

NUMERICAL SIMULATION ON BLACK-SCHOLES MODEL

PROJECT REPORT

Wenhao Yang

School of Mathematical Sciences

Peking University

yangwenhaosms@pku.edu.cn

ABSTRACT

In this report, I figure out the theoretical results and simulation results of stochastic differential equation on Black-Scholes model. The two simulation algorithms are Euler-Maruyama and Milstein scheme. Besides, multilevel monte carlo method is also applied to this model to make comparison with single monte carlo method.

1 MODEL DESCRIPTION AND THEORITICAL RESULTS

In this section, I will give some theoritical results of Black-Scholes model. The stochastic differential equation of Black-Scholes model is:

$$dS_t = rS_t dt + \sigma S_t dW_t \quad (1)$$

And the exact solution of S_t is: $S_t = S_0 e^{(r - \frac{1}{2}\sigma^2)t + \sigma W_t}$. In fact, we have:

$$\begin{aligned} d \log S_t &= \frac{1}{S_t} dS_t - \frac{1}{2S_t^2} (dS_t)^2 \\ &= r dt + \sigma dW_t - \frac{1}{2} \sigma^2 dt \\ &= (r - \frac{1}{2} \sigma^2) dt + \sigma dW_t \end{aligned} \quad (2)$$

$$\log S_t - \log S_0 = (r - \frac{1}{2} \sigma^2) t + \sigma W_t \quad (3)$$

$$S_t = S_0 e^{(r - \frac{1}{2} \sigma^2) t + \sigma W_t} \quad (4)$$

And we care about the random variable $P = e^{-r} \max\{0, S_1 - 1\}$, which has following properties such as:

$$\mathbb{P}(P = 0) = \mathbb{P}(S_1 \leq 1) = \mathbb{P}(W_1 \leq \frac{1}{2}\sigma - \frac{r}{\sigma} - \frac{\log S_0}{\sigma}) \quad (5)$$

$$\mathbb{P}(P > x) = \mathbb{P}(W_1 > \frac{\log(xe^r + 1)}{\sigma} - \frac{\log S_0}{\sigma} + \frac{\sigma}{2} - \frac{r}{\sigma}) \quad (6)$$

$$\mathbb{E}[P] = \int_0^{+\infty} \mathbb{P}(P > x) dx \quad (7)$$

As it is difficult to get the exact value of the expectation of P , we just leave it alone. The next sections will compute its numerical simulation result.

2 NUMERICAL APPROXIMATION METHODS AND SIMULATION RESULTS

There are two mainly approximation methods to simulate the stochastic process.

2.1 EULER-MARUYAMA APPROXIMATION

The first approximation method is called Euler-Maruyama:

$$S_{t_{m+1}}^{(n)} = S_{t_m}^{(n)} + rS_{t_m}^{(n)}\Delta t_m + \sigma S_{t_m}^{(n)}\eta_m \quad (8)$$

where $\Delta t_m = T/n = \delta$ and $\eta_m = \Delta B_{t_m} \sim N(0, \delta)$

2.2 MILSTEIN APPROXIMATION

The second approximation method is called Milstein:

$$S_{t_{m+1}}^{(n)} = S_{t_m}^{(n)} + rS_{t_m}^{(n)}\Delta t_m + \sigma S_{t_m}^{(n)}\eta_m + \frac{1}{2}\sigma^2 S_{t_m}^{(n)}(\eta_m^2 - \Delta t_m) \quad (9)$$

2.3 SIMULATION RESULTS

All of the parameters are set as: $S_0 = 1$, $r = 0.05$ and $\sigma = 0.2$. In order to compare the efficiency of the two simulation methods, mean absolute error is taken as quantification. As the orders of these two simulation methods are not the same, $\sqrt{\delta}$ and δ are two orders for Euler-Maruyama and Milstein method, respectively. The mean absolute errors of two simulation methods are shown in Fig 1, from which the Milstein method has a more quick convergence rate because of higher order.

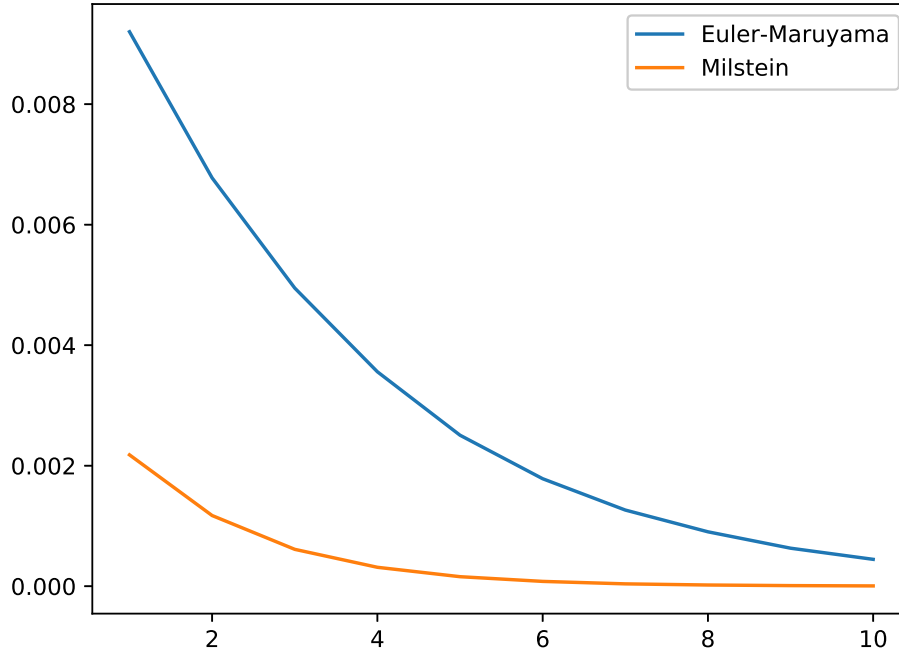


Figure 1: Mean absolute errors

Besides of comparison of strong convergence, weak convergence is also plotted in Fig 2, from which the two methods have a nearly same convergence rate because they both have same order in terms of weak convergence.

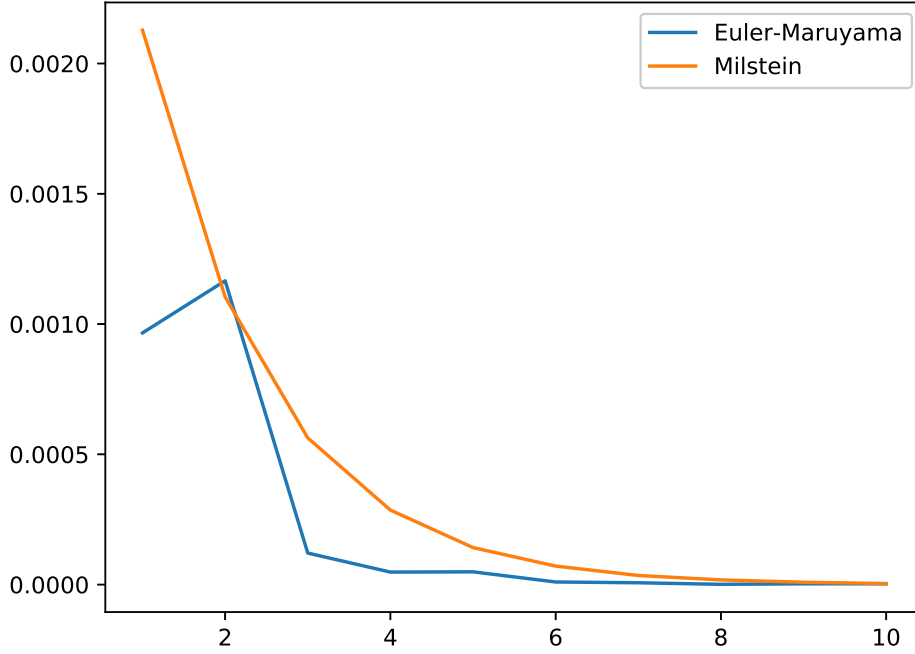


Figure 2: Absolute mean errors

3 MULTILEVEL MONTE CARLO SIMULATION METHOD

To reduce estimation variance, multilevel monte carlo is promoted. In this section, multilevel monte carlo method is used to simulate the Black-Scholes model.

3.1 MULTILEVEL MONTE CARLO

As our aim is to estimate the expectation $\mathbb{E}[f(S_{n,T})]$ to approximate $\mathbb{E}[f(S_T)]$, where n means discretize the time with M^{-n} . In this report, $M = 2$ is always supposed. In fact the expectation can be rewritten as:

$$\mathbb{E}[f(S_{n,T})] = \mathbb{E}[f(S_{1,T})] + \sum_{k=2}^n \mathbb{E}[f(S_{k,T}) - f(S_{k-1,T})] \quad (10)$$

So the estimation of $f(S_{n,T})$ could be written as:

$$\hat{f}(S_{n,T}) = \frac{1}{N_1} \sum_{i=1}^{N_1} f(S_{1,T}^{(i)}) + \sum_{k=2}^n \frac{1}{N_k} \sum_{i=1}^{N_k} [f(S_{k,T}^{(i)}) - f(S_{k-1,T}^{(i)})] \quad (11)$$

To reduce variance, for each k , $f(S_{k,T}^{(i)}) - f(S_{k-1,T}^{(i)})$ is independent with $f(S_{k-1,T}^{(j)}) - f(S_{k-2,T}^{(j)})$, which means, the estimation of $f(S_{k,T}^{(i)}) - f(S_{k-1,T}^{(i)})$ is depended on the path simulated by $f(S_{k,T}^{(i)})$ instead of $f(S_{k-1,T}^{(j)})$. So the variance of $\hat{f}(S_{n,T})$ is:

$$Var[\hat{f}(S_{n,T})] = \sum_{k=1}^n \frac{V_k}{N_k} \quad (12)$$

where $V_k = Var[f(S_{k,T}) - f(S_{k-1,T})]$

3.2 SIMULATION RESULTS

All the parameters are set the same as before and the maximum level is set as $L = 10$. For each level, 10^6 times monte carlo is used to do estimation. Here we give an investigation on V_k in Fig 3, where the x-axis is the level l and y-axis is the estimation variance V_l . The variance reduces as level goes high. So we can simulate less for higher levels but higher for lower levels. Here we leave the variance of V_0 as it performs the same under both algorithms.

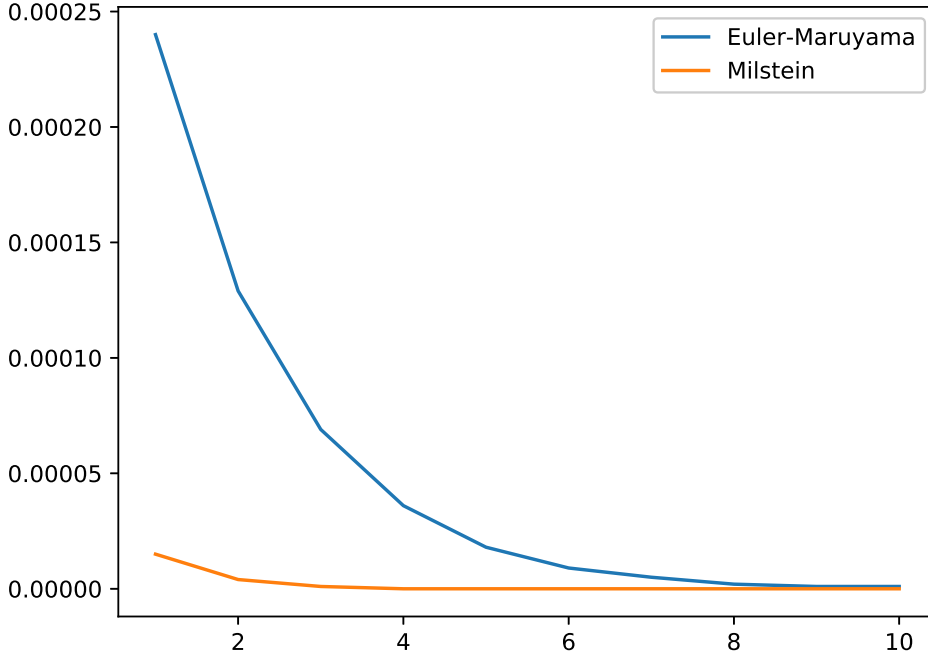


Figure 3: Variance

According to Giles (2008), the number of sampling at each level could be decided by:

$$N_l = \left\lceil 2\varepsilon^{-2} \sqrt{V_l h_l} \left(\sum_{l=0}^L \sqrt{V_l / h_l} \right) \right\rceil \quad (13)$$

where $h_l = M^{-l}T$.

Different convergence accuracies ε are set as 0.1, 0.01, 0.001, 0.0001. So the absolute mean errors $|\mathbb{E}f(S_{L,T}) - \mathbb{E}f(S_T)|$ for different accuracy are shown in Fig 4, from which we could find that the two algorithms have nearly the same performance on convergence. That is because only weak convergence could be test while strong convergence can't compute because of multilevel estimation. So it is nothing strange to find out the same performance because these two algorithms have same weak convergence order.

However, it is not to say these two algorithms are the same good in terms of multilevel monte carlo. From Fig 3 we find that Euler-Maruyama method has a higher variance than Milstein method, which means that Euler-Maruyama method should take more samples than Milstein method do. In terms of time cost, Milstein method has a better performance.

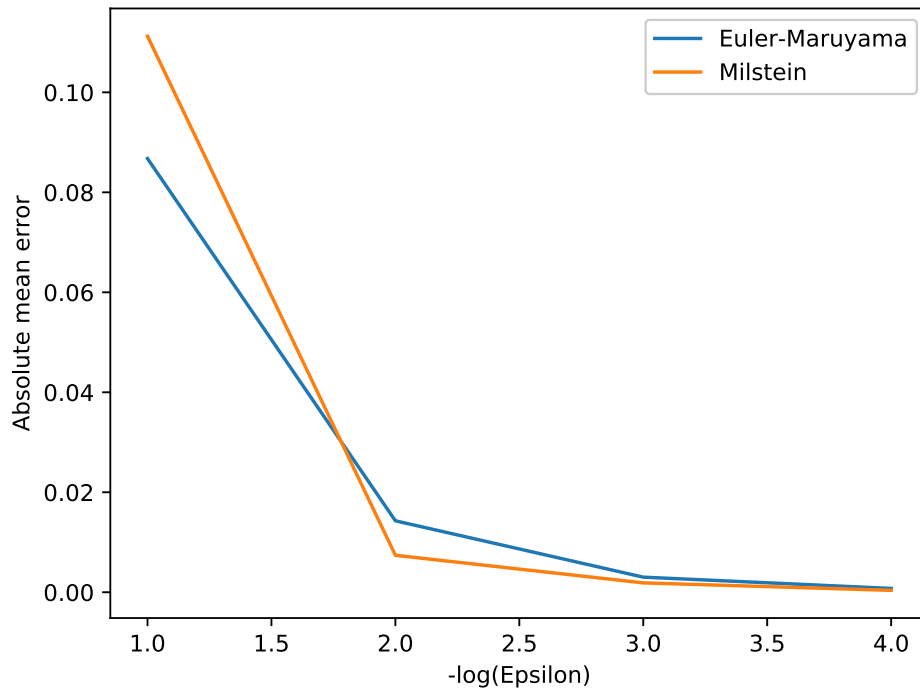


Figure 4: Error

EXPERIMENT DETAILS

All of my code and results can be found on website <https://github.com/yangwenh/stochastic-simulation>. The mainly parallel method is asynchronized algorithm, which is implemented by ZMQ.

ACKNOWLEDGMENTS

I would like to thank Prof Li giving me a oppertunity to do research on stochastic differential equation. Thank Teaching Assistant Xiong helps to figure out my problems on the research topic. Thank Deep Learning Lab providing me computation resources.

REFERENCES

Michael B. Giles. Multilevel monte carlo path simulation. *Operations Research*, 56(3):607–617, 2008.