithm. Similar to the definition of the work for the MLMC algorithm given in (4.2), we define the SLMC work by

$$\mathrm{WORK}(\mathrm{TOL}) = \mathrm{E}[M]\mathrm{E}[N], \tag{4.20}$$

where we recall that M denotes the number of samples of $g(\overline{X}(T))$ required to control the statistical error and N denotes the number of adaptive time steps required in the construction of a numerical realization $g(\overline{X}(T;\omega))$ to control the time discretization error $|\mathrm{E}[g(\overline{X}(T)) - g(X(T))]| \le \mathrm{TOL}_\mathrm{T}$. We start by bounding $\mathrm{E}[M]$.

Lemma 4.7. *Suppose the assumptions* (4.4)–(4.6) *hold. Then the expected value of the number of samples used in the approximation of* $\mathrm{E}[g(X(T))]$ *is bounded by*

$$\limsup_{\mathrm{TOL}\downarrow 0} \frac{\mathrm{E}[M]\mathrm{TOL}_S^2}{\mathrm{Var}(g(\overline{X}(T)))C_C^2} \le 2. \tag{4.21}$$

Proof. For a given $\delta > 0$, define the deterministic function

$$\widetilde{M}(\mathrm{TOL}_S) = \min\Big\{k \in 2^{\mathbb{N}+\lceil \log_2(\mathrm{TOL}^{-1})\rceil} \;\Big|\; \frac{k\mathrm{TOL}_S^2}{\mathrm{Var}(g(\overline{X}(T)))C_C^2} > 1+\delta\Big\}.$$

Assuming TOL is sufficiently small so that (4.12) holds, the relation (4.15), the fourth moment bound (4.4) and k-Statistics bounds on the variance of the sample variance, cf. [22], yield

$$\begin{aligned}
P(M = 2\widetilde{M}) &\le P\bigg(\frac{\mathcal{V}(g(\overline{X}(T)); \widetilde{M})}{\mathrm{Var}(g(\overline{X}(T)))} > \widetilde{M}\frac{\mathrm{TOL}_S^2}{\mathrm{Var}(g(\overline{X}(T)))C_C^2}\bigg) \le P\bigg(\frac{\mathcal{V}(g(\overline{X}(T)); \widetilde{M})}{\mathrm{Var}(g(\overline{X}(T)))} > 1+\delta\bigg)\\
&\le P(|\mathcal{V}(g(\overline{X}(T)); \widetilde{M}) - \mathrm{Var}(g(\overline{X}(T)))| > \delta\mathrm{Var}(g(\overline{X}(T))))\\
&\le 2\mathrm{E}\bigg[\frac{|\mathcal{V}(g(\overline{X}(T)); \widetilde{M}) - \mathrm{Var}(g(\overline{X}(T)))|^2}{\delta^2\mathrm{Var}(g(\overline{X}(T)))^2}\bigg] < \frac{C}{\delta^2\widetilde{M}}.
\end{aligned}$$

Furthermore, for $\ell = 1, 2, \dots$ we get that

$$\begin{aligned} P(M = 2^{\ell+1}\widetilde{M}) &\le P(|\mathcal{V}(g(\overline{X}(T)); 2^{\ell}\widetilde{M}) - \mathrm{Var}(g(\overline{X}(T)))| > 2^{\ell-1}\mathrm{Var}(g(\overline{X}(T)))) \\ &\le 2\mathrm{E}\Bigg[\frac{|\mathcal{V}(g(\overline{X}(T)); 2^{\ell}\widetilde{M}) - \mathrm{Var}(g(\overline{X}(T)))|^2}{2^{2(\ell-1)}\mathrm{Var}(g(\overline{X}(T)))^2}\Bigg] < \frac{C}{2^{2\ell}\widetilde{M}}. \end{aligned}$$

Consequently,

$$\begin{aligned} \frac{\mathrm{E}[M]\mathrm{TOL}_{\mathrm{S}}^2}{\mathrm{Var}(g(\overline{X}(T)))C_C^2} &\le \frac{[P(M \le \widetilde{M}) + \sum_{\ell=1}^{\infty} 2^{\ell} P(M = 2^{\ell}\widetilde{M})]\widetilde{M}\mathrm{TOL}_{\mathrm{S}}^2}{\mathrm{Var}(g(\overline{X}(T)))C_C^2} \\ &\le 2(1+\delta)\Bigg[P(M \le \widetilde{M}) + P(M = 2\widetilde{M}) + \sum_{\ell=1}^{\infty} 2^{\ell+1} P(M = 2^{\ell+1}\widetilde{M})\Bigg] \qquad (4.22) \\ &\le 2(1+\delta)\Bigg[P(M \le \widetilde{M}) + \frac{C}{\delta^2\widetilde{M}} + \frac{C}{\widetilde{M}}\sum_{\ell=1}^{\infty} 2^{-\ell}\Bigg]. \end{aligned}$$

By taking limits in the above inequality, we obtain

$$\limsup_{\mathrm{TOL}\downarrow 0} \frac{\mathrm{E}[M]\mathrm{TOL}_S^2}{\mathrm{Var}(g(\overline{X}(T)))C_C^2} \le 2(1+\delta).$$

Finally, noting that this result holds for any $\delta > 0$, the proof is finished. □

For an asymptotic bound on $\mathrm{E}[N]$, we recall [25, Theorem 3.5]. The bound given in this theorem is derived by studying the asymptotic form of the error indicators obtained by the stopping condition (2.19). The theorem further shows that up to a multiplicative constant, the mesh refinement scheme (2.19)–(2.20) yields stochastic meshes which are optimal in mean sense. The theorem is here stated as a lemma.

Lemma 4.8 (Single level asymptotic average number of time steps). *Suppose the assumptions of Lemma 1.3 and* (4.4)–(4.6) *hold. Then the final number of adaptive steps generated by the algorithm* (2.19) *and* (2.20) *satisfies asymptotically*

$$\limsup_{\mathrm{TOL}\downarrow 0} \mathrm{TOL}_T\, \mathrm{E}[N] \le \frac{4}{C_R}\Bigg(\mathrm{E}\Bigg[\int_0^T \sqrt{|\hat{\rho}(t)|}dt\Bigg]\Bigg)^2. \qquad (4.23)$$

The product of the asymptotic upper bounds for $\mathrm{E}[M]$ and $\mathrm{E}[N]$ and an optimization of the choice of TOL_T and TOL_S gives the following upper bound on the computational complexity for the adaptive SLMC algorithm.

Proposition 4.9 (SLMC computational complexity). *Suppose the assumptions of Lemma 1.3 and* (4.4)–(4.6) *hold. Then the work for the adaptive SLMC algorithm satisfies*

$$\limsup_{\mathrm{TOL}\downarrow 0} \mathrm{WORK}(\mathrm{TOL})\mathrm{TOL}^3 \le \frac{2\cdot 3^3\mathrm{Var}(g(X(T)))C_C^2 C_S}{C_R}\Bigg(\mathrm{E}\Bigg[\int_0^T \sqrt{|\hat{\rho}(t)|}dt\Bigg]\Bigg)^2, \qquad (4.24)$$

where C_C is the confidence parameter and C_R and C_S are refinement parameters described by (2.19) *and* (2.20).

Proof. Lemma 4.7 and 4.8 straightforwardly yield the upper bound

$$\limsup_{\mathrm{TOL}\downarrow 0} \mathrm{WORK}(\mathrm{TOL})\ \mathrm{TOL}_{\mathrm{S}}^2\mathrm{TOL}_{\mathrm{T}} \le \frac{2^3\mathrm{Var}(g(X(T)))C_C^2}{C_R}\Bigg(\mathrm{E}\Bigg[\int_0^T \sqrt{|\hat{\rho}(t)|}dt\Bigg]\Bigg)^2.$$

So $\mathrm{WORK}(\mathrm{TOL}) = \mathcal{O}(\mathrm{TOL}_{\mathrm{S}}^{-2}\mathrm{TOL}_{\mathrm{T}}^{-1})$. Minimizing $\mathrm{TOL}_{\mathrm{S}}^{-2}\mathrm{TOL}_{\mathrm{T}}^{-1}$ subject to the restriction $C_S\mathrm{TOL}_{\mathrm{T}} + \mathrm{TOL}_{\mathrm{S}} = \mathrm{TOL}$ yields

$$\mathrm{TOL}_{\mathrm{T}} = \frac{\mathrm{TOL}}{3C_S} \quad \text{and} \quad \mathrm{TOL}_{\mathrm{S}} = \frac{2\mathrm{TOL}}{3}.$$

These values for $\mathrm{TOL}_{\mathrm{T}}$ and $\mathrm{TOL}_{\mathrm{S}}$ lead to the upper bound (4.24). □

Remark 4.10. The optimal choices of $\mathrm{TOL_T}$ and $\mathrm{TOL_S}$ for minimizing $\mathrm{WORK(TOL)}$ are derived in the proof of Proposition 4.9 to be

$$\mathrm{TOL_T} = \frac{\mathrm{TOL}}{3C_S} \quad \text{and} \quad \mathrm{TOL_S} = \frac{2\mathrm{TOL}}{3}.$$

4.2 Multilevel results

We recall from the description of the adaptive MLMC algorithm in Section 2.2 that given an accuracy $\mathrm{TOL} = C_S\mathrm{TOL_T} + \mathrm{TOL_S}$, the adaptive MLMC algorithm generates realizations $g(\overline{X}_\ell(T))$ fulfilling the weak error bounds $|\mathrm{E}[g(\overline{X}_\ell(T)) - g(X(T))]| \lesssim \mathrm{TOL}_{\mathrm{T},\ell}$ on the levels $\ell = 0, 1, \dots, L$. The time discretization tolerance levels are given by $\mathrm{TOL}_{\mathrm{T},\ell} = 2^\ell \mathrm{TOL_T}$, and the number of levels is set by $L = \lfloor \log_2(\mathrm{TOL}_{\mathrm{T,Max}}/\mathrm{TOL_T})\rfloor$, where $\mathrm{TOL}_{\mathrm{T,Max}}$ is a predetermined max time discretization tolerance value, cf. (2.3). The multilevel sample average estimator of $\mathrm{E}[g(X(T))]$ is denoted by

$$\mathcal{A}_{\mathcal{ML}}(g(\overline{X}(T)); M_0) = \sum_{i=1}^{M_0} \frac{g(\overline{X}_0(T;\omega_{0,i}))}{M_0} + \sum_{\ell=1}^{L}\sum_{i=1}^{M_\ell} \frac{\Delta_\ell g(\overline{X}(T;\omega_{\ell,i}))}{M_\ell},$$

where $M_0 \in 2^{L+\lceil C_{\mathcal{ML}} L\rceil} 2^{\mathbb{N}}$ denotes the number of samples on the coarsest level with the constant $C_{\mathcal{ML}} \in (0,1)$, and the number of samples on higher levels is expressed in terms of M_0 by the ratio

$$\begin{aligned} M_\ell &= \frac{M_0}{2^L}\left\lceil 2^L \frac{\rho_{\mathrm{low}}(\mathrm{TOL}_{\mathrm{T},0})\mathrm{TOL}_{\mathrm{T},\ell}}{\rho_{\mathrm{low}}(\mathrm{TOL}_{\mathrm{T},\ell})\mathrm{TOL}_{\mathrm{T},0}}\right\rceil \\ &= \frac{M_0}{2^L}\left\lceil 2^{L+(\bar\gamma-1)\ell}\right\rceil, \quad \ell = 1, 2, \dots, L. \end{aligned} \tag{4.25}$$

The number of samples at the coarsest level is a stochastic process $M_0 : \mathbb{R}_+ \to 2^{\mathbb{N}+L+\lceil C_{\mathcal{ML}} L\rceil}$ defined by

$$M_0(\mathrm{TOL_S}) = \text{the smallest } k_0 \in 2^{\mathbb{N}+L+\lceil C_{\mathcal{ML}} L\rceil} \text{ such that } \mathcal{V}_{\mathcal{ML}}(g(\overline{X}(T)); k_0) < \frac{k_0 \mathrm{TOL_S}^2}{C_C^2}, \tag{4.26}$$

where

$$\begin{aligned} \mathcal{V}_{\mathcal{ML}}(g(\overline{X}(T)); k_0) &= \sum_{i=1}^{k_0} \frac{(g(\overline{X}_0(T;\omega_{0,i})) - \mathcal{A}(g(\overline{X}_0(T;\omega_{0,\cdot})); k_0))^2}{k_0 - 1} \\ &\quad + \sum_{\ell=1}^{L} \frac{k_0}{k_\ell} \sum_{i=1}^{k_\ell} \frac{(\Delta_\ell g(\overline{X}(T;\omega_{\ell,i})) - \mathcal{A}(\Delta_\ell g(\overline{X}(T;\omega_{\ell,\cdot})); k_\ell))^2}{k_\ell - 1} \\ &= \mathcal{V}(g(\overline{X}_0(T;\omega_{0,\cdot})); k_0) + 2^L \sum_{\ell=1}^{L} \frac{\mathcal{V}(\Delta_\ell g(\overline{X}_0(T;\omega_{\ell,\cdot})); k_\ell)}{\lceil 2^{L+\ell(\bar\gamma-1)}\rceil} \end{aligned} \tag{4.27}$$

and, analogous to the definition of M_ℓ,

$$k_\ell := \frac{k_0}{2^L}\left\lceil 2^{L+(\bar\gamma-1)\ell}\right\rceil, \quad \ell = 1, 2, \dots, L. \tag{4.28}$$

Remark 4.11. In the analysis of the adaptive SLMC algorithm, the requirement $M_0 \in 2^{\mathbb{N}+\lceil \log(1/\mathrm{TOL})\rceil}$ ensured that the number of samples used in the MC estimate fulfilled $\liminf_{\mathrm{TOL}\downarrow 0} M = \infty$. For the adaptive MLMC algorithm, we analogously ensure that $\liminf_{\mathrm{TOL}\downarrow 0} M_L = \infty$ by requiring that $M_0 \in 2^{\mathbb{N}+L+\lceil C_{\mathcal{ML}} L\rceil}$ for any positive constant $C_{\mathcal{ML}}$.

The stochastic process M_0 is defined in a similar way as the stochastic process M was defined for the SLMC algorithm, cf. (4.9). For the adaptive SLMC algorithm, asymptotic accuracy and complexity results were easily obtained by applying the asymptotic bounds of M, cf. Lemma 4.5. Applying the same strategy for the adaptive MLMC algorithm, we will derive asymptotic bounds for M_0 and use these bounds to prove the accuracy and complexity results of Theorem 4.1 and 4.2.

Lemma 4.12 (Asymptotic bounds for M_0). *Let*

$$\mathrm{Var}_{\mathcal{ML}}(g(\overline{X}(T))) := \mathrm{Var}(g(\overline{X}_0(T))) + 2^L \sum_{\ell=1}^{L} \frac{\mathrm{Var}(\Delta_\ell g(\overline{X}(T)))}{\lceil 2^{L+\ell(\bar{\gamma}-1)} \rceil}, \tag{4.29}$$

suppose that assumptions (4.4)–(4.6) *hold, and that* $\mathrm{Var}_{\mathcal{ML}}(g(\overline{X}(T))) > 0$ *for all sufficiently small* $\mathrm{TOL} > 0$. *Then* $M_0(\mathrm{TOL_S})$ *defined according to* (4.26) *fulfills*

$$\begin{aligned} \liminf_{\mathrm{TOL}\downarrow 0} \frac{M_0 \mathrm{TOL_S^2}}{\mathrm{Var}_{\mathcal{ML}}(g(\overline{X}(T)))C_C^2} &= 1 \quad \textit{in probability,} \\ \limsup_{\mathrm{TOL}\downarrow 0} \frac{M_0 \mathrm{TOL_S^2}}{\mathrm{Var}_{\mathcal{ML}}(g(\overline{X}(T)))C_C^2} &= 2 \quad \textit{in probability.} \end{aligned} \tag{4.30}$$

Proof. The definition of M_0 given in (4.26) implies that the following inequalities hold:

$$\frac{\mathcal{V}_{\mathcal{ML}}(g(\overline{X}(T)); M_0)}{\mathrm{Var}_{\mathcal{ML}}(g(\overline{X}(T)))} \le \frac{M_0 \mathrm{TOL_S^2}}{\mathrm{Var}_{\mathcal{ML}}(g(\overline{X}(T)))C_C^2} \le 2\frac{\mathcal{V}_{\mathcal{ML}}(g(\overline{X}(T)); M_0/2)}{\mathrm{Var}_{\mathcal{ML}}(g(\overline{X}(T)))}.$$

So to conclude the proof, we will show that

$$\lim_{\mathrm{TOL}\downarrow 0} \frac{\mathcal{V}_{\mathcal{ML}}(g(\overline{X}(T)); M_0)}{\mathrm{Var}_{\mathcal{ML}}(g(\overline{X}(T)))} = 1 \quad \text{in probability.} \tag{4.31}$$

Define the deterministic function

$$\tilde{k}_0(\mathrm{TOL_T}) = 2^{L(\mathrm{TOL_T}) + \lceil C_{\mathcal{ML}} L(\mathrm{TOL_T}) \rceil + 1}$$

and let $\{\tilde{k}_\ell\}_{\ell=1}^L$ be the corresponding level functions defined according to (4.28). Then, for a given $\epsilon > 0$, let us consider

$$\begin{aligned} &P\left(\left| \frac{\mathcal{V}_{\mathcal{ML}}(g(\overline{X}(T)); \tilde{k}_0)}{\mathrm{Var}_{\mathcal{ML}}(g(\overline{X}(T)))} - 1 \right| > \epsilon \right) \\ &\quad = P(|\mathcal{V}_{\mathcal{ML}}(g(\overline{X}(T)); \tilde{k}_0) - \mathrm{Var}_{\mathcal{ML}}(g(\overline{X}(T)))| > \mathrm{Var}_{\mathcal{ML}}(g(\overline{X}(T)))\epsilon) \\ &\quad \le P\Bigg(|\mathcal{V}(g(\overline{X}_0(T)); \tilde{k}_0) - \mathrm{Var}(g(\overline{X}_0(T)))| \\ &\qquad\qquad + \sum_{\ell=1}^{L} 2^L \lceil 2^{L+\ell(\bar{\gamma}-1)} \rceil^{-1} |\mathcal{V}(\Delta_\ell g(\overline{X}(T)); \tilde{k}_\ell) - \mathrm{Var}(\Delta_\ell g(\overline{X}(T)))| > \mathrm{Var}_{\mathcal{ML}}(g(\overline{X}(T)))\epsilon \Bigg) \\ &\quad \le P\left(|\mathcal{V}(g(\overline{X}_0(T)); \tilde{k}_0) - \mathrm{Var}(g(\overline{X}_0(T)))| > \frac{\mathrm{Var}_{\mathcal{ML}}(g(\overline{X}(T)))\epsilon}{L+1} \right) \\ &\qquad\qquad + \sum_{\ell=1}^{L} P\left(2^{(1-\bar{\gamma})\ell} |\mathcal{V}(\Delta_\ell g(\overline{X}(T)); \tilde{k}_\ell) - \mathrm{Var}(\Delta_\ell g(\overline{X}(T)))| > \frac{\mathrm{Var}_{\mathcal{ML}}(g(\overline{X}(T)))\epsilon}{L+1} \right). \end{aligned}$$

From the fourth moment bound (4.4), Chebycheff's inequality and k-Statistics bounds on the variance of the sample variance, cf. [22], we get that

$$P\left(|\mathcal{V}(g(\overline{X}_0(T)); \tilde{k}_0) - \mathrm{Var}(g(\overline{X}_0(T)))| > \frac{\mathrm{Var}_{\mathcal{ML}}(g(\overline{X}(T)))\epsilon}{L+1} \right) \le \frac{C(L+1)^2}{\mathrm{Var}_{\mathcal{ML}}(g(\overline{X}(T)))^2 \epsilon^2 \tilde{k}_0}.$$

The equality $2^{(1-\bar{\gamma})\ell} = \frac{\rho_{\mathrm{low}}(\mathrm{TOL}_{\mathrm{T},\ell})\mathrm{TOL}_{\mathrm{T},0}}{\rho_{\mathrm{low}}(\mathrm{TOL}_{\mathrm{T},0})\mathrm{TOL}_{\mathrm{T},\ell}}$ combined with (4.4) further yields that

$$P\left(2^{(1-\bar{\gamma})\ell} |\mathcal{V}(\Delta_\ell g(\overline{X}(T)); \tilde{k}_\ell) - \mathrm{Var}(\Delta_\ell g(\overline{X}(T)))| > \frac{\mathrm{Var}_{\mathcal{ML}}(g(\overline{X}(T)))\epsilon}{L+1} \right) \le \frac{C(L+1)^2}{\mathrm{Var}_{\mathcal{ML}}(g(\overline{X}(T)))^2 \epsilon^2 \tilde{k}_\ell}.$$

Since $\tilde{k}_0 = 2^{L + \lceil C_{\mathcal{ML}} L \rceil + 1}$, the definition of $\tilde{k}_\ell$ in (4.28) implies that

$$\tilde{k}_\ell \ge 2^{L + \lceil C_{\mathcal{ML}} L \rceil + 1 + (\bar{\gamma}-1)\ell} \quad \text{for } \ell = 1, 2, \ldots, L,$$

with $\bar{\gamma} \geq 0$ denoting the lower error density exponent in $\rho_{\mathrm{low}}(\mathrm{TOL_T}) = \mathrm{TOL_T^{\bar{\gamma}}}$, cf. (1.23). Consequently,

$$\begin{aligned}
P\left(\left|\frac{\mathcal{V}_{\mathcal{ML}}(g(\overline{X}(T)); \tilde{k}_0)}{\mathrm{Var}_{\mathcal{ML}}(g(\overline{X}(T)))} - 1\right| > \epsilon\right) &\leq \frac{C(L+1)^2}{\mathrm{Var}_{\mathcal{ML}}(g(\overline{X}(T)))^2 \epsilon^2 \tilde{k}_0} \sum_{\ell=0}^{L} \frac{\tilde{k}_0}{\tilde{k}_\ell} \\
&\leq \frac{C(L+1)^2}{\mathrm{Var}_{\mathcal{ML}}(g(\overline{X}(T)))^2 \epsilon^2 \tilde{k}_0} \sum_{\ell=0}^{L} 2^{(1-\bar{\gamma})\ell} \\
&< \frac{C(L+1)^2}{2^{\lceil C_{\mathcal{ML}} L\rceil + \bar{\gamma} L} \mathrm{Var}_{\mathcal{ML}}(g(\overline{X}(T)))^2 \epsilon^2}
\end{aligned}$$

which implies that for any $\epsilon > 0$,

$$\lim_{\mathrm{TOL}\downarrow 0} P\left(\left|\frac{\mathcal{V}_{\mathcal{ML}}(g(\overline{X}(T)); \tilde{k}_0)}{\mathrm{Var}_{\mathcal{ML}}(g(\overline{X}(T)))} - 1\right| > \epsilon\right) < \lim_{\mathrm{TOL}\downarrow 0} \frac{C(L+1)^2}{2^{\lceil C_{\mathcal{ML}} L\rceil + \bar{\gamma} L} \mathrm{Var}_{\mathcal{ML}}(g(\overline{X}(T)))^2 \epsilon^2} = 0.$$

Since $M_0 \geq \tilde{k}_0$ by definition, we conclude that also (4.31) holds, i.e.,

$$\lim_{\mathrm{TOL}\downarrow 0} P\left(\left|\frac{\mathcal{V}_{\mathcal{ML}}(g(\overline{X}(T)); M_0)}{\mathrm{Var}_{\mathcal{ML}}(g(\overline{X}(T)))} - 1\right| > \epsilon\right) = 0,$$

for any $\epsilon > 0$. □

Proof of Theorem 4.1

With the asymptotic bounds on M_0 we are ready to prove the main asymptotic accuracy result for the adaptive MLMC algorithm.

Proof. This proof is quite similar to the proof of Proposition 4.6 for the asymptotic accuracy in the single level setting, but for the sake of the differing details, a full proof is included in this setting also. For a given $\delta > 0$, we start by bounding the left-hand side of (4.1) by a product of the statistical error and the time discretization error

$$\begin{aligned}
&\liminf_{\mathrm{TOL}\downarrow 0} P(|\mathrm{E}[g(X(T))] - \mathcal{A}_{\mathcal{ML}}(g(\overline{X}(T)); M_0)| \leq \mathrm{TOL}) \\
&\geq \liminf_{\mathrm{TOL}\downarrow 0} P(|\mathrm{E}[g(X(T)) - g(\overline{X}_L(T))]| + |\mathrm{E}[g(\overline{X}_L(T))] - \mathcal{A}_{\mathcal{ML}}(g(\overline{X}(T)); M_0)| \leq C_S \mathrm{TOL_T} + \mathrm{TOL_S}) \\
&\geq \liminf_{\mathrm{TOL}\downarrow 0} P(|\mathrm{E}[g(X(T)) - g(\overline{X}_L(T))]| \leq (C_S + \delta)\mathrm{TOL_T} \text{ and } |\mathrm{E}[g(\overline{X}_L(T))] - \mathcal{A}_{\mathcal{ML}}(g(\overline{X}(T)); M_0)| \leq (1-\delta)\mathrm{TOL_S}) \\
&= \liminf_{\mathrm{TOL}\downarrow 0} P(|\mathrm{E}[g(X(T)) - g(\overline{X}_L(T))]| \leq (C_S + \delta)\mathrm{TOL_T}) \\
&\qquad\qquad \times P(|\mathrm{E}[g(\overline{X}_L(T))] - \mathcal{A}_{\mathcal{ML}}(g(\overline{X}(T)); M_0)| \leq (1-\delta)\mathrm{TOL_S}).
\end{aligned}$$

The time discretization error. The assumption that Lemma 1.3 and (4.4) hold implies that

$$\limsup_{\mathrm{TOL}\downarrow 0} \frac{|\mathrm{E}[g(X(T)) - g(\overline{X}(T))]|}{\mathrm{TOL}_T} \leq C_S,$$

cf. the proof of [25, Theorem 3.4]. By construction $\mathrm{TOL}_{\mathrm{T},L} = \mathrm{TOL_T}$, and this implies by the above that

$$\liminf_{\mathrm{TOL}\downarrow 0} P(|\mathrm{E}[g(X(T)) - g(\overline{X}_L(T))]| \leq (1+\delta) C_S \mathrm{TOL_T}) = 1.$$

The statistical error. From the above introduced $\delta > 0$, define the family of sets

$$\Omega_\delta(\mathrm{TOL_S}) = \left\{k \in 2^{\mathbb{N}+L+\lceil C_{\mathcal{ML}} L\rceil} \;\middle|\; 1-\delta < \frac{k\mathrm{TOL_S}^2}{\mathrm{Var}_{\mathcal{ML}}(g(\overline{X}(T)))C_C^2} \leq 2+\delta\right\}, \tag{4.32}$$

indexed by $\mathrm{TOL_S} > 0$. Lemma 4.12 then implies that $\lim_{\mathrm{TOL}\downarrow 0} P(M_0 \in \Omega_\delta) = 1$. Recall further that for the adaptive MLMC algorithm, the number of samples M_0 is determined in the step prior to generating the output $\mathcal{A}_{\mathcal{ML}}(g(\overline{X}(T)); M_0)$, so that M_0 is independent from $\mathcal{A}_{\mathcal{ML}}(g(\overline{X}(T)); M_0)$. Using this independence property and Fatou's lemma, the statistical error is bounded from below as follows:

$$\begin{aligned}
&\liminf_{\mathrm{TOL}\downarrow 0} P\big(|\mathrm{E}[g(\overline{X}_L(T))] - \mathcal{A}_{\mathcal{ML}}(g(\overline{X}(T)); M_0)| \le (1-\delta)\mathrm{TOL}_\mathrm{S}\big) \\
&\quad = \liminf_{\mathrm{TOL}\downarrow 0} \sum_{k_0 \in 2^{\mathbb{N}+L+\lceil C_{\mathcal{ML}} L\rceil}} P\big(|\mathrm{E}[g(\overline{X}_L(T))] - \mathcal{A}_{\mathcal{ML}}(g(\overline{X}(T)); k_0)| \le (1-\delta)\mathrm{TOL}_\mathrm{S}\big) P(M_0 = k_0) \\
&\quad \ge \liminf_{\mathrm{TOL}\downarrow 0} \sum_{k_0 \in \Omega_\delta} P\big(|\mathrm{E}[g(\overline{X}_L(T))] - \mathcal{A}_{\mathcal{ML}}(g(\overline{X}(T)); k_0)| \le (1-\delta)\mathrm{TOL}_\mathrm{S}\big) P(M_0 = k_0) \\
&\qquad\quad + \sum_{k_0 \in 2^{\mathbb{N}+L+\lceil C_{\mathcal{ML}} L\rceil}\setminus\Omega_\delta} \liminf_{\mathrm{TOL}\downarrow 0} P\big(|\mathrm{E}[g(\overline{X}_L(T))] - \mathcal{A}_{\mathcal{ML}}(g(\overline{X}(T)); k_0)| \le (1-\delta)\mathrm{TOL}_\mathrm{S}\big) P(M_0 = k_0) \qquad (4.33) \\
&\quad \ge \liminf_{\mathrm{TOL}\downarrow 0} \sum_{k_0 \in \Omega_\delta} P\bigg(\sqrt{k_0}\frac{|\mathrm{E}[g(\overline{X}_L(T))] - \mathcal{A}_{\mathcal{ML}}(g(\overline{X}(T)); k_0)|}{\sqrt{\mathrm{Var}_{\mathcal{ML}}(g(\overline{X}(T)))}} \le (1-\delta)^{3/2} C_C\bigg) P(M_0 = k_0) \\
&\quad \ge \int_{-(1-\delta)^{3/2}C_C}^{(1-\delta)^{3/2}C_C} \frac{e^{-x^2/2}}{\sqrt{2\pi}}\,dx.
\end{aligned}$$

The last inequality above follows from the application of Lindeberg–Feller's Central Limit Theorem (CLT) which is justified by Lemma 4.13 and the observation that $\mathrm{E}[\mathcal{A}_{\mathcal{ML}}(g(\overline{X}(T)); k_0)] = \mathrm{E}[g(\overline{X}_L(T))]$. The reasoning leading to inequality (4.33) is valid for any $\delta > 0$, so the proof of Theorem 4.1 is finished. □

Next we derive the weak convergence CLT result for the multilevel estimator $\mathcal{A}_{\mathcal{ML}}(g(\overline{X}(T)); k_0)$ which is needed in the proof of Theorem 4.1.

Lemma 4.13 (A CLT result). *Suppose the assumptions* (4.4)–(4.6) *hold and, in correspondence with the set defined in* (4.32), *let*

$$k_0(\mathrm{TOL}_\mathrm{S}) := \min\bigg\{k \in 2^{\mathbb{N}+L+\lceil C_{\mathcal{ML}} L\rceil} \;\bigg|\; \frac{k\mathrm{TOL}_\mathrm{S}{}^2}{\mathrm{Var}_{\mathcal{ML}}(g(\overline{X}(T)))C_C^2} > 1-\delta\bigg\},$$

for a given $\delta > 0$. *Then for any* $z \in \mathbb{R}_+$, *we have that*

$$\lim_{\mathrm{TOL}\downarrow 0} P\bigg(\sqrt{k_0}\frac{|\mathrm{E}[\mathcal{A}_{\mathcal{ML}}(g(\overline{X}(T)); k_0)] - \mathcal{A}_{\mathcal{ML}}(g(\overline{X}(T)); k_0)|}{\sqrt{\mathrm{Var}_{\mathcal{ML}}(g(\overline{X}(T)))}} \le z\bigg) = \int_{-z}^{z} \frac{e^{-x^2/2}}{\sqrt{2\pi}}\,dx. \qquad (4.34)$$

Proof. This lemma will be proved by verifying that the assumptions of the Lindeberg–Feller CLT are fulfilled, cf. Theorem A.1. Let us write

$$\sqrt{k_0}\frac{\mathrm{E}[\mathcal{A}_{\mathcal{ML}}(g(\overline{X}(T)); k_0)] - \mathcal{A}_{\mathcal{ML}}(g(\overline{X}(T)); k_0)}{\sqrt{\mathrm{Var}_{\mathcal{ML}}(g(\overline{X}(T)))}} = \sum_{i=1}^{K} Y_{K,i}$$

where $K := \sum_{\ell=0}^{L} k_\ell$ and the elements of $Y_{K,i}$ are independent and defined by

$$Y_{K,i} := \begin{cases}
\dfrac{\mathrm{E}[g(\overline{X}_0(T))] - g(\overline{X}_0(T;\omega_i))}{\sqrt{k_0}\sqrt{\mathrm{Var}_{\mathcal{ML}}(g(\overline{X}(T)))}} & \text{for } i = 1, 2, \dots, k_0, \\[2ex]
\dfrac{\sqrt{\frac{k_0}{k_1}}(\mathrm{E}[\Delta_1 g(\overline{X}(T))] - \Delta_1 g(\overline{X}(T;\omega_i)))}{\sqrt{k_1}\sqrt{\mathrm{Var}_{\mathcal{ML}}(g(\overline{X}(T)))}} & \text{for } i = k_0+1, \dots, k_0+k_1, \\
\vdots & \vdots \\
\dfrac{\sqrt{\frac{k_0}{k_L}}(\mathrm{E}[\Delta_L g(\overline{X}(T))] - \Delta_L g(\overline{X}(T;\omega_i)))}{\sqrt{k_L}\sqrt{\mathrm{Var}_{\mathcal{ML}}(g(\overline{X}(T)))}} & \text{for } i = k_{L-1}+1, \dots, K.
\end{cases}$$

Then it follows that

$$\sum_{i=1}^{K} \mathrm{E}[Y_{K,i}^2] = \frac{\mathrm{Var}_{\mathcal{ML}}(g(\overline{X}(T)))}{\mathrm{Var}_{\mathcal{ML}}(g(\overline{X}(T)))} = 1 \quad \text{for all TOL} > 0,$$

so condition (a) of Theorem A.1 is fulfilled. To verify that condition (b) of Theorem A.1 is fulfilled, one must show that for any $\epsilon > 0$,

$$\limsup_{\mathrm{TOL}\to 0} \sum_{i=1}^{K} \mathrm{E}[Y_{K,i}^2 1_{|Y_{K,i}|>\epsilon}] = 0.$$

The definition of k_ℓ, cf. (4.28), combined with the moment bound (4.4) implies that there exists a $C > 0$ such that

$$\mathrm{E}\biggl[\Bigl(\frac{k_0}{k_\ell}\Bigr)^2 |\Delta_\ell g(\overline{X}(T)) - \mathrm{E}[\Delta_\ell g(\overline{X}(T))]|^4\biggr] \le C \quad \text{for all } \ell \in \{1, 2, \dots, L\}.$$

Using Chebycheff's inequality and the fact that $k_L \ge 2^{\lceil C_{\mathcal{ML}} L\rceil + \bar{\gamma} L + 1}$, cf. (4.28), we derive that

$$\begin{aligned}
\sum_{i=1}^{K} \mathrm{E}[Y_{K,i}^2 1_{|Y_{K,i}|>\epsilon}] &\le \sum_{i=1}^{K} \epsilon^{-2} \mathrm{E}[Y_{K,i}^4] \\
&= \frac{1}{\epsilon^2 \operatorname{Var}_{\mathcal{ML}}(g(\overline{X}(T)))^2} \biggl\{ \frac{1}{k_0} \mathrm{E}[|g(\overline{X}_0(T)) - \mathrm{E}[g(\overline{X}_0(T))]|^4] \\
&\qquad + \sum_{\ell=1}^{L} \frac{1}{k_\ell} \mathrm{E}\biggl[\Bigl(\frac{k_0}{k_\ell}\Bigr)^2 |\Delta_\ell g(\overline{X}(T)) - \mathrm{E}[\Delta_\ell g(\overline{X}(T))]|^4\biggr]\biggr\} \\
&\le \frac{C}{\epsilon^2 \operatorname{Var}_{\mathcal{ML}}(g(\overline{X}(T)))^2} \sum_{\ell=0}^{L} k_\ell^{-1} \le \frac{C\,L}{k_L\, \epsilon^2 \operatorname{Var}_{\mathcal{ML}}(g(\overline{X}(T)))^2} \to 0 \quad \text{as TOL} \downarrow 0.
\end{aligned}$$

This verifies that condition (b) is fulfilled. □

We conclude the analysis of the adaptive MLMC algorithm by estimating the work required to fulfill the accuracy estimate (4.1). We recall that WORK(TOL), defined in (4.2) by

$$\mathrm{WORK(TOL)} = \sum_{\ell=0}^{L} \mathrm{E}[M_\ell]\mathrm{E}[N_\ell],$$

is an estimate of the average number of operations required in the generation of $\mathcal{A}_{\mathcal{ML}}(g(\overline{X}(T)); M_0)$ to approximate $\mathrm{E}[g(X(T))]$ with the prescribed confidence C_C and accuracy TOL. First, let us derive an asymptotic bound for $\mathrm{E}[M_0]$.

Lemma 4.14. *Suppose the assumptions* (4.4)–(4.6) *hold. Then the number of samples M_0 used at the base level of the MLMC algorithm approximation of* $\mathrm{E}[g(X(T))]$ *satisfies*

$$\limsup_{\mathrm{TOL}\downarrow 0} \frac{\mathrm{E}[M_0]\mathrm{TOL}_{\mathrm{S}}^2}{\operatorname{Var}_{\mathcal{ML}}(g(\overline{X}(T)))C_C^2} \le 2. \tag{4.35}$$

Proof. For given $\delta > 0$, define the deterministic function

$$\widetilde{M}_0(\mathrm{TOL}) = \min\biggl\{k \in 2^{\mathbb{N}+L+\lceil C_{\mathcal{ML}} L\rceil} \biggm| \frac{k_0 \mathrm{TOL}^2}{\operatorname{Var}_{\mathcal{ML}}(g(\overline{X}(T)))C_C^2} > 1 + \delta\biggr\}.$$

By the relation (4.15), the moment bound assumption (4.4), Hölder's inequality, and k-Statistics bounds on the variance of the sample variance, cf. [22], we derive that

$$\begin{aligned}
P(M_0 = 2\widetilde{M}_0) &\le P\biggl(\frac{\mathcal{V}_{\mathcal{ML}}(g(\overline{X}(T)); \widetilde{M}_0)}{\operatorname{Var}_{\mathcal{ML}}(g(\overline{X}(T)))} > \widetilde{M}_0 \frac{\mathrm{TOL}^2}{\operatorname{Var}_{\mathcal{ML}}(g(\overline{X}(T)))C_C^2}\biggr) \\
&\le P\biggl(\frac{\mathcal{V}_{\mathcal{ML}}(g(\overline{X}(T)); \widetilde{M}_0)}{\operatorname{Var}_{\mathcal{ML}}(g(\overline{X}(T)))} > 1 + \delta\biggr) \\
&\le P(\mathcal{V}_{\mathcal{ML}}(g(\overline{X}(T)); \widetilde{M}_0) - \operatorname{Var}_{\mathcal{ML}}(g(\overline{X}(T))) > \delta \operatorname{Var}_{\mathcal{ML}}(g(\overline{X}(T)))) \\
&\le \mathrm{E}\biggl[\frac{|\mathcal{V}_{\mathcal{ML}}(g(\overline{X}(T)); \widetilde{M}_0) - \operatorname{Var}_{\mathcal{ML}}(g(\overline{X}(T)))|^2}{\delta^2 \operatorname{Var}_{\mathcal{ML}}(g(\overline{X}(T)))^2}\biggr] \\
&\le \frac{\operatorname{Var}(\mathcal{V}(g(\overline{X}_0(T)); \widetilde{M}_0)) + \sum_{\ell=1}^{L} \operatorname{Var}(\mathcal{V}(\Delta_\ell g(\overline{X}(T)); \widetilde{M}_\ell))}{\delta^2 \operatorname{Var}_{\mathcal{ML}}(g(\overline{X}(T)))^2} \le \frac{CL}{\delta^2 \operatorname{Var}_{\mathcal{ML}}(g(\overline{X}(T)))^2 \widetilde{M}_L},
\end{aligned}$$

and for $\ell = 1, 2, \ldots$ that

$$\begin{aligned} P(M_0 = 2^{\ell+1}\widetilde{M}_0) &\le P(\mathcal{V}_{\mathcal{ML}}(g(\overline{X}(T)); 2^{\ell}\widetilde{M}_0) - \mathrm{Var}_{\mathcal{ML}}(g(\overline{X}(T))) > 2^{\ell-1}\mathrm{Var}_{\mathcal{ML}}(g(\overline{X}(T)))) \\ &\le \mathrm{E}\bigg[\frac{|\mathcal{V}_{\mathcal{ML}}(g(\overline{X}(T)); 2^{\ell}\widetilde{M}_0) - \mathrm{Var}_{\mathcal{ML}}(g(\overline{X}(T)))|^2}{2^{2(\ell-1)}\mathrm{Var}_{\mathcal{ML}}(g(\overline{X}(T)))^2}\bigg] \\ &< \frac{CL}{2^{3\ell}\,\mathrm{Var}_{\mathcal{ML}}(g(\overline{X}(T)))^2\widetilde{M}_L}. \end{aligned}$$

Consequently,

$$\begin{aligned} \frac{\mathrm{E}[M_0]\mathrm{TOL}_{\mathrm{S}}^2}{\mathrm{Var}(g(\overline{X}(T)))C_C^2} &\le \bigg[P(M_0 \le \widetilde{M}_0) + \sum_{\ell=1}^{\infty} 2^{\ell} P(M_0 = 2^{\ell}\widetilde{M}_0)\bigg]\frac{\widetilde{M}_0\mathrm{TOL}_{\mathrm{S}}^2}{\mathrm{Var}(g(\overline{X}(T)))C_C^2} \\ &\le 2(1+\delta)\bigg[P(M_0 \le \widetilde{M}_0) + P(M_0 = 2\widetilde{M}_0) + \sum_{\ell=1}^{\infty} 2^{\ell+1}P(M_0 = 2^{\ell+1}\widetilde{M}_0)\bigg] \\ &\le 2(1+\delta)\bigg[P(M_0 \le \widetilde{M}_0) + \frac{CL}{\delta^2\widetilde{M}_L} + \frac{CL}{\widetilde{M}_L}\sum_{\ell=1}^{\infty} 2^{-2\ell}\bigg]. \end{aligned}$$

Taking limits in the above inequality leads to

$$\limsup_{\mathrm{TOL}\downarrow 0} \frac{\mathrm{E}[M_0]\mathrm{TOL}_{\mathrm{S}}^2}{\mathrm{Var}_{\mathcal{ML}}(g(\overline{X}(T)))C_C^2} \le 2(1+\delta).$$

Finally, observe that since the obtained inequality holds true for any $\delta > 0$, the proof is finished. □

An asymptotic bound on $\mathrm{E}[N_\ell]$ may be deduced from the single level result of Lemma 4.8. For the convenience of the reader we present the result of Lemma 4.8 in a way that is fitting for the multilevel setting.

Lemma 4.15 (Multilevel asymptotic average number of time steps). *Suppose the assumptions of Lemma 1.3 and* (4.4)–(4.6) *hold. Then the final number of time steps generated by the adaptive MLMC algorithm with time steps* (2.19) *and* (2.20) *and* $\mathrm{TOL}_{\mathrm{T},\ell} = 2^{-\ell}\mathrm{TOL}_{\mathrm{T},0}$ *satisfies*

$$\limsup_{\ell\uparrow\infty} \mathrm{TOL}_{\mathrm{T},\ell}\mathrm{E}[N_\ell] \le \frac{4}{C_R}\bigg(\mathrm{E}\bigg[\int_0^T \sqrt{|\hat{\rho}(t)|}dt\bigg]\bigg)^2. \tag{4.36}$$

Proof of Theorem 4.2

With bounds for $\mathrm{E}[M_0]$ and $\mathrm{E}[N_\ell]$ at hand, we are ready to prove the main complexity theorem for the adaptive MLMC algorithm.

Proof. First, we note that the conditions $\bar{\gamma} \to 0$ and $L\bar{\gamma} \to \infty$ as $\mathrm{TOL} \downarrow 0$ yields a consistent lower error density, since it leads to

$$\rho_{\mathrm{low}}(\mathrm{TOL}_{\mathrm{T}}) = \mathrm{TOL}_{\mathrm{T}}^{\bar{\gamma}} = \mathcal{O}(2^{-L\bar{\gamma}}),$$

which implies that

$$\rho_{\mathrm{low}}(\mathrm{TOL}_{\mathrm{T}}) \to 0 \quad \text{as } \mathrm{TOL} \downarrow 0.$$

Lemma 4.15 implies that for any given $\delta > 0$, there exists an $\hat{L}(\delta)$ not depending on TOL such that

$$\mathrm{TOL}_{\mathrm{T},\ell}\mathrm{E}[N_\ell] \le (1+\delta)\frac{4}{C_R}\bigg(\mathrm{E}\bigg[\int_0^T \sqrt{|\hat{\rho}(\tau)|}d\tau\bigg]\bigg)^2 \quad \text{for all } \ell \ge \hat{L}. \tag{4.37}$$

Furthermore, recall that M_ℓ as defined in (4.25) fulfills

$$\mathrm{E}[M_\ell] \le (2^{\ell(\bar{\gamma}-1)} + 2^{-L})\mathrm{E}[M_0] \quad \text{for all } \ell \in \{0, 1, \ldots, L\}.$$

By this property, intequality (4.37), the monotonic relation $N_\ell \le N_{\ell+1}$, and recalling that by construction $\mathrm{TOL}_{\mathrm{T},0} > \mathrm{TOL}_{\mathrm{T,Max}}$,

$$\begin{aligned}\sum_{\ell=0}^{L} \mathrm{E}[M_\ell]\mathrm{E}[N_\ell] &\le \mathrm{E}[N_{\widehat{L}}]\mathrm{TOL}_{\mathrm{T},\widehat{L}} \sum_{\ell=0}^{\widehat{L}} \frac{\mathrm{E}[M_\ell]}{\mathrm{TOL}_{\mathrm{T},\widehat{L}}} + \sum_{\ell=\widehat{L}+1}^{L} \frac{\mathrm{E}[M_\ell]}{\mathrm{TOL}_{\mathrm{T},\ell}} \mathrm{E}[N_\ell]\mathrm{TOL}_{\mathrm{T},\ell} \\ &\le \frac{(1+\delta)4\mathrm{E}[M_0]}{C_R\,\mathrm{TOL}_{\mathrm{T},0}} \left(\mathrm{E}\left[\int_0^T \sqrt{|\widehat{\rho}(\tau)|d\tau}\right]\right)^2 \left(2^{\widehat{L}} \sum_{\ell=0}^{\widehat{L}-1} (2^{\ell(\bar{\gamma}-1)} + 2^{-L}) + \sum_{\ell=\widehat{L}}^{L} (2^{\ell\bar{\gamma}} + 2^{-L+\ell})\right) \\ &\le \frac{(1+\delta)4\mathrm{E}[M_0]}{C_R\,\mathrm{TOL}_{\mathrm{T,Max}}} \left(\mathrm{E}\left[\int_0^T \sqrt{|\widehat{\rho}(\tau)|d\tau}\right]\right)^2 \left(\frac{2^{\widehat{L}}}{1-2^{\bar{\gamma}-1}} + \widehat{L}2^{\widehat{L}-L} + \frac{2^{(L+1)\bar{\gamma}}}{\log(2^{\bar{\gamma}})} + 2\right).\end{aligned}$$

The asymptotics of $\bar{\gamma}$ imply that

$$\lim_{\mathrm{TOL}\downarrow 0} \frac{\bar{\gamma}}{2^{\bar{\gamma}L}} \left(\frac{2^{\widehat{L}}}{1-2^{\bar{\gamma}-1}} + \widehat{L}2^{\widehat{L}-L} + \frac{2^{(L+1)\bar{\gamma}}}{\log(2^{\bar{\gamma}})} + 2\right) = \frac{1}{\log(2)}.$$

Lemma 4.14 and (4.40) then yield

$$\limsup_{\mathrm{TOL}\downarrow 0} \frac{\mathrm{WORK}(\mathrm{TOL})\,\mathrm{TOL}_{\mathrm{S}}^2\,\bar{\gamma}}{\mathrm{Var}_{\mathcal{ML}}(g(\overline{X}(T)))\,2^{\bar{\gamma}L}} \le (1+\delta) \frac{8\,C_C^2}{\log(2)\,\mathrm{TOL}_{\mathrm{T,Max}}\,C_R} \left(\mathrm{E}\left[\int_0^T \sqrt{|\widehat{\rho}(\tau)|d\tau}\right]\right)^2. \tag{4.38}$$

We observe that $\mathrm{WORK}(\mathrm{TOL}) = \mathcal{O}(\mathrm{TOL}_{\mathrm{S}}^{-2}\mathrm{Var}_{\mathcal{ML}}(g(\overline{X}(T)))2^{\bar{\gamma}L})$. To obtain a bound on more explicit form, the assumption (4.4) on L^p convergence implies there exists a $C_G > 0$ such that[10]

$$\limsup_{\ell\uparrow\infty} \frac{\rho_{\mathrm{low}}(\mathrm{TOL}_{\mathrm{T},\ell})}{\mathrm{TOL}_{\mathrm{T},\ell}} \mathrm{E}[|\Delta_\ell g(\overline{X}(T))|^2] \le C_G. \tag{4.39}$$

Inequality (4.39) further implies that

$$\limsup_{\mathrm{TOL}\downarrow 0} \frac{\mathrm{Var}_{\mathcal{ML}}(g(\overline{X}(T)))}{L} \le C_G, \tag{4.40}$$

which in turn yields

$$\limsup_{\mathrm{TOL}\downarrow 0} \frac{\mathrm{WORK}(\mathrm{TOL})\,\mathrm{TOL}_{\mathrm{S}}^2\,\bar{\gamma}}{\mathrm{Var}_{\mathcal{ML}}(g(\overline{X}(T)))\,2^{\bar{\gamma}L}} \le (1+\delta) \frac{8\,C_C^2\,C_G}{\log(2)\,\mathrm{TOL}_{\mathrm{T,Max}}\,C_R} \left(\mathrm{E}\left[\int_0^T \sqrt{|\widehat{\rho}(\tau)|d\tau}\right]\right)^2. \tag{4.41}$$

We approximately minimize the complexity by the splitting choice

$$\mathrm{TOL}_{\mathrm{S}} = \frac{2}{2+\bar{\gamma}(\mathrm{TOL})}\mathrm{TOL} \quad \text{and} \quad \mathrm{TOL}_{\mathrm{T}} = \frac{\bar{\gamma}(\mathrm{TOL})}{(2+\bar{\gamma}(\mathrm{TOL}))C_S}\mathrm{TOL},$$

which fulfills the restrictions $C_S\mathrm{TOL}_{\mathrm{T}} + \mathrm{TOL}_{\mathrm{S}} = \mathrm{TOL}$ and $\mathrm{TOL}_{\mathrm{T}} \le \mathrm{TOL}_{\mathrm{S}}$. Applying this splitting choice in (4.41) and noting that the proof argument is valid for all $\delta > 0$ leads to (4.3). □

For settings where $\hat{\rho}$ is bounded from below by a positive real, adaptive MLMC has the same complexity as uniform MLMC.

Corollary 4.16. *Suppose that the assumptions of Lemma 1.3 and* (4.4)–(4.6) *hold, that* $\rho_{\mathrm{low}}(\mathrm{TOL}_{\mathrm{T}}) = \rho_{\min} \in \mathbb{R}_+$, *and*

$$\min_{\tau\in[0,T]} |\widehat{\rho}(\tau)| \ge \rho_{\min} \quad \textit{a.s.} \tag{4.42}$$

Then

$$\limsup_{\mathrm{TOL}\downarrow 0} \frac{\mathrm{WORK}(\mathrm{TOL})\mathrm{TOL}^2}{L^2} \le \frac{8\,C_C^2\,C_G}{\mathrm{TOL}_{\mathrm{T,Max}}\,C_R} \left(\mathrm{E}\left[\int_0^T \sqrt{|\widehat{\rho}(\tau)|d\tau}\right]\right)^2. \tag{4.43}$$

[10] See Remark 4.18 for a discussion on how to estimate C_G.

Proof. For $\rho_{\text{low}}(\text{TOL}_{\text{T}}) = \rho_{\min}$, M_ℓ as defined in (4.25) fulfills

$$\mathrm{E}[M_\ell] = 2^{-\ell}\mathrm{E}[M_0] \quad \text{for all } \ell \in \{0, 1, \dots, L\}. \tag{4.44}$$

By inequality (4.37), equation (4.44), and the monotonic relation $N_\ell \le N_{\ell+1}$,

$$\begin{aligned}
\sum_{\ell=0}^{L} \mathrm{E}[M_\ell]\mathrm{E}[N_\ell] &\le \mathrm{E}[N_{\hat{L}}]\text{TOL}_{\text{T},\hat{L}} \sum_{\ell=0}^{\hat{L}} \frac{\mathrm{E}[M_\ell]}{\text{TOL}_{\text{T},\hat{L}}} + \sum_{\ell=\hat{L}+1}^{L} \frac{\mathrm{E}[M_\ell]}{\text{TOL}_{\text{T},\ell}} \mathrm{E}[N_\ell]\text{TOL}_{\text{T},\ell} \\
&\le \frac{4(1+\delta)\mathrm{E}[M_0]}{C_R\, \text{TOL}_{\text{T},0}} \left(\mathrm{E}\left[\int_0^T \sqrt{|\hat{\rho}(\tau)|}\, d\tau \right] \right)^2 \left(2^{\hat{L}} \sum_{\ell=0}^{\hat{L}-1} 2^{-\ell} + \sum_{\ell=\hat{L}}^{L} 1 \right) \\
&\le \frac{4(1+\delta)\mathrm{E}[M_0]}{C_R\, \text{TOL}_{\text{T},0}} \left(\mathrm{E}\left[\int_0^T \sqrt{|\hat{\rho}(\tau)|}\, d\tau \right] \right)^2 (2^{\hat{L}+1} + (L - \hat{L})).
\end{aligned} \tag{4.45}$$

Recalling the definition $L = \lfloor \log_2(\text{TOL}_{\text{T,Max}}/\text{TOL}_{\text{T}}) \rfloor$ and that $\hat{L}$ is fixed, it follows that

$$\lim_{\text{TOL}\downarrow 0} \frac{2^{\hat{L}+1} + (L - \hat{L})}{L} = 1.$$

Using (4.45) combined with Lemma 4.14 and recalling that $\text{TOL}_{\text{T},0} > \text{TOL}_{\text{T,Max}}/2$, we obtain the bound

$$\limsup_{\text{TOL}\downarrow 0} \frac{\text{WORK(TOL)}\, \text{TOL}_{\text{S}}^2}{\text{Var}_{\mathcal{ML}}(g(\overline{X}(T)))L} \le \frac{16(1+\delta)\mathrm{E}[M_0]}{C_R\, \text{TOL}_{\text{T,Max}}} \left(\mathrm{E}\left[\int_0^T \sqrt{|\hat{\rho}(\tau)|}\, d\tau \right] \right)^2. \tag{4.46}$$

To approximately minimize the complexity, we introduce the splitting choice

$$\begin{aligned}
\text{TOL}_{\text{S}} &= \frac{\log(\text{TOL}^{-1})}{\log(\text{TOL}^{-1}) + \log(\log(\text{TOL}^{-1}))}\text{TOL}, \\
\text{TOL}_{\text{T}} &= \frac{\log(\log(\text{TOL}^{-1}))}{(\log(\text{TOL}^{-1}) + \log(\log(\text{TOL}^{-1})))C_S}\text{TOL}.
\end{aligned}$$

Combining (4.39) with the above splitting choice in inequality (4.46), and noting that this bound is valid for any $\delta > 0$ leads to (4.43). □

Remark 4.17 (Splitting of the tolerance). The optimal choices of TOL_{S} and TOL_{T} given TOL obtained in the proof allocates most of the tolerance to the statistical error when TOL is small. This differs from the equal splitting between TOL_{S} and TOL_{T} used in the numerical experiments which were sub-optimal in that sense.

Remark 4.18 (Particular estimate for the constant C_G). It is possible to estimate the asymptotic constant C_G given in inequality (4.39). For instance, when the exact error density is bounded away from zero so there exists a constant $\rho_{\min}$ such that $\hat{\rho} > \rho_{\min} > 0$ a.s. and the SDE is given by

$$\begin{aligned}
dX(t) &= b(X(t))dW(t), \quad t > 0, \\
X(0) &= X_0,
\end{aligned}$$

then we have

$$C_G \le C_S \mathrm{E}\left[\left\| \frac{(b'b)^2(X(t))(\varphi)^2(t)}{\hat{\rho}(t)} \right\|_{L^\infty([0,T])} \right].$$

Here $\varphi(t) = g'(X(T))\frac{X'(T)}{X'(t)}$ and the first variation $X'(s)$ solves, for $s > 0$, the linear equation

$$dX'(s) = b'(X(s))X'(s)dW(s),$$

with initial condition $X'(0) = 1$. The constant C_S is the parameter in the stopping condition (2.19).

Remark 4.19 (Jump diffusions). It is possible to extend these results of adaptive multilevel weak approximation for diffusions to the case of jump diffusions with time dependent jump measure analyzed in [27].

5 Conclusions

In this work we presented and analyzed an adaptive multilevel Monte Carlo algorithm, where the multilevel simulations are performed on adaptively generated mesh hierarchies based on computable a posteriori weak error estimates. The theoretical analysis of the adaptive algorithm showed that the algorithm stops after a finite number of steps, and proceeded to show accuracy and efficiency results under natural assumptions in Theorems 4.1 and 4.2. In particular, Theorem 4.1 states that the probability of the weak error being bounded by the specified tolerance TOL is asymptotically bounded by any desired probability through the confidence parameter. Theorem 4.2 states computational complexity results where the involved constants are explicitly given in terms of algorithm parameters and problem properties. It shows that the $L^{1/2}$-quasi norm of the error density appears as a multiplicative constant in the complexity bounds, instead of the larger L^1-norm of the same error density that would appear using a uniform time stepping MLMC algorithm; the difference between these two factors can be arbitrarily large even in problems with smooth coefficients where they are both finite. Disregarding the constants the result shows that, depending on assumptions on the limit error density and the lower bound on the computed error density used by the adaptive algorithm, the complexity can be either the same as or nearly the same as the complexity uniform MLMC has in cases where the order of strong convergence of the Euler–Maruyama method is $1/2$.

Numerical results for scalar SDEs confirmed the theoretical analysis. For the two problems with reduced weak convergence order a simple single level Monte Carlo method has complexity $\mathcal{O}(\mathrm{TOL}^{-4})$ while the adaptive MLMC method has the improved complexity $\mathcal{O}(\mathrm{TOL}^{-2}\log_2(\mathrm{TOL}_0/\mathrm{TOL})^2)$. The use of advanced Monte Carlo methods such as the adaptive MLMC algorithm presented in this paper is most attractive for SDEs in higher dimension, where the corresponding standard PDE-based computational techniques are not competitive. It would also be interesting to compare adaptive MLMC with uniform MLMC for Barrier problems in higher dimensions, since it is not clear that the order of strong convergence of the Euler–Maruyama method will be $(1-\delta)/2$, for any positive δ, in that case. The fact that computational complexity of uniform multilevel Monte Carlo, disregarding constants, depends on the strong convergence indicates that adaptive mesh refinements based on strong error estimates can also be used to improve the computational efficiency; such methods are also subjects of ongoing research and higher-dimensional examples will be treated in that context.

In this paper the adaptive algorithms were presented with global error control in the quantity of interest, starting from a given coarse mesh. Alternatively, local error estimates can be applied to control the adaptive time stepping in the computation of the forward problem. This approach can be used on its own when global error control is deemed unnecessary or too computationally expensive, but it can also be used together with the global error control in situations with stiff SDEs where any given initial mesh can be too coarse depending on the realization. This is particularly relevant for MLMC simulations where stability issues in the computations on the coarsest level can destroy the results of the whole multilevel simulation, as was pointed out by Hutzenthaler, Jentzen, and Kloeden in [18].

A Theorem

Theorem A.1 (Lindeberg–Feller Theorem [7, p. 114]). *For each n, let $X_{n,m}$, $1 \le m \le n$, be independent random variables with* $\mathrm{E}[X_{n,m}] = 0$. *Suppose:*

(a) *we have*

$$\sum_{m=1}^{n} \mathrm{E}[X_{n,m}^2] \to \sigma^2 > 0,$$

(b) *for all* $\delta > 0$,

$$\lim_{n\to\infty} \sum_{m=1}^{n} \mathrm{E}[X_{n,m}^2 1_{|X_{n,m}|>\delta}] = 0.$$

Then the Central Limit Theorem holds, i.e., the random variable

$$S_n := \sum_{m=1}^{n} X_{n,m} \rightharpoonup \sigma\Xi \text{ as } n \to \infty,$$

where Ξ is a standard normal distributed random variable.

Acknowledgement: The authors would like to thank Mike Giles and the two anonymous reviewers for valuable comments.

Funding: This work was supported by the Dahlquist fellowship at the Royal Institute of Technology in Stockholm, Sweden; the Department of Scientific Computing in Florida State University; the University of Austin Subcontract (Project Number 024550, Center for Predictive Computational Science); the VR project “Effektiva numeriska metoder för stokastiska differentialekvationer med tillämpningar”; the Center for Industrial and Applied Mathematics (CIAM) at the Royal Institute of Technology; and King Abdullah University of Science and Technology (KAUST). The first author and the fourth author are members of the Research Center on Uncertainty Quantification in Computational Science and Engineering at KAUST, and the second author was a member while doing part of the research.

References

[1] F. K. Abdullaev, J. C. Bronski and G. Papanicolaou, Soliton perturbations and the random Kepler problem, *Phys. D* **135** (2000), 369–386.

[2] C. Bayer, H. Hoel, E. von Schwerin and R. Tempone, On non-asymptotic optimal stopping criteria in Monte Carlo simulations, KTH, Numerical Analysis, NA, Report no. 2012:7, 2012.

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

[4] F. M. Buchmann and W. P. Petersen, An exit probability approach to solving high dimensional Dirichlet problems, *SIAM J. Sci. Comput.* **28** (2006), 1153–1166.

[5] E. Cancès, F. Legoll and G. Stoltz, Theoretical and numerical comparison of some sampling methods for molecular dynamics, *M2AN Math. Model. Numer. Anal.* **41** (2007), 351–389.

[6] Y. S. Chow and H. Robbins, On the asymptotic theory of fixed-width sequential confidence intervals for the mean, *Ann. Math. Statist.* **36** (1965), 457–462.

[7] R. Durrett, *Probability: Theory and Examples*, 2nd ed., Duxbury Press, Belmont,1996.

[8] A. Dzougoutov, K.-S. Moon, E. von Schwerin, A. Szepessy and R. Tempone, Adaptive Monte Carlo algorithms for stopped diffusion, in: *Multiscale Methods in Science and Engineering*, Lect. Notes Comput. Sci. Eng. 44, Springer-Verlag, Berlin (2005), 59–88.

[9] J. G. Gaines and T. J. Lyons, Variable step size control in the numerical solution of stochastic differential equations, *SIAM J. Appl. Math.* **57** (1997), 1455–1484.

[10] M. B. Giles, Improved multilevel Monte Carlo convergence using the Milstein scheme, in: *Monte Carlo and Quasi-Monte Carlo Methods 2006* (Ulm 2006), Springer-Verlag, Berlin (2008), 343–358.

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

[12] M. B. Giles, D. J. Higham and X. Mao, Analysing multi-level Monte Carlo for options with non-globally Lipschitz payoff, *Finance Stoch.* **13** (2009), 403–413.

[13] P. Glasserman, *Monte Carlo Methods in Financial Engineering*, Appl. Math. 53, Springer-Verlag, New York, 2004.

[14] E. Gobet, Weak approximation of killed diffusion using Euler schemes, *Stochastic Process. Appl.* **87** (2000), 167–197.

[15] S. Heinrich, Monte Carlo complexity of global solution of integral equations, *J. Complexity* **14** (1998), 151–175.

[16] S. Heinrich and E. Sindambiwe, Monte Carlo complexity of parametric integration, *J. Complexity* **15** (1999), 317–341.

[17] H. Hoel, E. von Schwerin, A. Szepessy and R. Tempone, Adaptive multilevel Monte Carlo simulation, in: *Multiscale Methods in Science and Engineering*, Lect. Notes Comput. Sci. Eng. 82, Springer-Verlag, Berlin (2012), 217–234.

[18] M. Hutzenthaler, A. Jentzen and P. E. Kloeden, Divergence of the multilevel Monte Carlo Euler method for nonlinear stochastic differential equations, preprint (2013), http://arxiv.org/abs/1105.0226v3.

[19] E. Jouini, J. Cvitanić and M. Musiela, *Option Pricing, Interest Rates and Risk Management*, Handb. Math. Finance, Cambridge University Press, Cambridge, 2001.

[20] I. Karatzas and S. E. Shreve, *Brownian Motion and Stochastic Calculus*, 2nd ed., Grad. Texts in Math. 113, Springer-Verlag, New York, 1991.

[21] A. Kebaier, Statistical Romberg extrapolation: A new variance reduction method and applications to option pricing, *Ann. Appl. Probab.* **15** (2005), 2681–2705.

[22] E. S. Keeping, *Introduction to Statistical Inference*, D. Van Nostrand, Princeton, 1962.

[23] A. J. Majda, I. Timofeyev and E. Vanden Eijnden, A mathematical framework for stochastic climate models, *Comm. Pure Appl. Math.* **54** (2001), 891–974.

[24] G. N. Milstein and M. V. Tretyakov, Simulation of a space-time bounded diffusion, *Ann. Appl. Probab.* **9** (1999), 732–779.

[25] K.-S. Moon, A. Szepessy, R. Tempone and G. E. Zouraris, Convergence rates for adaptive weak approximation of stochastic differential equations, *Stoch. Anal. Appl.* **23** (2005), 511–558.

[26] K.-S. Moon, E. von Schwerin, A. Szepessy and R. Tempone, An adaptive algorithm for ordinary, stochastic and partial differential equations, in: *Recent Advances in Adaptive Computation*, Contemp. Math. 383, American Mathematical Society, Providence (2005), 325–343.

[27] E. Mordecki, A. Szepessy, R. Tempone and G. E. Zouraris, Adaptive weak approximation of diffusions with jumps, *SIAM J. Numer. Anal.* **46** (2008), 1732–1768.

[28] B. Øksendal, *Stochastic Differential Equations*, 5th ed., Universitext, Springer-Verlag, Berlin, 1998.

[29] A. Szepessy, R. Tempone and G. E. Zouraris, Adaptive weak approximation of stochastic differential equations, *Comm. Pure Appl. Math.* **54** (2001), 1169–1214.

Received November 12, 2012; accepted September 12, 2013.