The Multilevel Monte Carlo Method for Stochastic Differential Equations driven by Jump-Diffusion Processes

Assyr Abdulle*, Adrian Blumenthal† and Evelyn Buckwar‡

Abstract

In this article we discuss the multilevel Monte Carlo method for stochastic differential equations driven by jump-diffusion processes. We show that for a reasonable jump intensity the multilevel Monte Carlo method for jump-diffusions reduces the computational complexity compared to the standard Monte Carlo method significantly for a given mean square accuracy. Carrying out numerical experiments on various examples, we compare the multilevel Monte Carlo method to standard Monte Carlo methods with and without variance reduction techniques. These experiments corroborate our theoretical findings and show for a sufficiently small mean square accuracy a significant reduction of the computational complexity of the multilevel Monte Carlo method compared to standard Monte Carlo methods.

Keywords: Multilevel Monte Carlo, Stochastic Differential Equations, Jump-Diffusion Processes, Complexity Theorem, Variance Reduction Techniques

AMS subject classification (2010): 65C05, 60H35, 91G60

1 Introduction

Monte Carlo methods are commonly applied when we are interested in computing expectations of functionals depending on a stochastic process. Here, we assume that the stochastic process is given by a stochastic differential equation (SDE) incorporating a jump term with a finite rate intensity. In a Monte Carlo (MC) approach, sample paths of the solution of an SDE are computed by a numerical integrator and the expected value of the given functional is approximated by the average over those samples. This procedure represents computing a statistical estimator of the desired quantity, and bias and statistical errors are introduced due to the numerical method and the approximation of the expectation, respectively. The bound on the statistical error for the MC method involves the inverse of the square root of the number of samples, as well as the variance of the process. The nature of this bound can not be changed, however many strategies to reduce the variance of the estimators and hence the complexity of the procedure have been proposed in the past few years. Among them, we mention variance reduction techniques such as estimators based on control variates or antithetic variates (see e.g. [14]).

*Department of Mathematics, Swiss Federal Institute of Technology Lausanne, Switzerland, e-mail: assyr.abdulle@epfl.ch.

†Department of Mathematics, Swiss Federal Institute of Technology Lausanne, Switzerland, e-mail: adrian.blumenthal@epfl.ch.

‡Institute for Stochastics, Johannes Kepler University Linz, Austria, e-mail: Evelyn.Buckwar@jku.at.

A recent approach, originating with Heinrich [16], proposed by Kebaier [20] as a statistical Romberg method with two levels and extend by Giles [12] to the so-called multilevel Monte Carlo (MLMC) method, allows to significantly speed up the classical MC method thanks to hierarchical sampling. By applying the Monte Carlo method for several nested time step sizes and choosing the right balance between the step sizes and the number of simulations at each level, it is possible to reduce the computational complexity of the Monte Carlo method for a given mean square accuracy. More precisely, to compute the expectation of functionals with an accuracy [I] of $\mathcal{O}(\varepsilon)$, the computational cost of $\mathcal{O}(\varepsilon^{-3})$ for the MC method is reduced to $\mathcal{O}(\varepsilon^{-2}(\log \varepsilon)^2)$ for the MLMC method. In this paper, we study the MLMC method for jump-diffusion processes. This class of processes becomes important for example in financial modeling, when stock prices based on diffusion processes should be modelled by taking into account sudden, unforeseeable events [1, 33]. Then, models based on jump-diffusion processes are required for more realistic modeling [29]. Furthermore, some physical processes cannot be modeled by continuous processes and need to take into account single events. We mention here the modeling of biological network dynamics [34] or chemical kinetics [13].
The MLMC method for jump-diffusions with finite rate activity has first been studied in [36, 35] and [2], whereas the case of Lévy processes with infinite rate activity has been studied in [8, 28]. In this paper we discuss the complexity theorem of the MLMC for jump-diffusion problems, in particular for the jump-adapted version of the Euler-Maruyama method. We study the MLMC for jump-diffusion problems on various examples and show significant speed-up compared to standard MC computations. The new approach is also compared to estimators based on variance reduction techniques (antithetic variates, control variates). The results show that for sufficiently small errors, the MLMC method always outperforms these other techniques for the models considered.
This paper is organised as follows. In Section 2 we discuss jump-diffusion processes and numerical methods used to approximate such processes. In Section 3 we construct the MLMC method for jump-diffusions and we give an extended version of the complexity theorem presented in [12]. Finally, in Section 4 we present some numerical experiments to illustrate our theoretical findings.

2 Preliminaries

Throughout this paper let $(\Omega, \mathcal{F}, \mathbb{P}, \mathcal{F}_t)$ be a filtered probability space where the filtration $(\mathcal{F}_t)_{t\geq 0}$ satisfies the usual condtions (see e.g. [27, pp. 4–6,12-13] or [32, pp. 1–2,49–51]). We use in this paper the terms *computational cost* and *computational complexity* synonymously to represent overall for an algorithm the number of time steps of a numerical discretisation, the number of samples generated and the number of function evaluations. These terms measure the complexity of algorithms and they will be used when we compare the performance of algorithms.
We consider stochastic processes $(S(t))_{t\in[0,T]}$ on the bounded interval $[0,T]$ described by the stochastic differential equation incorporating diffusion and jump terms

$$\begin{cases} \mathrm{d}S(t) &= a(t,S(t-))\mathrm{d}t + b(t,S(t-))\mathrm{d}W(t) + c(t,S(t-))\mathrm{d}J(t), \qquad 0 \leq t \leq T, \\ S(0) &= S_0\,, \end{cases} \tag{2.1}$$

[I] Here, the square root of the mean square error is chosen as a measure of the accuracy.

with $S(t-)$ denoting $S(t-) = \lim_{s \nearrow t} S(s)$. Here $(W(t))_{t\in[0,T]}$ is an $m-$dimensional Wiener process and $(J(t))_{t\in[0,T]}$ an $r-$dimensional compound Poisson process, $J(t) = (J^1(t), \ldots, J^r(t))$. Each component $J^k(t)$ is defined by

$$J^k(t) = \sum_{i=1}^{N^k(t)} (V_i^k - 1),$$

where $N^k(t)$ is a Poisson process with intensity λ_k and where the jump sizes are characterised by V_i^k. Further, the functions $a : [0,T] \times \mathbb{R}^d \to \mathbb{R}^d$, $b : [0,T] \times \mathbb{R}^d \to \mathbb{R}^{d\times m}$, and $c : [0,T] \times \mathbb{R}^d \to \mathbb{R}^{d\times r}$ represent the drift, the diffusion and the jump coefficient, respectively. We assume standard Lipschitz and linear growth conditions on a, b and c to ensure the existence of a strong solution of the SDE (2.1).

Remark 2.1. *Note that if γ_i^k corresponds to the i-th jump time of $J^k(t)$, then the jump size of this jump is given by*

$$S\left(\gamma_i^k\right) - S\left(\gamma_i^k-\right) = c^k\left(\gamma_i^k, S\left(\gamma_i^k-\right)\right)\left(V_i^k - 1\right).$$

If we now assume $c^k\left(\gamma_i^k, S\left(\gamma_i^k-\right)\right) = S\left(\gamma_i^k-\right)$, as this is the case for the models we consider in the following, then

$$S\left(\gamma_i^k\right) - S\left(\gamma_i^k-\right) = S\left(\gamma_i^k-\right)\left(V_i^k - 1\right),$$

and consequently

$$S\left(\gamma_i^k\right) = S\left(\gamma_i^k-\right) V_i^k,$$

which means V_i^k corresponds to the ratio of the stochastic process before and after the i-th jump of $J^k(t)$. Thus the choice of $V_i^k - 1$ above (see also e.g. [14, p.135]).

2.1 Numerical Schemes

In this section we recall two numerical schemes, the regular and the jump-adapted Euler method, to approximate solutions of the SDE (2.1). The former is iterated using a uniform step-size, whereas the latter includes the jump times into the otherwise uniform time grid. we discuss the strong and the weak convergence of these numerical schemes. For details and additional references on numerical methods for jump-diffusion problems see [4, 3, 10, 18, 25, 26, 31, 5].

Regular Euler Method. The first numerical method that we consider is a natural extension of the Euler-Maruyama method for SDEs driven by diffusion processes. The regular Euler method is defined by a uniform step size h and the grid is given by

$$\tau_h^{T/h} = \{\tau_0, \tau_1, \ldots, \tau_{T/h}\}, \tag{2.2}$$

where $\tau_j = jh \ \forall j \in \{0, 1, \ldots, T/h\}$. Now, let S_j be an approximation of the stochastic process $S(t)$ at $t = \tau_j$, i.e. $S_j \approx S(\tau_j)$. The regular Euler scheme is defined by

$$S_j = S_{j-1} + a(\tau_{j-1}, S_{j-1})(\tau_j - \tau_{j-1}) + b(\tau_{j-1}, S_{j-1})\Delta W_j + c(\tau_{j-1}, S_{j-1})\Delta J_j \tag{2.3}$$

with $j \in \{1, 2, \ldots, T/h\}$. By the initial condition, we have $S_0 = S(0)$ and the increments of the m-dimensional Wiener process and the r-dimensional compound Poisson process

are given by $\Delta W_j = W(\tau_j) - W(\tau_{j-1})$ and $\Delta J_j = J(\tau_j) - J(\tau_{j-1})$, respectively. Note that, by setting the jump coefficient c to zero, we get the Euler-Maruyama method, which can be used for numerical approximations of SDEs driven by diffusions (see e.g. [17]).

Jump-Adapted Euler Method. The second numerical method that we consider, is the jump-adapted Euler scheme. Unlike the previous scheme, this method does not have a uniform step size if there is at least one jump. In the case of the jump-adapted Euler scheme, the jump times have to be added to the regular grid with uniform step size h defined in (2.2). Recall that we consider a r-dimensional jump process, and thus, the number of jumps in the time interval $[0, T]$ are specified by the Poisson variables $N^1(T), N^2(T), \ldots, N^r(T)$ with intensities $\lambda_1 T, \lambda_2 T, \ldots, \lambda_r T$. Hence, the grid for the jump-adapted Euler scheme is given by

$$\{\tau_0, \tau_1, \ldots, \tau_{T/h}\} \cup \{\gamma_1^1, \gamma_2^1, \ldots, \gamma_{N^1(T)}^1\} \cup \cdots \cup \{\gamma_1^r, \gamma_2^r, \ldots, \gamma_{N^r(T)}^r\}, \tag{2.4}$$

where $\gamma_1^k, \gamma_2^k, \ldots, \gamma_{N^k(T)}^k$ are the jump times in the interval $[0, T]$ of the jump component $J^k(t)$. Thus, there are in total $T/h + \sum_{i=1}^r N^i(T)$ time steps. Adding the jump times to the regular grid and rearranging the grid such that the j-th entry τ_j corresponds to the j-th time step. The jump-adapted Euler scheme is identical to the regular method in (2.3) but with $j \in \{1, 2, \ldots, T/h + \sum_{i=1}^r N^i(T)\}$.

Remark 2.2. *To ease the notation we describe in the sequel both numerical schemes by (2.3) with $j \in \{1, 2, \ldots, G\}$, where $G = T/h + \sum_{i=1}^r N^i(T)$ in the jump-adapted case and $G = T/h$ in the regular case.*

Convergence of Numerical Methods. In this paper we consider two types of convergence: the strong convergence and the weak convergence, respectively, as given in the definition below.

Definition 2.3. *Let $S(t_j)$ be the exact solution of the SDE (2.1) at $t = t_j$ and let $(S_j)_{j \in \mathbb{N}}$ be the approximate solution by a numerical method at the same time point.*

1. *A numerical method is converging with a strong order of convergence γ_{strong} if*

$$\exists C \in \mathbb{R}_+ \text{ such that } \max_{0 \le j \le T/h} \left(\mathbb{E}\left[|S_j - S(\tau_j)|^2 \right] \right)^{1/2} \le C h^{\gamma_{strong}}, \tag{2.5}$$

   where $\tau_j = jh \in [0, T]$ and h is tending to 0.

2. *A numerical method is converging with weak order of convergence γ_{weak} if there exists $C \in \mathbb{R}_+$ such that for all functions p in a certain class (usually p satisfies smoothness and polynomial growth conditions) we have*

$$|\mathbb{E}[p(S_j)] - \mathbb{E}[p(S(\tau_j)]| \le C h^{\gamma_{weak}} \tag{2.6}$$

   for any $\tau_j = jh \in [0, T]$ fixed and h tending to 0.

Note that a possible class for the functions p is given by $\mathcal{C}_P^l(\mathbb{R})$. This class contains functions of the type $p : \mathbb{R} \to \mathbb{R}$ that are l times continuously differentiable and that, together with their partial derivatives up to order l, have polynomial growth (see e.g. [21, p.153]).

The regular Euler method (see e.g. [4, p.276 and pp.291-292] and [3, p.6]) and the jump-adapted Euler scheme (see e.g. [4, pp.350-351] or [3, p.9]) have, under appropriate conditions for the coefficient functions a, b and c, a strong convergence of order $1/2$. Under appropriate conditions for the drift function a, the diffusion function b, the jump function c, the initial condition S_0 and the jump intensities $\lambda_1, \lambda_2, \ldots, \lambda_r$, the regular Euler scheme (see e.g. [4, p.508 and pp.517-520] and [3, p.11]) as well as the jump-adapted Euler scheme (see e.g. [4, p.524] and [3, p.12]) have a weak convergence of order 1.

3 Multilevel Monte Carlo Method for Jump-Diffusions

In this section we generalise the multilevel Monte Carlo method to stochastic differential equations driven by jump-diffusions. We present here the construction of the Monte Carlo method and the multilevel Monte Carlo method for jump-diffusions. Furthermore, we state and prove the corresponding complexity theorem. Consider the jump-diffusion process $(S(t))_{t\in[0,T]}$ solution of the SDE (2.1) and a numerical approximation (e.g., the Euler method or the jump-adapted Euler method previously introduced). For a Lipschitz continuous function $f : \mathbb{R}^n \to \mathbb{R}$ we want to estimate the expectation $\mathbb{E}\left[f\left(S(T)\right)\right]$ from many realisations of the numerical solution of (2.1).
For simplicity of the presentation we describe in the sections 3.1 and 3.2 the Monte Carlo and the multilevel Monte Carlo method for one-dimensional jump processes with intensity $\lambda = 1$. We emphasise that our complexity theorem will be presented for the general case of a r-dimensional jump process.

3.1 Monte Carlo Method for Jump-Diffusions

We recall the standard Monte Carlo estimator, which is given by

$$\mathbb{E}\left[f(S(T))\right] \approx \frac{1}{N}\sum_{i=1}^{N} f\left(S_G^{(i)}\right) =: \widehat{Y}, \tag{3.1}$$

where we take a sample average over N independent paths with S_G a numerical approximation of $S(t)$ at the time end point T (see Section 2.1).

Remark 3.1. *Note that h, as defined in Section 2.1, is the uniform time step size of the regular Euler scheme. For the jump-adapted Euler scheme, the jump times are added to the regular grid. Hence, if there is at least one jump, the grid is not regular any more. However, in that case h corresponds to the maximum step size, i.e.*

$$h = \max_{j\in\{1,2,\ldots,G\}} \left(\tau_j - \tau_{j-1}\right).$$

Applying the Monte Carlo method, two types of error arise (see e.g. [30, p.137]). Firstly, there is an error due to the numerical approximation of $S(t)$. This error introduces a bias. In fact, we approximate the stochastic process $(S(t))_{t\in[0,T]}$ at $t = T$ using a numerical scheme, so that $S(T) \approx S_G$. Evaluating the function f and taking the expectation on both sides leads to

$$\mathbb{E}\left[f\left(S(T)\right)\right] \approx \mathbb{E}\left[f\left(S_G\right)\right].$$

By linearity of the expectation and the fact that the samples $S_G^{(i)}$ are identically distributed, we have

$$\mathbb{E}\left[\widehat{Y}\right] = \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N} f\left(S_G^{(i)}\right)\right] = \mathbb{E}\left[f\left(S_G\right)\right].$$

Hence, we obtain

$$\text{bias}\left(\widehat{Y}\right) = \mathbb{E}\left[\widehat{Y}\right] - \mathbb{E}\left[f\left(S(T)\right)\right] = \mathbb{E}\left[f\left(S_G\right)\right] - \mathbb{E}\left[f\left(S(T)\right)\right] = \mathcal{O}\left(h\right), \tag{3.2}$$

where we have used the first order weak convergence of the numerical schemes (see Section 2.1) for the last equality.
Secondly, there is an error arising from the estimation of the expectation. The expectation, which is an integral, is approximated by taking the sample average over N simulations. Due to the strong law of large numbers (see e.g. [9, pp.77-78]) and the central limit theorem (see e.g. [6, p.233]), this approximation is almost surely unbiased. However, there is a certain variance that depends on the number of simulations N (see e.g. [11, p.231]). Indeed, for the variance of the estimator we have

$$\text{Var}\left(\widehat{Y}\right) = \frac{1}{N^2}\sum_{i=1}^{N}\text{Var}\left(f\left(S_G^{(i)}\right)\right) = \frac{\text{Var}\left(f\left(S_G^{(1)}\right)\right)}{N} = \mathcal{O}\left(N^{-1}\right), \tag{3.3}$$

where we have used first the independence of $S_G^{(1)}, S_G^{(2)}, \ldots, S_G^{(N)}$ and then the fact that they are identically distributed.
One way to describe the trade-off between the bias and the variance is given by the mean square error, which can be decomposed as

$$\text{MSE}\left(\widehat{Y}\right) = \mathbb{E}\left[\left(\widehat{Y} - \mathbb{E}\left[f(S(T))\right]\right)^2\right] = \text{Var}\left(\widehat{Y}\right) + \left(\text{bias}\left(\widehat{Y}\right)\right)^2. \tag{3.4}$$

By (3.2) and (3.3), we get

$$\text{MSE}\left(\widehat{Y}\right) = \mathcal{O}(N^{-1}) + \left(\mathcal{O}\left(h\right)\right)^2 = \mathcal{O}\left(N^{-1} + h^2\right). \tag{3.5}$$

In other words, for N large enough and h sufficiently small, there exists two constants C_1 and C_2 such that

$$\text{MSE}\left(\widehat{Y}\right) \approx C_1 N^{-1} + C_2 h^2.$$

Now, let ε, a positive constant, be the desired mean square accuracy in the sense that

$$\text{MSE}\left(\widehat{Y}\right) = \mathcal{O}\left(\varepsilon^2\right).$$

To achieve such an accuracy, one requires $N = \mathcal{O}\left(\varepsilon^{-2}\right)$ simulations and a regular step size $h = \mathcal{O}\left(\varepsilon\right)$. We have to distinguish now between the two numerical schemes. The regular Euler method has T/h steps, which is proportional to $h^{-1} = \mathcal{O}\left(\varepsilon^{-1}\right)$, and thus, a computational complexity of $\mathcal{O}\left(\varepsilon^{-3}\right)$ is required for the regular scheme. In the jump-adapted case, the jump times are added to the regular grid. The number of jumps over the time interval $[0, T]$ is given by the random variable $N(T)$ which follows a Poisson distribution with intensity λT. The expected number of jumps is given by $\mathbb{E}\left[N(T)\right] = \lambda T$. Therefore, for the jump-adapted scheme, there are $T/h + \lambda T = \mathcal{O}\left(\varepsilon^{-1} + \lambda\right)$ steps, and thus, the computational complexity amounts to $\mathcal{O}\left(\varepsilon^{-2}\left(\frac{1}{\varepsilon} + \lambda\right)\right)$.

Summing up the results for the standard Monte Carlo method for jump-diffusions, to achieve a mean square error of order $\mathcal{O}(\varepsilon^2)$, the regular Euler approach requires a computational cost of $\mathcal{O}(\varepsilon^{-3})$. For the jump-adapted Euler approach, a computational cost of $\mathcal{O}\left(\varepsilon^{-2}\left(\frac{1}{\varepsilon}+\lambda\right)\right)$ is necessary. Note that, by setting the jump intensity λ to zero, we reproduce the result for diffusion processes in [12].

3.2 Multilevel Monte Carlo Method for Jump-Diffusions

The idea of the multilevel Monte Carlo method [13] is to apply the Monte Carlo method for several nested levels of time step sizes and to compute different numbers of paths on each level, from a few paths when the time stept size is small to many paths when the step size is large. By choosing the right balance between the step sizes and the number of simulated trajectories at each level it is possible to reduce the computational complexity compared to that of the standard Monte Carlo method for a given mean square accuracy.
We introduce now the multilevel Monte Carlo method for stochastic differential equations driven by jump-diffusions. Fix a positive number T as the time end point, an integer $M \geq 2$ as the refinement factor and an integer L as the total number of levels. Define the uniform nested time step sizes

$$h_l = \frac{T}{M^l}, \;\; l = 0, 1, \ldots, L.$$

Furthermore, we fix an m-dimensional Wiener process $(W(t))_{t\in[0,T]}$ and a one-dimensional compound Poisson process $(J(t))_{t\in[0,T]}$. Let P denote the payoff function (e.g., $P = f(S(T))$) and approximate P by P_l, where $P_l = f(S_{M^l})$ is an approximation of P based on the numerical discretisation of $S(t)$ with a regular step size h_l. Applying the telescopic sum, we can write

$$\begin{aligned} P_L &= P_0 + P_1 - P_0 + P_2 - P_1 \pm \ldots + P_{L-1} - P_{L-2} + P_L - P_{L-1} \\ &= P_0 + \sum_{l=1}^{L} (P_l - P_{l-1}). \end{aligned}$$

Taking the expectation on both sides and using the linearity of the expectation we obtain

$$\mathbb{E}[P_L] = \mathbb{E}[P_0] + \sum_{l=1}^{L} \mathbb{E}[P_l - P_{l-1}]. \tag{3.6}$$

The idea of the multilevel Monte Carlo method is to approximate each term on the right-hand side independently. In fact for the first term we have

$$\mathbb{E}[P_0] \approx \frac{1}{N_0} \sum_{i=1}^{N_0} P_0^{(i)} =: \widehat{Y}_0,$$

where we take the average over N_0 independent samples. The other terms are estimated using N_l independent samples such that

$$\mathbb{E}[P_l - P_{l-1}] \approx \frac{1}{N_l} \sum_{i=1}^{N_l} \left(P_l^{(i)} - P_{l-1}^{(i)}\right) =: \widehat{Y}_l,$$

for $l \in \{1, 2, \ldots, L\}$. We emphasise that the estimates $P_l^{(i)}$ and $P_{l-1}^{(i)}$ are based on the same jump-diffusion path, i.e., the same Brownian motion path and also on the same sample path of the compound Poisson process. Therefore the estimator for the MLMC method is given by

$$\mathbb{E}\left[P_L\right] \approx \frac{1}{N_0}\sum_{i=1}^{N_0} P_0^{(i)} + \sum_{l=1}^{L}\frac{1}{N_l}\sum_{i=1}^{N_l}\left(P_l^{(i)} - P_{l-1}^{(i)}\right) = \sum_{l=0}^{L}\widehat{Y}_l =: \widehat{Y}. \tag{3.7}$$

Next we derive the variance and the computational cost for the MLMC estimator $\widehat{Y}$. Firstly we point out that in the partial estimator $\widehat{Y}_l = \frac{1}{N_l}\sum_{i=1}^{N_l}\left(P_l^{(i)} - P_{l-1}^{(i)}\right)$ each term in the sum is produced from a jump-diffusion process that is independent of the jump-diffusion processes used for the other summands. Using this independence combined with the fact that the jump-diffusion processes are identically distributed and denoting the variance of a single sample of $P_l^{(i)} - P_{l-1}^{(i)}$ by V_l, the variance of the partial estimator $\widehat{Y}_l$ is given by

$$\mathrm{Var}\left(\widehat{Y}_l\right) = \mathrm{Var}\left(\frac{1}{N_l}\sum_{i=1}^{N_l}\left(P_l^{(i)} - P_{l-1}^{(i)}\right)\right) = \frac{1}{N_l^2}\sum_{i=1}^{N_l}\underbrace{\mathrm{Var}\left(P_l^{(i)} - P_{l-1}^{(i)}\right)}_{=:V_l} = \frac{V_l}{N_l}.$$

Thus the variance of the combined estimator $\widehat{Y}$ is given by

$$\mathrm{Var}\left(\widehat{Y}\right) = \mathrm{Var}\left(\sum_{l=0}^{L}\widehat{Y}_l\right) = \sum_{l=0}^{L}\mathrm{Var}\left(\widehat{Y}_l\right) = \sum_{l=0}^{L}\frac{V_l}{N_l}. \tag{3.8}$$

Note that we have used the independence of the partial estimators $\widehat{Y}_l$ resulting from the independence of the jump-diffusion processes.
Secondly, concerning the computational complexity of $\widehat{Y}$, at each level l there are N_l Monte Carlo simulations required to approximate the expectation. Furthermore, d numerical discretisations (one for each component of the d-dimensional SDE (2.1)) are carried out with a regular step size h_l. For the jump-adapted Euler scheme, the expected number of jumps λT of each discretisation has to be added to the number of steps resulting from the regular grid, $\frac{T}{h_l}$. Hence, there are $d\left(\frac{T}{h_l} + \lambda T\right)$ steps necessary in the jump-adapted case. Considering the evaluation of the function f as a single operation, the computational cost is given by

$$\mathrm{Cost}\left(\widehat{Y}\right) = \sum_{l=0}^{L} 2N_l d\left(\frac{T}{h_l} + \lambda T\right) = \sum_{l=0}^{L} 2N_l d\frac{T}{\left(\frac{h_l}{1+\lambda h_l}\right)}.$$

For the regular Euler scheme, the jumps do not affect the time grid, and thus there are $\frac{T}{h_l}$ steps. The computational cost, given by $\mathrm{Cost}\left(\widehat{Y}\right) = \sum_{l=0}^{L} 2N_l d\frac{T}{h_l}$, is the same as for diffusion problems. Taking also into account that the order of weak and strong convergence are the same as for the Euler-Maruyama method, the construction for regular Euler is identical to the one in the diffusion case. Therefore we concentrate in the following on the jump-adapted Euler approach.
Finally, fixing a positive constant D as the fixed computational budget, we can set up an optimisation problem, which minimises the variance of $\widehat{Y}$ for a fixed computational

complexity:

$$\begin{cases} \text{minimise} \;\; \mathrm{Var}\left(\widehat{Y}\right) = \displaystyle\sum_{l=0}^{L} \frac{V_l}{N_l} \\ \\ \text{subject to} \;\; \mathrm{Cost}\left(\widehat{Y}\right) = \displaystyle\sum_{l=0}^{L} 2N_l d \frac{T}{\left(\frac{h_l}{1+\lambda h_l}\right)} = D \,, \end{cases} \tag{3.9}$$

where we want to find a solution with respect to the positive variables $N_l > 0$, with $l \in \{0, 1, \ldots, L\}$. The optimisation problem (3.9) can be solved by standard methods (e.g., using Lagrange multipliers). The following proposition holds.

Proposition 3.2. *The solution to the optimisation problem for the continuous variables $N_0, N_1, \ldots, N_L$ given in (3.9) satisfies for $l \in \{0, 1, \ldots, L\}$*

$$N_l = \frac{D}{2dT} \frac{\sqrt{V_l \frac{h_l}{1+\lambda h_l}}}{\displaystyle\sum_{k=0}^{L} \sqrt{\frac{V_k}{\frac{h_k}{1+\lambda h_k}}}} \,. \tag{3.10}$$

Now we show that $V_l = \mathcal{O}(h_l)$. Note that since the jump-diffusion processes are i.i.d., we work in the following with the notation P_l instead of $P_l^{(i)}$. The strong order of convergence 1/2 of the jump-adapter Euler method (see Section 2.1) yields

$$\mathbb{E}\left[|S_{M^l} - S(T)|^2\right] = \mathcal{O}(h_l) \quad \text{as} \;\; l \to \infty. \tag{3.11}$$

For the variance of one single sample we have

$$V_l = \mathrm{Var}(P_l - P_{l-1}) \leq \left(\mathrm{Var}(P_l - P)^{1/2} + \mathrm{Var}(P_{l-1} - P)^{1/2}\right)^2 ,$$

where we have used the Cauchy-Schwarz inequality.
Furthermore, using the property of f being Lipschitz continuous, we have

$$\begin{aligned} \mathrm{Var}(P_l - P) &\leq \mathbb{E}\left[(P_l - P)^2\right] \\ &= \mathbb{E}\left[\left(f(S_{M^l}) - f(S)\right)^2\right] \\ &\leq C\mathbb{E}\left[|S_{M^l} - S|^2\right] , \end{aligned}$$

where $C \in \mathbb{R}_+$. Thus, by (3.11),

$$\mathrm{Var}\left(P_l - P\right) = \mathcal{O}(h_l) \quad \text{as} \;\; l \to \infty.$$

Note that we also have $\mathrm{Var}\left(P_{l-1} - P\right) = \mathcal{O}(h_l)$, as $h_{l-1} = Mh_l$ with M being constant. We thus have $V_l = \mathcal{O}(h_l)$ and in other words, for h_l sufficiently small, there is a positive constant K such that $V_l = Kh_l$.
We can then write the number of simulations per level as

$$N_l = \frac{\frac{D}{2dT}\sqrt{Kh_l \frac{h_l}{1+\lambda h_l}}}{\displaystyle\sum_{k=0}^{L} \sqrt{\frac{Kh_k}{\frac{h_k}{1+\lambda h_k}}}} = \frac{D}{2dT} \frac{h_l \sqrt{\frac{1}{1+\lambda h_l}}}{\displaystyle\sum_{k=0}^{L} \sqrt{1+\lambda h_k}}. \tag{3.12}$$

Combining this with (3.8), we obtain for the variance of the MLMC estimator

$$\mathrm{Var}\left(\widehat{Y}\right) = \sum_{l=0}^{L} \frac{V_l}{N_l} = \sum_{l=0}^{L} \frac{Kh_l}{\frac{D}{2dT}\frac{h_l\sqrt{\frac{1}{1+\lambda h_l}}}{\sum_{k=0}^{L}\sqrt{1+\lambda h_k}}} = \frac{2dTK}{D}\left(\sum_{l=0}^{L}\sqrt{1+\lambda h_l}\right)^2. \tag{3.13}$$

Note that by increasing the computational budget the variance can be made as small as desired. However, usually the computational budget is limited. Thus we fix now a mean square accuracy of $\mathcal{O}\left(\varepsilon^2\right)$ and we determine the corresponding computational budget D and the number of levels L.
We consider now the mean square error (3.4). For the bias we have

$$\mathrm{bias}\left(\widehat{Y}\right) = \mathbb{E}\left[\widehat{Y}\right] - \mathbb{E}[P] = \mathbb{E}\left[P_L\right] - \mathbb{E}[P] = \mathbb{E}\left[P_L - P\right],$$

where we have used the linearity of the expectation. Using the first order weak convergence of the jump-adapted Euler scheme (see Section 2.1), we obtain

$$\mathrm{bias}\left(\widehat{Y}\right) = \mathbb{E}\left[P_L - P\right] = \mathcal{O}\left(h_L\right). \tag{3.14}$$

Hence, to achieve a mean square error of $\mathrm{MSE}\left(\widehat{Y}\right) = \mathcal{O}\left(\varepsilon^2\right)$, we require in particular that the bias satisfies $\mathrm{bias}\left(\widehat{Y}\right) = \mathcal{O}\left(\varepsilon\right)$, and thus,

$$h_L = \frac{T}{M^L} = \widetilde{K}\varepsilon,$$

for L large enough and where $\widetilde{K}$ is a positive constant. Rearranging terms and taking the natural logarithm we obtain $L = \frac{1}{\log M}\left(\log T + \log \widetilde{K}^{-1} + \log \varepsilon^{-1}\right)$. Hence, the number of levels L satisfies

$$L = \frac{\log \varepsilon^{-1}}{\log M} + \mathcal{O}\left(1\right). \tag{3.15}$$

This shows us how to choose L. Finally, to achieve a mean square accuracy of $\mathcal{O}\left(\varepsilon^2\right)$ we also require $\mathrm{Var}\left(\widehat{Y}\right) = \mathcal{O}\left(\varepsilon^2\right)$. Considering (3.13), this is equivalent to

$$\frac{2dTK}{D}\left(\sum_{l=0}^{L}\sqrt{1+\lambda h_l}\right)^2 = \widehat{K}\varepsilon^2,$$

where $\widehat{K}$ is a positive constant. Rearranging terms and taking an upper bound, we get

$$\begin{aligned} D &= 2dTK\widehat{K}^{-1}\varepsilon^{-2}\left(\sum_{l=0}^{L}\sqrt{1+\lambda h_l}\right)^2 \\ &\leq 2dTK\widehat{K}^{-1}\varepsilon^{-2}\left(\sqrt{1+\lambda T}\sum_{l=0}^{L}1\right)^2 \\ &= 2dTK\widehat{K}^{-1}\varepsilon^{-2}\left(1+\lambda T\right)\left(L+1\right)^2. \end{aligned} \tag{3.16}$$

Hence, the computational budget satisfies

$$D = \mathcal{O}\left(d\varepsilon^{-2}(\log\varepsilon)^2(1+\lambda T)\right),$$

where we have used the result for L in (3.15).
Therefore, considering the multilevel Monte Carlo method for jump-diffusions and using the jump-adapted Euler method, to achieve a mean square error of

$$\mathrm{MSE}\left(\widehat{Y}\right) = \mathcal{O}\left(\varepsilon^2\right)$$

a computational complexity of

$$\mathrm{Cost}\left(\widehat{Y}\right) = \mathcal{O}\left(d\varepsilon^{-2}(\log\varepsilon)^2(1+\lambda T)\right)$$

is necessary.

Remark 3.3. *Note that when considering the jump-adapted method instead of the regular one, the additional term $(1+\lambda T)$ appears in the computational cost. Usually the time end point T is fixed and thus the parameter of interest is the jump intensity λ. Observe that by setting λ to zero, we produce the results for the MLMC method for multi-dimensional diffusions. Hence, the approach in this paper delivers a natural extension of the MLMC method to jump-diffusion processes.*

Remark 3.4. *As mentioned earlier, the construction of the MLMC method for jump-diffusions is based on a one-dimensional jump process. In a more general approach, where jumps are driven by an r-dimensional compound Poisson process, the computational cost for a fixed mean square accuracy of $MSE\left(\widehat{Y}\right) = \mathcal{O}\left(\varepsilon^2\right)$ is given by*

$$Cost\left(\widehat{Y}\right) = \mathcal{O}\left(d\varepsilon^{-2}(\log\varepsilon)^2\left(1+\sum_{i=1}^{r}\lambda_i T\right)\right).$$

3.3 Complexity Theorem for Jump-Diffusions

In this section we give an extended version (to jump-diffusions) of the complexity theorem given in [12] for diffusion processes. In particular the jump-adapted version of the Euler-Maruyama method is considered. The proof largely follows the original proof of Giles [12]. For the sake of completeness we present the details of the arguments.

Theorem 3.5 (Complexity Theorem for Jump-Adapted Schemes)**.** *Fix two positive integers T and M such that $M \geq 2$ and let $\lambda_1, \lambda_2, \ldots, \lambda_r$ be r positive numbers. Let $(S(t))_{t\in[0,T]} \subset \mathbb{R}^d$ be a solution to the stochastic differential equation (2.1). Let P be a functional of $S(t)$. Denote by P_l an approximation of P using a jump-adapted numerical approximation with a regular time step size $h_l = T/M^l$.*
Suppose that there exist independent estimators $\widehat{Y}_l$ (based on N_l Monte Carlo simulations) and that there exist positive constants $\alpha \geq 1/2$, $\beta > 0$, $c_i > 0$ ($i \in \{1,2,3\}$) such that:

(i) $\mathbb{E}\left[P_l - P\right] \leq c_1 h_l^{\alpha}$,

(ii) $\mathbb{E}\left[\widehat{Y}_l\right] = \begin{cases} \mathbb{E}\left[P_0\right], & l = 0 \\ \mathbb{E}\left[P_l - P_{l-1}\right], & l > 0 \end{cases}$,

(iii) $Var\left(\widehat{Y}_l\right) \leq c_2 \frac{h_l^\beta}{N_l}$,

(iv) $Cost\left(\widehat{Y}_l\right) \leq c_3 d \frac{N_l}{h_l}(1 + \tilde{\lambda} h_l)$,

where $\tilde{\lambda} = \sum_{i=1}^{r} \lambda_i$. *If Conditions (i)-(iv) hold, then there exists a positive constant* c_4 *such that for all* $\varepsilon < \frac{1}{e}$ *there exist positive integers* $L \in \mathbb{N}_*$ *and* $N_l \in \mathbb{N}_*$ *such that the combined estimator of* $\mathbb{E}[P]$,

$$\widehat{Y} = \sum_{l=0}^{L} \widehat{Y}_l,$$

has a mean square error that is bounded by

$$MSE\left(\widehat{Y}\right) = \mathbb{E}\left[\left(\widehat{Y} - \mathbb{E}[P]\right)^2\right] \leq \varepsilon^2$$

with a computational complexity bounded by

$$Cost\left(\widehat{Y}\right) \leq \begin{cases} c_4 d \varepsilon^{-2}(1 + \tilde{\lambda} T), & \beta > 1, \\ c_4 d \varepsilon^{-2} (\log \varepsilon)^2 (1 + \tilde{\lambda} T), & \beta = 1, \\ c_4 d \varepsilon^{-2-(1-\beta)/\alpha}(1 + \tilde{\lambda} T), & 0 < \beta < 1, \end{cases}$$

where the logarithm is taken with the natural basis.

An immediate consequence of the theorem above is the result for the regular Euler-Maruyama scheme presented in the following corollary, which also holds for multi-dimensional diffusion problems (the proof of the corollary can also be obtained by following [12]).

Corollary 3.6 (Regular Schemes)**.** *Suppose we use a regular scheme for the numerical approximation. Subject to the assumptions (i)-(iii) as in Theorem 3.5, and replacing (iv) with* $Cost\left(\widehat{Y}_l\right) \leq c_3 d \frac{N_l}{h_l}$, *the mean square error is bounded by* $MSE\left(\widehat{Y}\right) \leq \varepsilon^2$ *and the bound for the computational cost is characterised by*

$$Cost\left(\widehat{Y}\right) \leq \begin{cases} c_4 d \varepsilon^{-2}, & \beta > 1, \\ c_4 d \varepsilon^{-2} (\log \varepsilon)^2, & \beta = 1, \\ c_4 d \varepsilon^{-2-(1-\beta)/\alpha}, & 0 < \beta < 1. \end{cases}$$

Proof of Theorem 3.5. Throughout this proof the notation $\lceil x \rceil$ is used for rounding up the real number x to the next higher integer. First, let $\varepsilon < \frac{1}{e}$ and we choose the total number of levels L to be equal to

$$L = \left\lceil \frac{\log\left(\sqrt{2} c_1 T^\alpha \varepsilon^{-1}\right)}{\alpha \log M} \right\rceil. \tag{3.17}$$

In the first part we prove that the squared bias of $\widehat{Y}$ is bounded above by $\frac{\varepsilon^2}{2}$. In the second part we prove that the variance of $\widehat{Y}$ has an upper bound given by $\frac{\varepsilon^2}{2}$. Combining the results of the two parts leads to an upper bound of

$$\mathrm{MSE}\left(\widehat{Y}\right) = \mathrm{Var}\left(\widehat{Y}\right) + \left(\mathrm{bias}\left(\widehat{Y}\right)\right)^2 \leq \frac{\varepsilon^2}{2} + \frac{\varepsilon^2}{2} = \varepsilon^2. \tag{3.18}$$

(a) Estimation of $\left(\textit{bias}\left(\widehat{Y}\right)\right)^2$

Using L as defined in (3.17), the following inequalities can be established:

$$\begin{aligned} \frac{\log\left(\sqrt{2}c_1T^\alpha\varepsilon^{-1}\right)}{\alpha\log M} &\leq \quad L \quad < \frac{\log\left(\sqrt{2}c_1T^\alpha\varepsilon^{-1}\right)}{\alpha\log M}+1 \\ \iff \qquad \sqrt{2}c_1\varepsilon^{-1} &\leq \underbrace{\left(\frac{M^L}{T}\right)^\alpha}_{=h_L^{-\alpha}} < \sqrt{2}c_1\varepsilon^{-1}M^\alpha. \end{aligned}$$

Therefore we get the inequalities

$$\frac{M^{-\alpha}\varepsilon}{\sqrt{2}} < c_1h_L^\alpha \leq \frac{\varepsilon}{\sqrt{2}}. \tag{3.19}$$

Recall that $\widehat{Y}$ is an estimator of $\mathbb{E}[P]$, and thus the bias of the combined estimator is given by

$$\text{bias}\left(\widehat{Y}\right) = \mathbb{E}\left[\widehat{Y}\right] - \mathbb{E}[P].$$

Taking into account the linearity of the expectation and condition (ii), we have

$$\mathbb{E}\left[\widehat{Y}\right] = \mathbb{E}\left[\sum_{l=0}^{L}\widehat{Y}_l\right] = \sum_{l=0}^{L}\mathbb{E}\left[\widehat{Y}_l\right] = \mathbb{E}[P_0] + \sum_{l=1}^{L}\mathbb{E}[P_l - P_{l-1}] = \mathbb{E}[P_L].$$

Using in addition Condition (i), the squared bias satisfies $\left(\text{bias}\left(\widehat{Y}\right)\right)^2 \leq c_1^2h_L^{2\alpha}$, and taking into account the upper bound of (3.19), we obtain

$$\left(\text{bias}\left(\widehat{Y}\right)\right)^2 \leq \frac{\varepsilon^2}{2}.$$

(b) Estimation of $\textit{Var}\left(\widehat{Y}\right)$

Now it remains to show that the variance of $\widehat{Y}$ is bounded by $\frac{\varepsilon^2}{2}$. First we establish the inequality

$$\sum_{l=0}^{L}h_l^{-1} < \frac{M^2}{M-1}\left(\sqrt{2}c_1\right)^{1/\alpha}\varepsilon^{-2}. \tag{3.20}$$

In fact we have by the definition of the time step size h_l

$$\sum_{l=0}^{L}h_l^{-1} = \sum_{l=0}^{L}\left(\frac{T}{M^l}\right)^{-1} = h_L^{-1}\sum_{l=0}^{L}M^{l-L} < h_L^{-1}\frac{M}{M-1}.$$

In addition, by the lower limit of (3.19), we have $h_L^{-1} < M\left(\frac{\varepsilon}{\sqrt{2}c_1}\right)^{-1/\alpha}$, and thus

$$\sum_{l=0}^{L}h_l^{-1} < \frac{M^2}{M-1}\left(\sqrt{2}c_1\right)^{1/\alpha}\varepsilon^{-1/\alpha}.$$

Finally, since by assumption $\varepsilon < 1$ the following inequalities are equivalent:

$$\varepsilon^{-1/\alpha} \leq \varepsilon^{-2} \Leftrightarrow \alpha \geq \frac{1}{2}.$$

As by assumption $\alpha \geq \frac{1}{2}$ we obtain (3.20).
To pursue the proof we need to distinguish three cases.

Case 1: $\beta = 1$**.** Inspired by (3.12), we set the number of samples at level l to

$$N_l = \left\lceil 2\varepsilon^{-2}(L+1)c_2 \frac{h_l}{\sqrt{1+\tilde{\lambda}h_l}} \sqrt{1+\tilde{\lambda}T} \right\rceil . \tag{3.21}$$

Now, by considering first the independence of the partial estimators $\widehat{Y}_l$, followed by Condition (iii) and then the definition of N_l in (3.21), we have for the variance an upper bound given by $\frac{\varepsilon^2}{2}$.

Indeed, using $\mathrm{Var}\left(\widehat{Y}\right) = \mathrm{Var}\left(\sum_{l=0}^{L} \widehat{Y}_l\right) = \sum_{l=0}^{L} \mathrm{Var}\left(\widehat{Y}_l\right)$, we have

$$\begin{aligned}
\sum_{l=0}^{L} \mathrm{Var}\left(\widehat{Y}_l\right) &\leq \sum_{l=0}^{L} c_2 N_l^{-1} h_l \\
&\leq \sum_{l=0}^{L} c_2 h_l \left(\frac{\varepsilon^2}{2} (L+1)^{-1} c_2^{-1} \frac{\sqrt{1+\tilde{\lambda}h_l}}{h_l} \frac{1}{\sqrt{1+\tilde{\lambda}T}} \right) \\
&= \tfrac{\varepsilon^2}{2}(L+1)^{-1} \tfrac{1}{\sqrt{1+\tilde{\lambda}T}} \sum_{l=0}^{L} \sqrt{1+\tilde{\lambda}h_l}, \quad \leq \quad \frac{\varepsilon^2}{2},
\end{aligned}$$

where we used $\sum_{l=0}^{L} \sqrt{1+\tilde{\lambda}h_l} \leq \sqrt{1+\tilde{\lambda}T}(L+1)$. Therefore, in the case of $\beta = 1$, the mean square error of $\widehat{Y}$ is bounded by ε^2, i.e., (3.18) is satisfied. We derive now an upper bound for the computational complexity of the combined estimator in the case of $\beta = 1$. The idea is to bound first N_l and then to use condition (iv). By definition of N_l (see (3.21)), we have in particular

$$N_l < 2\varepsilon^{-2}(L+1)c_2 \frac{h_l}{\sqrt{1+\tilde{\lambda}h_l}} \sqrt{1+\tilde{\lambda}T} + 1. \tag{3.22}$$

We aim to find an upper bound for $L+1$. By definition of L (see (3.17)), the number of levels is bounded above by

$$L \quad < \quad \frac{\log\left(\sqrt{2}c_1 T^{\alpha} \varepsilon^{-1}\right)}{\alpha \log M} + 1 \quad = \quad \frac{\log \varepsilon^{-1}}{\alpha \log M} + \frac{\log\left(\sqrt{2}c_1 T^{\alpha}\right)}{\alpha \log M} + 1.$$

Furthermore we notice that $\varepsilon < \frac{1}{e} \Leftrightarrow e < \varepsilon^{-1} \Leftrightarrow 1 < \log \varepsilon^{-1}$. Thus we get

$$L + 1 \quad < \quad \frac{\log \varepsilon^{-1}}{\alpha \log M} + \frac{\log\left(\sqrt{2}c_1 T^{\alpha}\right)}{\alpha \log M} + 2 \quad \leq \quad c_5 \log \varepsilon^{-1}, \tag{3.23}$$

where $c_5 = \frac{1}{\alpha \log M} + \max\left(0, \frac{\log\left(\sqrt{2}c_1 T^{\alpha}\right)}{\alpha \log M}\right) + 2$. Using first condition (iv), followed by the upper bound (3.22) of N_l and the inequality (3.20) as well as the bound (3.23) for $L+1$, we end up with the computational complexity for $\mathrm{Cost}\left(\widehat{Y}\right) =$

$\sum_{l=0}^{L} \text{Cost}\left(\widehat{Y}_l\right)$ bounded by

$$\begin{aligned}
\text{Cost}\left(\widehat{Y}\right) &\leq \sum_{l=0}^{L} c_3 d N_l \frac{1+\tilde{\lambda}h_l}{h_l} \\
&< \sum_{l=0}^{L} c_3 d \frac{1+\tilde{\lambda}h_l}{h_l} \left(2\varepsilon^{-2}(L+1)c_2 \frac{h_l}{\sqrt{1+\tilde{\lambda}h_l}} \sqrt{1+\tilde{\lambda}T} + 1 \right) \\
&= c_3 d 2\varepsilon^{-2}(L+1)c_2 \sqrt{1+\tilde{\lambda}T} \sum_{l=0}^{L} \sqrt{1+\tilde{\lambda}h_l} + c_3 d \sum_{l=0}^{L} \frac{1+\tilde{\lambda}h_l}{h_l} \\
&\leq c_3 d 2\varepsilon^{-2}(L+1)^2 c_2 (1+\tilde{\lambda}T) + c_3 d (1+\tilde{\lambda}T) \sum_{l=0}^{L} h_l^{-1} \\
&\leq (1+\tilde{\lambda}T) d \varepsilon^{-2} (\log \varepsilon)^2 \left[c_3 2 c_5^2 c_2 + c_3 \tfrac{M^2}{M-1} \left(\sqrt{2} c_1 \right)^{1/\alpha} \right].
\end{aligned}$$

Hence an upper bound for the computational cost is given by

$$\text{Cost}\left(\widehat{Y}\right) \leq c_4 d \varepsilon^{-2} (\log \varepsilon)^2 (1+\tilde{\lambda}T),$$

where $c_4 = 2c_3 c_5^2 c_2 + c_3 \frac{M^2}{M-1} \left(\sqrt{2} c_1 \right)^{1/\alpha}$.

Case 2: $\beta > 1$**.** The number of simulations at level l is chosen such that

$$N_l = \left\lceil 2\varepsilon^{-2} c_2 T^{(\beta-1)/2} \left(1 - M^{-(\beta-1)/2}\right)^{-1} \frac{h_l^{(\beta+1)/2}}{\sqrt{1+\tilde{\lambda}h_l}} \sqrt{1+\tilde{\lambda}T} \right\rceil. \tag{3.24}$$

Observe that the choice of N_l is the same as in the first case if the parameter β is fixed at $\beta = 1$. In the following the derivation of the upper bound of the variance of the combined estimator is very similar as in the case with $\beta = 1$. Due to the independence of the partial estimators $\widehat{Y}_l$, condition (iii) and the definition of N_l in (3.24), we have

$$\sum_{l=0}^{L} \text{Var}\left(\widehat{Y}_l\right) \leq \sum_{l=0}^{L} c_2 \frac{h_l^{\beta}}{N_l} \leq \tfrac{\varepsilon^2}{2} T^{-(\beta-1)/2} \left(1 - M^{-(\beta-1)/2}\right) \sum_{l=0}^{L} h_l^{(\beta-1)/2}.$$

Using the above upper bound with the inequality

$$\sum_{l=0}^{L} h_l^{(\beta-1)/2} < T^{(\beta-1)/2} \frac{M^{(\beta-1)/2}}{M^{(\beta-1)/2} - 1}, \tag{3.25}$$

we obtain

$$\text{Var}\left(\widehat{Y}\right) \leq \tfrac{\varepsilon^2}{2} T^{-(\beta-1)/2} \left(1 - M^{-(\beta-1)/2}\right) \sum_{l=0}^{L} h_l^{(\beta-1)/2} < \tfrac{\varepsilon^2}{2}.$$

Hence the inequality (3.18) for the mean square error of $\widehat{Y}$ holds also in this case. It remains to find the appropriate upper limit of the computational complexity

of the combined estimator $\widehat{Y}$ in this case where $\beta > 1$. By the choice of N_l in (3.24), we have in particular

$$N_l < 2\varepsilon^{-2} c_2 T^{(\beta-1)/2} \left(1 - M^{-(\beta-1)/2}\right)^{-1} \frac{h_l^{(\beta+1)/2}}{\sqrt{1+\tilde{\lambda}h_l}} \sqrt{1+\tilde{\lambda}T} + 1.$$

Combining this with condition (iv) and the inequalities (3.20) and (3.25), we arrive at

$$\begin{aligned}
\text{Cost}\left(\widehat{Y}\right) &\leq \sum_{l=0}^{L} c_3 d N_l \frac{1+\tilde{\lambda}h_l}{h_l} \\
&< \sum_{l=0}^{L} c_3 d \frac{(1+\tilde{\lambda}h_l)}{h_l} \left[2\varepsilon^{-2} c_2 T^{(\beta-1)/2} \left(1 - M^{-(\beta-1)/2}\right)^{-1} \frac{h_l^{(\beta+1)/2}\sqrt{1+\tilde{\lambda}T}}{\sqrt{1+\tilde{\lambda}h_l}} \right. \\
&\qquad \left. +1 \right] \\
&\leq (1+\tilde{\lambda}T) d \left[2\varepsilon^{-2} c_2 c_3 T^{(\beta-1)/2} \left(1 - M^{-(\beta-1)/2}\right)^{-1} \sum_{l=0}^{L} h_l^{(\beta+1)/2} \right. \\
&\qquad \left. + c_3 \sum_{l=0}^{L} h_l^{-1} \right] \\
&< (1+\tilde{\lambda}T) d \left[2\varepsilon^{-2} c_2 c_3 T^{(\beta-1)} \left(1 - M^{-(\beta-1)/2}\right)^{-2} + c_3 \tfrac{M^2}{M-1} \left(\sqrt{2} c_1\right)^{1/\alpha} \varepsilon^{-2} \right].
\end{aligned}$$

Rearranging this expression we get the required upper bound for the computational cost of $\widehat{Y}$:

$$\text{Cost}\left(\widehat{Y}\right) \leq c_4 d \varepsilon^{-2} (1+\tilde{\lambda}T)$$

with $c_4 = 2c_2c_3T^{\beta-1}\left(1 - M^{-(\beta-1)/2}\right)^{-2} + c_3 \frac{M^2}{M-1}\left(\sqrt{2}c_1\right)^{1/\alpha}$.

Case 3: $0 < \beta < 1$**.** In the last case, we set the number of simulations for level l to

$$N_l = \left\lceil 2\varepsilon^{-2} c_2 h_L^{-(1-\beta)/2} \left(1 - M^{-(1-\beta)/2}\right)^{-1} \frac{h_l^{(\beta+1)/2}}{\sqrt{1+\tilde{\lambda}h_l}} \sqrt{1+\tilde{\lambda}T} \right\rceil. \tag{3.26}$$

Similarly to the previous cases, taking into account the independence of the partial estimators $\widehat{Y}_l$, condition (iii) and the definition of N_l in (3.26), we obtain

$$\sum_{l=0}^{L} \text{Var}\left(\widehat{Y}_l\right) \leq \sum_{l=0}^{L} c_2 \frac{h_l^{\beta}}{N_l} \leq \tfrac{\varepsilon^2}{2} h_L^{(1-\beta)/2} \left(1 - M^{-(1-\beta)/2}\right) \sum_{l=0}^{L} h_l^{-(1-\beta)/2}.$$

In addition we observe that

$$\sum_{l=0}^{L} h_l^{-(1-\beta)/2} = h_L^{-(1-\beta)/2} \sum_{l=0}^{L} \left(M^{-(1-\beta)/2}\right)^l,$$

where we applied the definition of the time step size $h_l = \frac{T}{M^l}$. Using next (recall that $\beta \in]0,1[$)

$$\sum_{l=0}^{L} \left(M^{-(1-\beta)/2}\right)^l < \sum_{l=0}^{\infty} \left(M^{-(1-\beta)/2}\right)^l = \left(1 - M^{-(1-\beta)/2}\right)^{-1},$$

we have

$$\sum_{l=0}^{L} h_l^{-(1-\beta)/2} < h_L^{-(1-\beta)/2} \left(1 - M^{-(1-\beta)/2}\right)^{-1}. \tag{3.27}$$

Therefore the variance upper bound is given by

$$\begin{aligned}
\mathrm{Var}\left(\widehat{Y}\right) &\leq \tfrac{\varepsilon^2}{2} h_L^{(1-\beta)/2} \left(1 - M^{-(1-\beta)/2}\right) \sum_{l=0}^{L} h_l^{-(1-\beta)/2} \\
&< \tfrac{\varepsilon^2}{2} h_L^{(1-\beta)/2} \left(1 - M^{-(1-\beta)/2}\right) h_L^{-(1-\beta)/2} \left(1 - M^{-(1-\beta)/2}\right)^{-1} \\
&= \tfrac{\varepsilon^2}{2}.
\end{aligned}$$

Finally we have to find the appropriate upper bound for the computational complexity of the combined estimator in the case where $0 < \beta < 1$. First, using condition (iv), the upper bound of N_l as in (3.26), we obtain

$$\begin{aligned}
\mathrm{Cost}\left(\widehat{Y}\right) &\leq \sum_{l=0}^{L} c_3 d N_l \frac{1 + \tilde{\lambda} h_l}{h_l} \\
&< \sum_{l=0}^{L} c_3 d \frac{1 + \tilde{\lambda} h_l}{h_l} \left[2\varepsilon^{-2} c_2 h_L^{-(1-\beta)/2} \left(1 - M^{-(1-\beta)/2}\right)^{-1} \frac{h_l^{(\beta+1)/2} \sqrt{1 + \tilde{\lambda} T}}{\sqrt{1 + \tilde{\lambda} h_l}} \right. \\
&\qquad \left. + 1 \right] \\
&\leq (1 + \tilde{\lambda} T) d \left[2\varepsilon^{-2} c_2 c_3 h_L^{-(1-\beta)/2} \left(1 - M^{-(1-\beta)/2}\right)^{-1} \sum_{l=0}^{L} h_l^{(\beta-1)/2} \right. \\
&\qquad \left. + c_3 \sum_{l=0}^{L} h_l^{-1} \right].
\end{aligned}$$

Taking into account inequality (3.27), we observe

$$h_L^{-(1-\beta)/2} \left(1 - M^{-(1-\beta)/2}\right)^{-1} \sum_{l=0}^{L} h_l^{-(1-\beta)/2} < h_L^{-(1-\beta)} \left(1 - M^{-(1-\beta)/2}\right)^{-2}$$

and then using the lower bound given in (3.19), we obtain

$$\begin{aligned}
& h_L^{-(1-\beta)/2} \left(1 - M^{-(1-\beta)/2}\right)^{-1} \sum_{l=0}^{L} h_l^{-(1-\beta)/2} \\
< \; & \left(\sqrt{2} c_1\right)^{(1-\beta)/\alpha} M^{1-\beta} \varepsilon^{-(1-\beta)/\alpha} \left(1 - M^{-(1-\beta)/2}\right)^{-2}.
\end{aligned}$$

In addition we notice that since $\beta \in]0, 1[$, $\alpha > 0$ and $\varepsilon < e^{-1} < 1$,

$$\varepsilon^{-2} < \varepsilon^{-2-(1-\beta)/\alpha}.$$

Combining this results with the inequality (3.20), we end up with

$$\mathrm{Cost}\left(\widehat{Y}\right) < c_4 d \varepsilon^{-2-(1-\beta)/\alpha} (1 + \tilde{\lambda} T),$$

where

$$c_4 = 2c_3c_2\left(\sqrt{2}c_1\right)^{(1-\beta)/\alpha} M^{1-\beta}\left(1 - M^{-(1-\beta)/2}\right)^{-2} + c_3\frac{M^2}{M-1}\left(\sqrt{2}c_1\right)^{1/\alpha}.$$

This completes the last case and therefore the proof of the complexity theorem.

□

Remark 3.7. *Note that by setting the jump intensities $\lambda_1, \lambda_2, \ldots, \lambda_r$ to zero, and by considering the one-dimensional case, we reproduce the complexity theorem for the multilevel Monte Carlo method based on diffusions stated in [12].*

Now we briefly discuss the conditions of the complexity theorem. The first condition, (i), defines an upper bound for the bias of the estimator P_l. The constant α can be obtained by looking at the order of weak convergence of the numerical approximation method. For the regular and the jump-adapted Euler method, under appropriate conditions for $a(t, S(t))$, $b(t, S(t))$, $c(t, S(t))$ and the jump intensities $\lambda_1, \lambda_2, \ldots, \lambda_r$, the weak order is one and therefore $\alpha = 1$ (see Section 2.1). The second condition, (ii), and the last condition, (iv), limit the choice of the partial estimators $\widehat{Y}_l$. Condition (ii) fixes the mean of $\widehat{Y}_l$ and the fourth condition (iv) defines an upper bound for the computational complexity of the partial estimators $\widehat{Y}_l$. Note that these two conditions can usually be met by choosing partial estimators that are located around a given point (can be obtained by deducting the bias of the estimator from the estimator and using this difference as a new estimator) and by considering an upper limit for the computational complexity of the estimators. The most delicate condition is the third one, (iii). This condition demands the variance of $\widehat{Y}$ to be bounded by the term given on the right-hand side of (iii). In the case of the regular and the jump-adapted Euler method, as in the approach in Section 3.2, an upper bound for the variance, in particular β, can be found using the strong convergence property of the numerical method. In particular for the Euler method, we have $\beta = 1$ (see Section 2.1).

4 Numerical Examples

In this section we consider two jump-diffusion models, the Merton and the Kou model, and compare numerically the performance of the proposed multilevel Monte Carlo method to the standard Monte Carlo method without any variance reduction technique, as well as to the standard Monte Carlo method with variance reduction techniques, in this case we use antithetic variates and control variates.

4.1 The Merton and the Kou Model

The Merton model, introduced in 1976 by Robert C. Merton in [15], is historically the first jump-diffusion model in finance. The Kou model was first presented in 2002 by Steven G. Kou in [22]. Both models are specified by the particular form ($d = 1$, $m = 1$, $r = 1$) of the SDE (2.1):

$$\begin{cases} \mathrm{d}S(t) &= \mu S(t-)\mathrm{d}t + \sigma S(t-)\mathrm{d}W(t) + S(t-)\mathrm{d}J(t), \qquad 0 \le t \le T, \\ S(0) &= S_0, \end{cases} \tag{4.1}$$

where $J(t) = \sum_{i=1}^{N(t)} (V_i - 1)$, and where $N(t)$ is a Poisson process with intensity λ. In the Merton model the jump sizes are characterised by

$$\log(V_i) \overset{\text{iid}}{\sim} \mathcal{N}\left(\eta, \nu^2\right)$$

with $\eta \in \mathbb{R}$ and $\nu > 0$. In the Kou model the jump sizes are double exponentially distributed, i.e.,

$$\log(V_i) \overset{\text{iid}}{\sim} \mathcal{K}\left(\eta_1, \eta_2, p\right),$$

where $\mathcal{K}$ is an asymmetric exponential distribution, whose density function is given by

$$f_k(x) = p\eta_1 e^{-\eta_1 x} 1_{\{x \geq 0\}} + (1-p)\eta_2 e^{\eta_2 x} 1_{\{x<0\}} \tag{4.2}$$

with $x \in \mathbb{R}$, $\eta_1 > 1$, $\eta_2 > 0$ and $p \in [0,1]$. The parameters η_1 and η_2 define the decay of the tails in the distribution of positive and negative jumps and the parameter p specifies the probability of an upward jump (see e.g. [7, pp.111-112]).
Both jump-diffusion models admit an analytical solution given by

$$S(t) = S_0 \exp\left\{\left(\mu - \frac{\sigma^2}{2}\right)t + \sigma W(t)\right\} \prod_{i=1}^{N(t)} V_i$$

(see e.g.[15, p.129] and [22, p.1088]).
In the following we focus on pricing European call options, that is we intend to estimate the expectation $\mathbb{E}\left[f(S(T))\right]$ with f given as

$$f(S(T)) = e^{-rT} \max\left(S(T) - K, 0\right),$$

where r is the risk-free interest rate, T the maturity and K the strike price of the option, and $S(T)$ is specified by Equation (4.1) (see e.g. [7, pp.321-324]).

Remark 4.1. *To price options one works with risk-neutral measures. Under the risk-neutral probability measure, i.e. under the measure such that the discounted underlying* $\left(e^{-rt}S(t)\right)_{t\geq 0}$ *is a martingale (see [24, pp.67-71]), the Merton model is given by (4.1) with drift*

$$\mu = r - \lambda\left(\mathbb{E}\left[V_i\right] - 1\right) = r - \lambda\left(\exp\left(\eta + \frac{\nu^2}{2}\right) - 1\right)$$

(see e.g. [15, pp.128-129] and [14, p.137]). Similarly, the drift for the Kou model in the risk-neutral case can be specified by

$$\mu = r - \lambda\left(\frac{p\eta_1}{\eta_1 - 1} + \frac{q\eta_2}{\eta_2 + 1} - 1\right)$$

(see e.g. [23, p.1179]).

4.2 Two Variance Reduction Techniques

We now describe the two variance reduction techniques which we compare in the following numerically with the MLMC method.

4.2.1 Antithetic Variates

The idea of the antithetic variates is to produce for every sample path an antithetic one, which is based on the realisations of the original path, and thus is computationally cheap to get. For instance, if the random variable U is uniformly distributed over the interval $[0,1]$, then so is $1-U$. Suppose sample paths are generated by realisations $u_1, u_2, \ldots$ of U. Then the antithetic paths are produced using $1-u_1, 1-u_2, \ldots$ as realisations (see e.g. [14, p.205]).
The antithetic variates estimator is given by

$$\widehat{Y}_{AV} = \frac{1}{2N}\left(\sum_{i=1}^{N} f\left(S_G^{(i)}\right) + \sum_{i=1}^{N} f\left(\tilde{S}_G^{(i)}\right)\right),$$

where the realisations $\tilde{S}_G^{(i)}$ are based on an antithetic path. To produce sample paths of the two jump-diffusion models presented in Section 4.1 one requires realisations of the normal distribution for the diffusion part, and the log-normal and the double exponential distribution for either of the jump parts. If x is a realisation of the normal distribution, then we take $\tilde{x} = -x$ for the antithetic path. For the Merton model, if x is log-normally distributed, then the antithetic realisation is obtained by

$$\tilde{x} = \exp\left(\mu - \sigma\left(\frac{\log(x)-\mu}{\sigma}\right)\right).$$

For the Kou model, observe that a realisation of the double exponential distribution with density given in (4.2) can be obtained by

$$x = \begin{cases} \frac{1}{\eta_2}\ln\left(\frac{u}{1-p}\right), & \text{if } u < 1-p, \\ -\frac{1}{\eta_1}\left[\ln\left(\frac{1}{p}\right) + \ln(1-u)\right], & \text{if } u \geq 1-p, \end{cases}$$

where u is a realisation of a standard uniform distribution. Hence, for the antithetic path we take into account $1-u$ instead of u, i.e.

$$\tilde{x} = \begin{cases} \frac{1}{\eta_2}\ln\left(\frac{1-u}{1-p}\right), & \text{if } 1-u < 1-p, \\ -\frac{1}{\eta_1}\left[\ln\left(\frac{1}{p}\right) + \ln(u)\right], & \text{if } 1-u \geq 1-p. \end{cases}$$

4.2.2 Control Variates

Suppose we would like to estimate the mean of a random variable Y. Let $\bar{Y}$ be an unbiased estimator of $\mathbb{E}[Y]$ and let X be another random variable (called control variate) with known mean. Then $Y^* = \bar{Y} - \xi(X - \mathbb{E}[X])$ is also an unbiased estimator of $\mathbb{E}[Y]$ for any coefficient ξ, but the variance of Y^* can be minimised. Note that the parameter ξ can be estimated using a least-squares approach (see e.g. [14] pp.186-187). The same idea also applies for functionals of random variables.
For our example, under the risk-neutral measure the process $\left(e^{-rt}S(t)\right)_{t\geq 0}$ is a martingale. Hence,

$$\mathbb{E}\left[e^{-rT}S(T)\right] = S_0,$$

and thus, $S(T)$ can be used as control variate (see [14, pp.188-189]). The control variates estimator is given by

$$\widehat{Y}_{CV} = \frac{1}{N}\sum_{i=1}^{N}\left(f\left(S_G^{(i)}\right) - \widehat{\xi}_N\left(S_G^{(i)} - e^{rT}S_0\right)\right),$$

where $\widehat{\xi}_N$ is the least-squares estimator of ξ (see e.g. [14, p.188]).

4.3 Numerical Results

We consider Equation (4.1) in the setting of Section 4.1 with $T = 1$, $S_0 = 1$, $K = 1$, $r = 0.05$, $\sigma = 0.2$, $\lambda = 1$. Further we employ the refinement factor $M = 4$ in the MLMC method. For the Merton model, as in Example 10.2 in [7], we fix $\eta = -0.1$ and $\nu = 0.1$. For the Kou model, we set the probability of an upward jump to $p = 0.3$ and we fix $\eta_1 = 50$ and $\eta_2 = 25$, as, for instance, in [23, pp.1184-1185].

Remark 4.2. *With regard to a financial model, the chosen data have the following meaning: The parameter T represents the maturity of the option and the initial share price S_0 as well as the strike price are 1 (we do not specify the currency here). The risk-free interest rate is set to 5% and the implied volatility is chosen to be 20% (see e.g. [19, pp.296-297]). In a financial context the jump intensity typically lies between 0.05 and 2, see e.g. [3, pp.369-371]. Here we have chosen $\lambda = 1$ as this is often the case in [7] (see e.g. Example 10.1 and Example 10.2).*

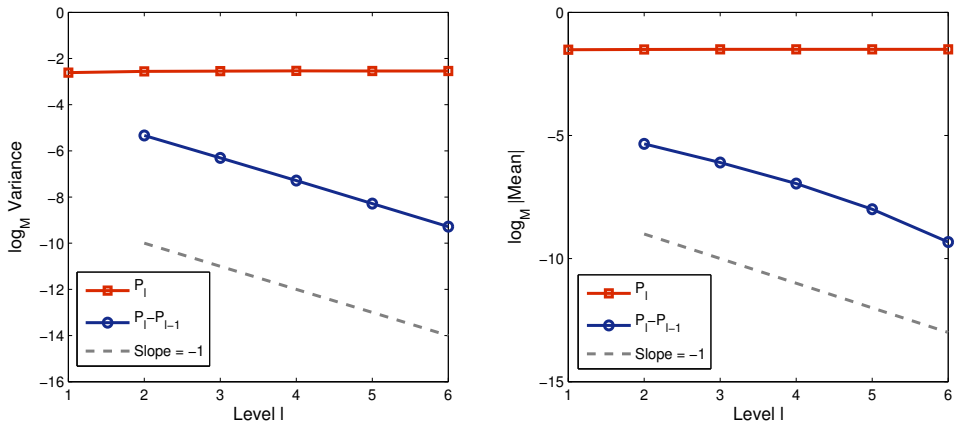

Figure 1: Comparison of the variance (left figure) and the mean (right figure) of the MLMC and the MC method over different levels using the Merton model and jump-adapted Euler.

In Figure 1 we compare the variance and the mean of the MLMC method and the standard Monte Carlo method without any variance reduction technique using the jump-adapted Euler method applied to the Merton model. To produce the two plots, 2×10^4 sample paths have been generated. The left plot shows the logarithm with base M of the variance of P_l, the discrete approximation of the variable P using the time step size $h_l = \frac{T}{M^l}$, and P_{l-1}, respectively, against the number of levels l. One observes that the curve for $P_l - P_{l-1}$ is almost parallel to the straight line of slope minus one. This indicates that the variance of a single sample verifies $\mathrm{Var}\left(P_l - P_{l-1}\right) = \mathcal{O}\left(h_l\right)$ as suggested by the theory, see Section 3.2. Note that the variance of P_l, used for the standard MC method, is more or less constant whereas the variance of $P_l - P_{l-1}$, used for the MLMC method, decreases as l increases. At level $l = 6$, $\mathrm{Var}\left(P_l - P_{l-1}\right)$ is approximately 4^6 times smaller than $\mathrm{Var}\left(P_l\right)$.
The right plot of Figure 1 represents the logarithm with base M of the absolute value of the mean of P_l and $P_l - P_{l-1}$, respectively, against the number of levels l. The curve for

$\mathbb{E}[P_l - P_{l-1}]$ is almost linear with slope minus one. Therefore, we have $\mathbb{E}[P_l - P_{l-1}] = \mathcal{O}(h_l)$, which corresponds to the weak convergence of order one of the jump-adapted Euler method, see Section 2.1. At level $l = 6$, the absolute value of $\mathbb{E}[P_l - P_{l-1}]$, used for the MLMC method, is about 4^8 times smaller than the absolute value of $\mathbb{E}[P_l]$, used for the standard Monte Carlo method.
We next also compare the MLMC method to the standard MC method with two variance reduction techniques described in Section 4.2. As a measure of the effectiveness of the methods we use the mean square error (MSE) described in Section 3.1 (see also, e.g., [14]). The MSE can be approximated by a sample average over N^* simulations:

$$\mathrm{MSE}\left(\widehat{Y}\right) = \mathbb{E}\left[\left(\widehat{Y} - \mathbb{E}\left[f(S(T))\right]\right)^2\right] \approx \frac{1}{N^*}\sum_{i=1}^{N^*}\left(\widehat{Y}^{(i)} - Y\right)^2, \tag{4.3}$$

where $\widehat{Y}^{(i)}$ are independent realisations at time T. For our computations we have used $N^* = 1000$. We also report the computational cost of the different methods for ε being the desired parameter in the mean square accuracy of $\mathrm{MSE}\left(\widehat{Y}\right) = \mathcal{O}(\varepsilon^2)$ for the MLMC method. Based on (3.16) in Section 3.2 we can bound the computational cost of the MLMC method in the jump-adapted case by

$$\mathrm{Cost}\left(\widehat{Y}\right) \leq 2T\varepsilon^{-2}(1 + \lambda T)(L + 1)^2, \tag{4.4}$$

where we have set the constants K and $\widehat{K}$ in (3.16) to 1. Similarly we get in the regular case the same upper bound without the factor $(1 + \lambda T)$. The computational cost of the other three methods is given by

$$\mathrm{Cost}\left(\widehat{Y}\right) = \begin{cases} 2N\left(\frac{T}{h} + \lambda T\right), & \text{for jump-adapted Euler,} \\ 2N\frac{T}{h}, & \text{for regular Euler.} \end{cases} \tag{4.5}$$

Remark 4.3. *The only additional cost, compared to the standard Monte Carlo method, of the control variates estimator is due to the computation of $\widehat{\xi}_N$. However, this cost is negligible compared to the other cost since $\widehat{\xi}_N$ can often be computed using vector multiplication. For the antithetic variates, antithetic paths are generated, but these are based on realisations of the original sample paths, and thus, no significant additional cost occurs.*

ε	MLMC	MC	AV	CV
0.1	1.00e-2	1.10e-3	2.89e-4	2.44e-5
0.01	1.00e-4	7.98e-5	3.96e-5	1.99e-5
0.005	2.50e-5	3.68e-5	2.56e-5	1.31e-5
0.002	4.00e-6	2.81e-5	2.33e-5	1.14e-5
0.001	1.00e-6	2.27e-5	2.21e-5	1.06e-5
0.0001	1.00e-8	2.18e-5	2.15e-5	1.03e-5

Table 1: Estimated mean square error of the methods MLMC, standard MC, antithetic variates (AV) and control variates (CV) for different values of ε using the Merton model and jump-adapted Euler.

Table 1 shows the estimated mean square error of the four methods for different values of ε using the Merton model and jump-adapted Euler. Similar results have been obtained

in the regular case and for the Kou model. The results are obtained through the following procedure. For a given ε, using (4.4), the upper bound for the computational cost of the MLMC is computed and then fixed. Furthermore, the total number of levels L is determined according to (3.15). For the MC, the antithetic and the control variates methods, we fix the step size $h = h_L = T/M^L$ as in [20] or [12]. Then, the number of simulations N is computed for these methods by (4.5). Finally, we run the N simulations and approximate the MSE of the different methods according to (4.3). Figure 2 illustrates the results in a loglog-plot. The accuracy ε is taken from the set

$$\varepsilon \in \{0.1, 0.01, 0.005, 0.002, 0.001, 0.0001\}\,.$$

We observe that as ε gets smaller, i.e., as the accuracy increases, the MLMC method has the lowest mean square error of all methods, followed by the control variates method, the antithetic method and the standard Monte Carlo method. In addition we notice that the MSE of the MLMC method decays linearly as ε tends to zero, whereas the MSE of the other three methods first roughly decays linearly to finally hardly decay as ε decreases.

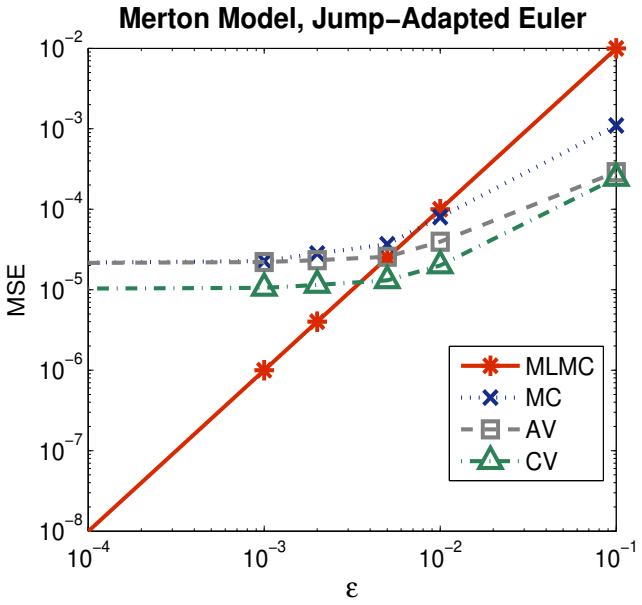

Figure 2: Estimated mean square error of the methods MLMC, standard MC, antithetic variates (AV) and control variates (CV) for different values of ε using the Merton model and jump-adapted Euler.

5 Conclusion

In this paper we have extended the multilevel Monte Carlo method to multi-dimensional stochastic differential equations driven by jump-diffusions for estimating the expectation of functionals depending on d-dimensional SDEs driven by an m-dimensional Wiener process and an r-dimensional compound Poisson process. Numerical experiments have been carried out to compare the MLMC method to the Monte Carlo method without any variance reduction technique as well as with two variance reduction techniques, the antithetic variates and the control variates. The numerical results confirm

our theoretical findings and show for a sufficiently small mean square accuracy a significant reduction of the computational complexity of the MLMC method compared to the other methods.

References

[1] Y. Ait-Sahalia. Telling from discrete data whether the underlying continuous-time model is a diffusion. *Journal of Finance*, 57:2075–2112, 2002.

[2] A. Blumenthal. The Multilevel Monte Carlo Method for SDEs driven by Jump-Diffusions with Applications in Finance. Master's thesis, Swiss Federal Institute of Technology Lausanne, January 2011.

[3] N. Bruti-Liberati and E. Platen. Approximation of Jump Diffusions in Finance and Economics. *Computational Economics*, 29:283–312, 2007.

[4] N. Bruti-Liberati and E. Platen. *Numerical Solution of Stochastic Differential Equations with Jumps in Finance*. Springer, Sydney, 2010.

[5] E. Buckwar and M. G. Riedler. Runge-Kutta methods for jump-diffusion differential equations. *Journal of Computational and Applied Mathematics*, 236(6):1155–1182, 2011.

[6] G. M. Clarke and D. Cooke. *A basic course in statistics*. Edward Arnold, London, third edition, 1992.

[7] R. Cont and P. Tankov. *Financial Modelling With Jump Processes*. Chapman & Hall/CRC, London, 2004.

[8] S. Dereich. Multilevel Monte Carlo algorithms for Lévy-driven SDEs with Gaussian correction. *Ann. Appl. Probab.*, 21(1):283–311, 2011.

[9] V. Fabian and J. Hannan. *Introduction to Probability and Mathematical Statistics*. John Wiley & Sons, Michigan, 1985.

[10] A. Gardoń. The order of approximations for solutions of Itô-type stochastic differential equations with jumps. *Stochastic Analysis and Applications*, 22(3):679–699, 2004.

[11] J. E. Gentle. *Random Number Generation and Monte Carlo Methods*. Springer, New York, 2003.

[12] M. B. Giles. Multilevel Monte Carlo path simulation. *Operations Research*, 56(3):607–617, 2008.

[13] D. Gillespie. Stochastic simulation of chemical kinetics. *Annu. Rev. Phys. Chem.*, 58, 2007.

[14] P. Glasserman. *Monte Carlo Methods in Financial Engineering*. Springer, New York, 2004.

[15] P. Glasserman and N. Merener. Numerical solution of jump-diffusion LIBOR market models. *Finance and Stochastics*, 7:1–27, 2003.

[16] S. Heinrich. Monte Carlo complexity of global solution of integral equations. *Journal of Complexity*, 14:151–175, 1998.

[17] D. J. Higham. An Algorithmic Introduction to Numerical Simulation of Stochastic Differential Equations. *SIAM Review*, 43(3):525–546, 2001.

[18] D. J. Higham and P. E. Kloeden. Numerical methods for nonlinear stochastic differential equations with jumps. *Numerische Mathematik*, 101:101–119, 2005.

[19] J. Hull. *Options, futures, and other derivatives*. Pearson Prentice Hall, New Jersey, 2009.

[20] A. Kebaier. Statistical Romberg Extrapolation: A new variance reduction method and applications to option pricing. *Annals of Applied Probability*, 15(4):2681–2705, 2005.

[21] P. E. Kloeden and E. Platen. *Numerical Solution of Stochastic Differential Equations*. Springer-Verlag, Berlin, 1995.

[22] S. G. Kou. A Jump-Diffusion Model for Option Pricing. *Management Science*, 48:1086–1101, 2002.

[23] S. G. Kou and H. Wang. Option pricing under a double exponential jump diffusion model. *Management Science*, 50:1178–1192, 2004.

[24] D. Lamberton and B. Lapeyre. *Introduction to Stochastic Calculus Applied to Finance*. ellipses, Paris, 1997.

[25] X. Q. Liu and C. W. Li. Weak approximations and extrapolations of stochastic differential equations with jumps. *SIAM J. Numer. Anal.*, 37:1747–1767, 2000.

[26] Y. Maghsoodi. Mean square efficient numerical solution of jump-diffusion stochastic differential equations. *Indian J. Statist.*, 58:25–47, 1996.

[27] X. Mao and C. Yuan. *Stochastic Differential Equations with Markovian Switching*. Imperial College Press, London, 2006.

[28] H. Marxen. The Multilevel Monte Carlo methof used on a Lévy driven SDE. *Monte Carlo methods and Applications*, 16(2):167–190, 2010.

[29] R. C. Merton. Option pricing when underlying stock returns are discontinuous. *Journal of Financial Economics*, 3:125–144, 1976.

[30] G. N. Milstein and M. V. Tretyakov. *Stochastic Numerics for Mathematical Physics*. Springer, Heidelberg, 2004.

[31] E. Mordecki, A. Szepessy, and R. Tempone. Adaptive weak approximation of diffusions with jumps. *SIAM J. Numer. Anal.*, 4:1732–1768, 2008.

[32] S. E. Shreve. *Stochastic Calculus for Finance II: Continuous-Time Models*. Springer Finance, Pittsburgh, 2004.

[33] P. Tankov and E. Voltchkova. Jump-diffusion models: a practitioner's guide. *Banque et MarchÃ ˆs*, 2009.

[34] D. Wilkinson. *Stochastic modeling for systems biology.* Chapman and Hall/CRC, 2006.

[35] Y. Xia. Multilevel Monte Carlo method for jump-diffusion SDEs. Technical report, University of Oxford, http://arxiv.org/abs/1106.4730, 2011.

[36] Y. Xia and M. B. Giles. Multilevel path simulation for jump-diffusion SDEs. *to appear in Monte Carlo and Quasi-Monte Carlo Methods 2010, Springer-Verlag*, 2012.