Multilevel Monte-Carlo Simulation Applied to Lévy Driven Assets

Azadeh Ghasemifard[1]
PhD student, Isfahan University of Technology, Isfahan, Iran

Mahdieh Tahmasebi
Assistant professor, Tarbiat Modares University, Tehran, Iran

Abstract

Inspired by recent advances in the application of the Multilevel Monte-Carlo (MLMC) approach to Lévy driven assets, we benefit this method to price European and Asian options for the Merton jump-diffusion model. We first explain the idea of MLMC scheme and its variants: strong (classical) and weak. Additionally, theoretical results are confirmed by numerical experiments. We use weak Euler scheme to numerically estimate the asset and apply weak MLMC method to price the option.

Keywords: Multilevel Monte-Carlo method, Lévy process, price estimation
Mathematics Subject Classification [2010]: 65C30, 65C05, 91G80

1 Introduction

Introduced by M. Giles [3] the MLMC method is a way to efficiently distribute the computational complexity caused by the variance and the bias over a series of levels. In contrast, the standard Monte-Carlo (MC) method deals with these two problems at the same time.

To show the idea assume a filtered probability space $(\Omega, \mathcal{F}, (\mathcal{F}_t)_{t\geq 0}, \mathbb{P})$ in which we have a n-dimensional process $(X_t)_{t\geq 0}$, which solves the following Lévy-driven stochastic differential equation (SDE)

$$\mathrm{d}X_t = a(X_{t-})\mathrm{d}Z_t, \tag{1}$$

with X_0 being a known $\mathbb{R}^n$-valued random variable, $Z_t = (Z_{t,1}, \dots, Z_{t,q})$, $t \geq 0$ is a q-dimensional Lévy process and the mapping $a : \mathbb{R}^n \to \mathbb{R}^n \times \mathbb{R}^q$ is Lipschitz continuous and satisfies the linear growth condition such that the solution exists and is unique [5].

We are not interested in the solution of (1) per se. but rather the expectation of a functional hereof, i.e.:

$$P = \mathbb{E}[f(X_T)], \tag{2}$$

which gives the fair price of the option based on the payoff f on the asset X, or for short, just the price [2, 4]. Using the numerical approximation X_t^Δ with stepsize $\Delta = T/2^L$ instead and expanding (2) into a telescopic sum we obtain:

$$\begin{aligned}
\tilde{P}^\Delta &= \mathbb{E}[f(X_T^T)] && (3)\\
&+ \mathbb{E}\left[f(X_T^{T/2}) - f(X_T^T)\right] && (4)\\
&+ \mathbb{E}\left[f(X_T^{T/4}) - f(X_T^{T/2})\right] && (5)\\
&\vdots && (6)\\
&+ \mathbb{E}\left[f(X_T^{\Delta}) - f(X_T^{2\Delta})\right]. && (7)
\end{aligned}$$

[1] speaker

It is typical to denote equation (3) as level $l = 0$, equation (4) as level $l = 1$, etc. equation (7) as level $l = L$.

Rather than taking the real expectation we apply an approximation Θ_l with M_l paths, e.g. the average of i.i.d. paths and obtain

$$\begin{aligned}\tilde{P}^{\Delta} &= \Theta_0(f(X_T^T)) \\ &+ \Theta_1\left(f(X_T^{T/2}) - f(X_T^T)\right) \\ &+ \Theta_2\left(f(X_T^{T/4}) - f(X_T^{T/2})\right) \\ &\vdots \\ &+ \Theta_L\left(f(X_T^{\Delta}) - f(X_T^{2\Delta})\right).\end{aligned}$$

for brevity denote $\Theta_l\left(f(X_T^{T/2^l}) - f(X_T^{T/2^{l-1}})\right) = \hat{\Theta}_l$, l=0,...,L.

This idea has 2 main benefits:

- A lot of the paths can be simulated where they are cheap, i.e. l small,
- If the stepsize has to be reduced, we do not have to throw already calculated paths away, we just add a new level.

In total, this gives us the error bound and computational complexity as explained below [3].

Let $\Theta_l(X)$ be independent estimators with M_l realisations of X, with

$$M_l = 2\varepsilon^{-2}\sqrt{V_l\Delta_l}\left(\sum_{i=0}^{L}\sqrt{V_i/\Delta_i}\right), \tag{8}$$

where V_l is the variance of $\hat{\Theta}_l$, Δ_l is the stepsize at level l and ε^2 is the chosen mean square error, i.e. $\mathbb{E}[(\tilde{P}^{\Delta} - \mathbb{E}[P])^2] < \varepsilon^2$.

If there exists positive constants $\alpha \geq 1/2$, β, c_1, c_2 and c_3 such that

1. $|\mathbb{E}[\tilde{P}^{\Delta} - P]| \leq c_1\Delta^{\alpha}$,
2. Θ_l is unbiased,
3. C_l the computational complexity of $\hat{\Theta}_l$, is bounded by $C_l \leq c_2\Delta_l^{-1}$,
4. $var(\hat{\Theta}_l) \leq c_3\Delta_l^{\beta}$,

then there exists a positive constant c_4 such that the multilevel estimator, $\tilde{P}^{\Delta}$, satisfies the mean square bound with computational complexity

$$C \leq \begin{cases} c_4\varepsilon^{-2}, & \beta > 1, \\ c_4\varepsilon^{-2}(log\varepsilon)^2, & \beta = 1, \\ c_4\varepsilon^{-2-(1-\beta)/\alpha}, & 0 < \beta < 1. \end{cases} \tag{9}$$

Out of the four conditions, the difficult one to satisfy is the fourth. The classical way to satisfy is through the strong convergence, i.e. if the numerical integrator is of strong order $\beta/2$ then $var(\hat{\Theta}_l) = O(\Delta_l^{\beta})$ [1], this is the classical (strong) MLMC.

However, requiring strong convergence, when one is only interested in the expected value is not suitable. This is especially clear for Lévy driven assets as the exact Lévy increments are typically very time consuming or sometimes even impossible to simulate. Belomestny and Nagapetyan [1] then introduced the weak MLMC scheme which relax the requirement of strong convergence with the aid of coupling idea. The key difference between weak and strong MLMC schemes is that in strong version, we can reuse already calculated paths as the level increases.

2 Main results

Consider a jump-diffusion process $(X_t)_{t\geq 0}$

$$\mathrm{d}X_t = r(X_{t-})\mathrm{d}t + \sigma(X_{t-})\mathrm{dW}_t + h(X_{t-})\mathrm{dJ}_t, \tag{10}$$

with $X_{t-} = \lim_{s\to t-} X_s$, $X_t = \lim_{s\to t+} X_s$ and

$$\mathrm{J}(t) = \sum_{i=1}^{N_t}(Y_i - 1)$$

where Y_i are random variables and N_t is a counting process, typically N_t is the Poisson process, which for Y_i i.i.d. makes J_t a compound Poisson process. As usual the functions r, σ and h satisfy the global Lipschitz condition and the linear growth condition, to make the problem well defined.

The standard simulation of the above is to first simulate the arrival times of the counting process and then add these points to the initial discretisation. The simulation of counting processes is in itself not an efficient task, but additionally for high intensities, this causes that we take significantly smaller steps than wanted. To circumvent this, we would like to use a weak approximation.

For this, assume that N_t is indeed the Poisson process, then as discussed by [1], we can gather the jumps on the discretisation points, and use the binomial distribution to simulate the number of jumps, $\eta_{l,i}$ at each level l and discretisation point i,

$$\eta_{l,i} = \mathrm{Bin}(2^{L-l}, 2^{-L}\lambda),$$

where λ is the intensity parameter of the Poisson process. Note that the second argument is the probability of a single outcome under the Bernoulli model, thus that value have to be $0 \leq 2^{-L}\lambda \leq 1$ and one have to ensure that the initial number of levels is sufficiently high. On the other hand, the i.i.d. random variables ξ_j satisfying in

$$\left|\mathbb{E}[\xi_j]\right| + \left|\mathbb{E}[\xi_j^2] - \Delta\right| + \left|\mathbb{E}[\xi_j^3]\right| = O(\Delta^2).$$

is applied to weak approximation of the Brownian increments on step j, where Δ is the time step. The simplest way to construct such variables is given by the two point distribution [4]

$$P(\xi_j = \pm\sqrt{\Delta}) = \frac{1}{2}.$$

Thus applying the weak MLMC, we observe that we sum together shifted and scaled Bernoulli random numbers, which can be generated by the Binomial distribution

$$\frac{\xi_{j,l}}{2\sqrt{\Delta_L}} + 2^{L-l-1} \sim \mathrm{Bin}\left(2^{L-l}, \frac{1}{2}\right).$$

Note that one should use table lookup methods or similar algorithms to generate the Binomial random numbers rather than using a sum of Bernoulli numbers to achieve a speed up. See [1] for a discussion. However, the classical Box-Müller or the newer Ziggurat method are so efficient at generating standard normal numbers, the weak MLMC is not necessarily faster than the standard MLMC.

1-Dimensional Merton Jump-Diffusion

Consider the asset driven by jump-diffusion problem with linear functions

$$\mathrm{d}X_t = \left(r - \lambda \cdot \left(e^{m+0.5\theta^2} - 1\right)\right) X_{t-}\mathrm{d}t + \sigma X_{t-}\mathrm{dW}_t + X_{t-}\mathrm{dJ}_t, \tag{11}$$

where N_t is the Poisson process with intensity λ and $ln(Y_i) \sim N(m, \theta^2)$, which is simulated per se. and not changed to a weak version.

We are once again interested in the European option

$$f(X_T) = e^{-rT} \max(X_T - K, 0),$$

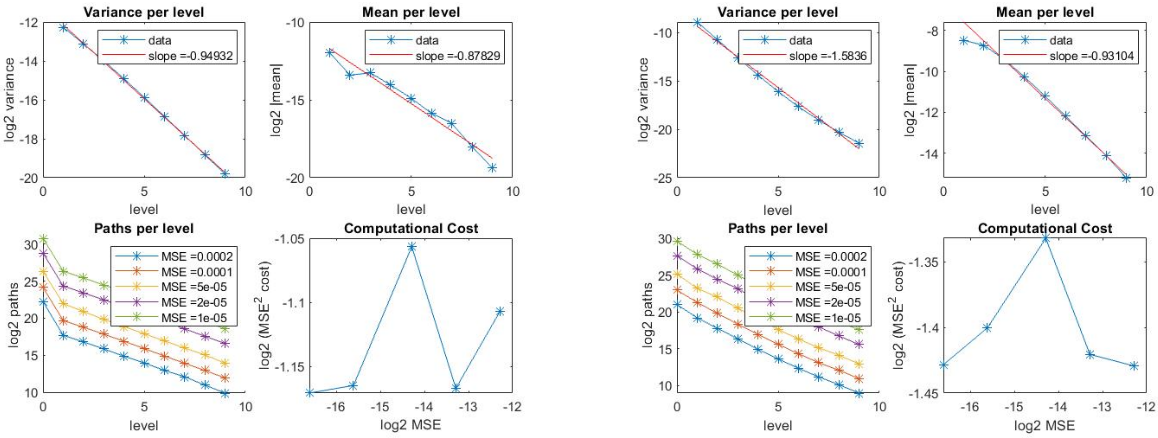
and the Asian option

$$f(X_t) = e^{-rT} \max\left(\frac{1}{T}\int_0^T X_s ds - K, 0\right),$$

with parameters

$$r = 0.05, \quad \sigma = 0.2, \quad T = 1, \quad K = 1, \quad X_0 = 1, \quad \lambda = 0.5, \quad m = 0.05, \quad \theta = 0.25, \tag{12}$$

and let the algorithm automatically choose the right number of levels as $\varepsilon = \{2e{-}4, 1e{-}4, 5e{-}5, 2e{-}5, 1e{-}5\}$ is varied.



(a) European option (b) Asian option

Figure 1: Merton jump-diffusion Model (value ≈ 0.1276)

The results are shown in fig. 1. As expected we see first order variance reduction and weak convergence for European option. The number of paths decrease by increasing the levels as they should be for the MLMC algorithm. The computational complexity order is less than 2 as we expect for $\beta = 1$. The results of Asian option shows better behavior in variance and convergence order which predictably can be duo to more smoothness of payoff.

References

[1] D. Belomestny and T. Nagapetyan, *Multilevel path simulation for weak approximation schemes with application to Lévy -driven SDEs*, Bernoulli, 23 (2017)), pp. 927–950.

[2] F. Black and M. Scholes *The Pricing of Options and Corporate Liabilities*, Journal of Political Economy, 81 (1973), pp. 6377–654.

[3] M. B. Giles, *Multilevel Monte Carlo Path Simulation*, Operations Research, 56 (2008), pp. 607–617.

[4] P. E. Kloeden and E. Platen, *Numerical Solution of Stochastic Differential Equations*, Vol. 23 of Applications of Mathematics (New York), Springer-Verlag, Corrected Third Printing 1999.

[5] E. Platen and N. Bruti-Liberati, *Numerical Solution of Stochastic Differential Equations with Jumps in Finance*, Vol. 64, Springer, 2010.

Email: `azadeh.ghasemi@math.iut.ac.ir`
Email: `tahmasebi@modares.ac.ir`