Option pricing methods

Amine Ben Kraiem

January 2022

Exercise 1

The geometric Brownian motion S, solves

$$dS(t) = rS(t)dt + \sigma S(t)dW(t),$$
$$S(0) = S_0.$$

where, in the context of option pricing, r and σ are constants representing the short interest rate and the volatility

We denote $Y_t = \log S_t$.
By using the Itô formula for $f : [0,T]\times\mathbb{R} \to \mathbb{R}$ a given bounded function in $C^2([0,T]\times\mathbb{R})$ with $f(t,x) = \log(x)$
Then $Y(t) \equiv f(t, S(t))$ satisfies the SDE

$$dY(t) = (\partial_t f + \partial_x f a + .\partial_x^2 f \frac{b^2}{2})(t, S(t))dt + (\partial_x f b)(t, S(t))dW(t)$$

With

$$a = rS(t)$$

$$b = S(t)\sigma$$

$$dY(t) = rS(t)\frac{1}{S(t)}dt + \frac{S(t)^2}{2}\frac{-1}{S(t)^2}\sigma^2 dt + S(t)\sigma\frac{1}{S(t)}dW(t)$$

$$dY(t) = rdt - \frac{\sigma^2}{2}dt + \sigma dW(t)$$

$$Y(T) = Y_0 + rT - \frac{\sigma^2}{2}T + \sigma W(T)$$

Since $Y(T) = \log S(T)$, we finally get the exact solution for (1) is

$$S(T) = \exp\left(\left(r - \sigma^2/2\right)T + \sigma W(T)\right)S_0$$

We want to compute a numerical approximation to the option value

$$\pi = e^{-rT}E[g(S(T))]$$

First we consider M independent samples $S^{(m)}(T)$, $m = 1, .., M$ of the random variable

$$S(T) = \exp\left(\left(r - \sigma^2/2\right)T + \sigma W(T)\right)S_0$$

and we use the Monte Carlo estimator

$$\tilde{\pi} = e^{-rT}M^{-1}\sum_{m=1}^{M} g(S^{(m)})$$

We have 3 payoff functions :

1. g(x) = max(x - K,0)
2. g(x) = 1 if $\frac{K}{2}$<x<K and 0 else
3. g(x) = $\frac{x}{\frac{x}{K}^2+1}$

Exact payoff calculations for a call

1. We already know the value of a call options from previous homeworks the formula is below

$$C = \Phi(d_1) S_t - \Phi(d_2) K e^{-r(T-t)}$$
$$d_1 = \frac{1}{\sigma\sqrt{T-t}}\left[\ln\left(\frac{S_t}{K}\right) + \left(r + \frac{\sigma^2}{2}\right)(T-t)\right]$$
$$d_2 = d_1 - \sigma\sqrt{T-t}$$

cumulative function of standard normal distribution

Numerical application we have C = 8.4333

2.

$$c = e^{-rT}\int_{\frac{k}{2}<s(x)<k} \frac{1}{\sqrt{2\pi\tau}} e^{-\frac{1}{2}\frac{x^2}{T}} \cdot dx$$

$$\frac{K}{2} < S(x) < K$$
$$\frac{K}{2} < S_0 e^{\left(r-\frac{\sigma^2}{2}\right)T+\sigma x} < K$$
$$\frac{K}{2S0} < e^{\left(r-\frac{\sigma^2}{2}\right)T+\sigma x} < (\frac{K}{S0}$$
$$ln(\frac{K}{2S0}) < r - \frac{\sigma^2}{2}T + \sigma x < ln((\frac{K}{S0}))$$
$$\frac{ln(\frac{K}{2S0}) - r + \frac{\sigma^2}{2}T}{\sigma} < x < \frac{ln(\frac{K}{S0}) - r + \frac{\sigma^2}{2}T}{\sigma}$$

we pose :

$$\alpha_1 = \frac{ln(\frac{K}{2S0}) - r + \frac{\sigma^2}{2}T}{\sigma}$$

$$\alpha_2 = \frac{ln(\frac{K}{S0}) - r + \frac{\sigma^2}{2}T}{\sigma}$$

back to our integral :

$$c = e^{-rT}\int_{alpha_1<x<alpha_2} \frac{1}{\sqrt{2\pi\tau}} e^{-\frac{1}{2}\frac{x^2}{T}} \cdot dx$$

$$c = e^{-rT}\mathbb{P}(\frac{alpha_1}{\sqrt{T}} < Z < \frac{alpha_2}{\sqrt{T}})$$

With Z a standard normal random variable so that can be expressed in term of the cumulative function of standard normal distribution as follow :

$$c = e^{-rT}(1 - \Phi(\frac{alpha_1}{\sqrt{T}}) - \Phi(-\frac{alpha_2}{\sqrt{T}}))$$

Numerical application C = 0.5144504433957848

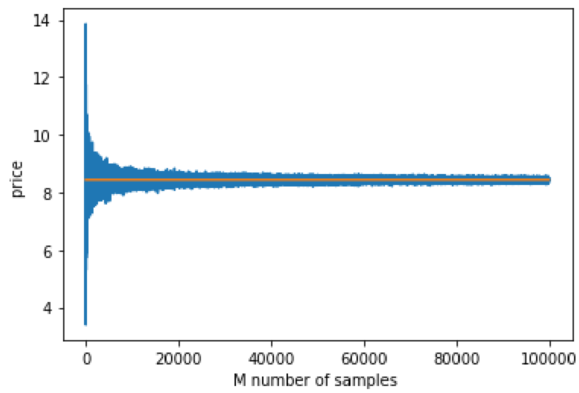
3. complicated, so we use a numerical integration method Numerical application using python to evaluate the option price is 48.556064546301315

To mention that in the next plots the statistcal error formula is a follow :

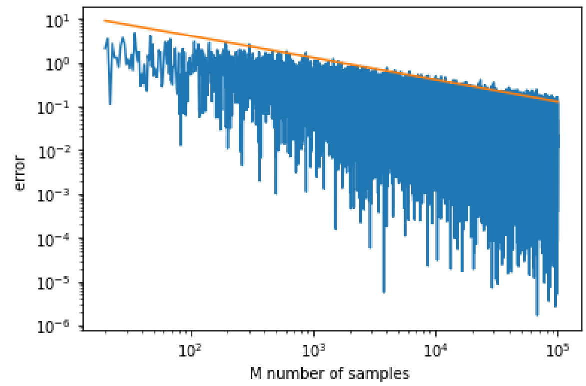
$$3\sqrt{\frac{Var(g(S)}{M}}$$

where M is the number of samples

Simulation results Exact error and Statistical error

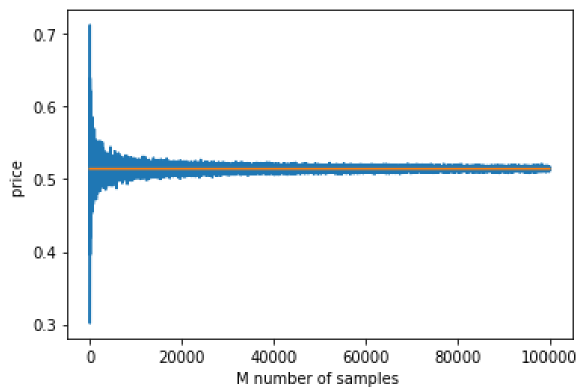


(a)

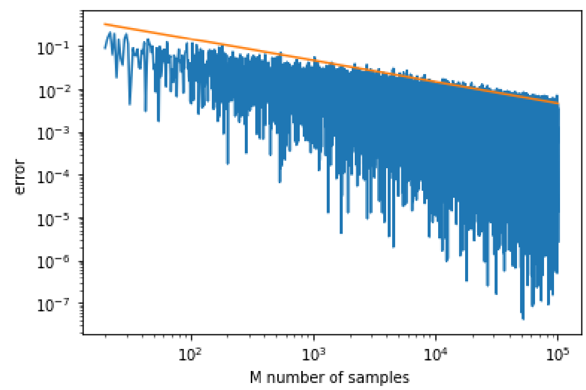


(b)

Payoff function 1

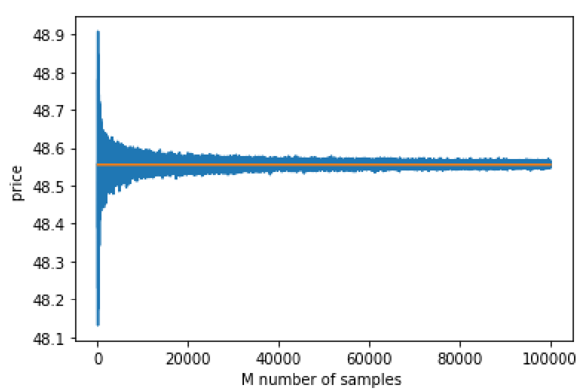


(c)

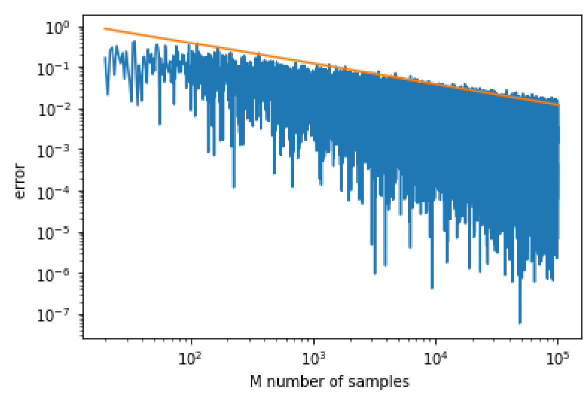


(d)

Payoff function 2



(e)

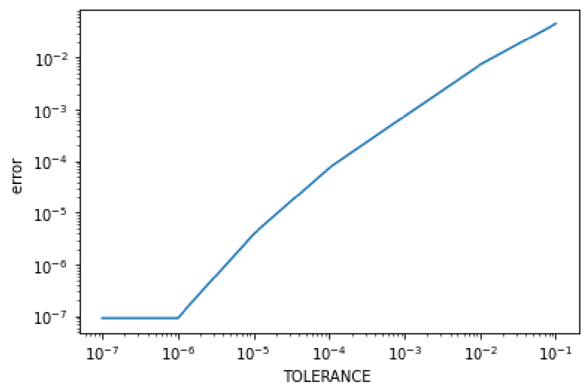


(f)

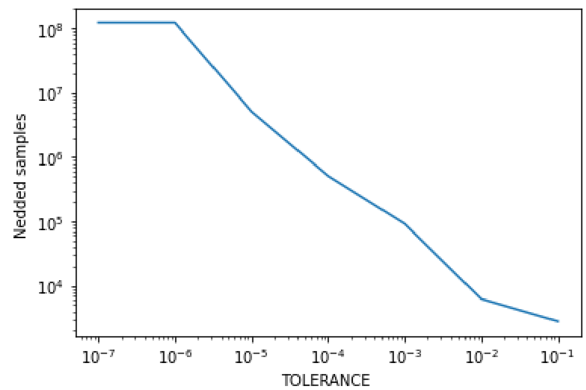
Payoff function 3

Figure 1: MC simulaton with the exact price for each payoff function on the left.The statistical error and the exact error on the right

Simulation results Exact error and Statistical error

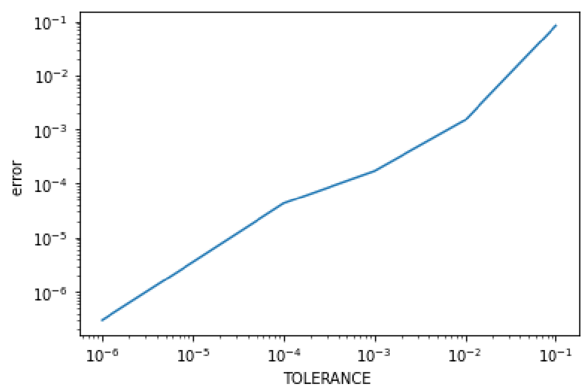


(a)

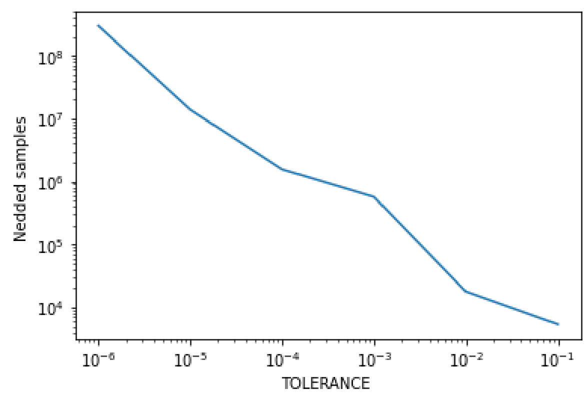


(b)

Payoff function 1

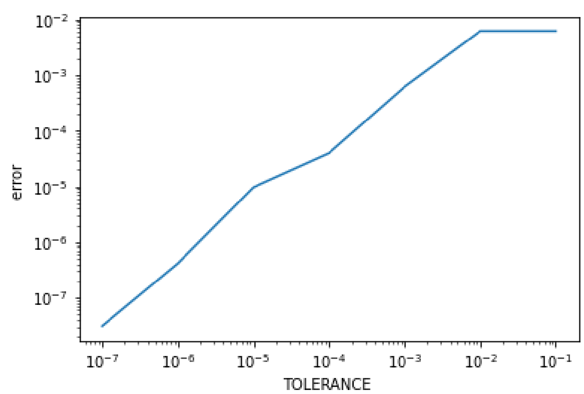


(c)

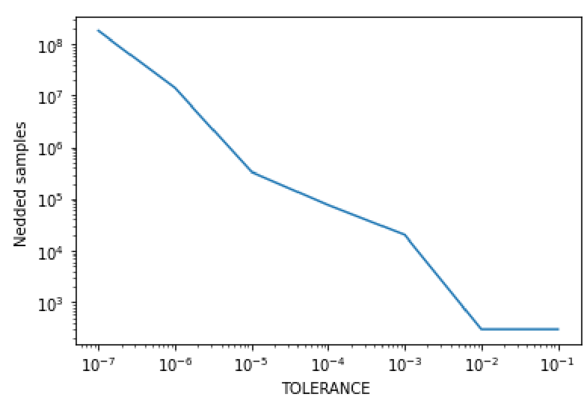


(d)

Payoff function 2



(e)



(f)

Payoff function 3

Figure 2: MC simulaton Error vs desired tolerance for each payoff function on the left.The needed number of Samples to achieve a certain toleranceon the right

Exercise 2

We want to solve Exercise 1 by instead approximating the corresponding Black-Scholes PDE

$$\partial_t f + rS\partial_s f + \frac{1}{2}\sigma^2 S^2 \partial_{ss} f - rf = 0$$
$$f(T, s) = g(s)$$

a.

Backward Euler

$$\partial_t f = \frac{f_i^n - f_i^{n-1}}{\Delta t} + \mathcal{O}(\Delta t)$$
$$\partial_s f = \frac{f_{i+1}^{n-1} - f_{i-1}^{n-1}}{2\Delta S} + \mathcal{O}(\Delta S^2)$$
$$\partial_{ss} f = \frac{f_{i+1}^{n-1} - 2f_i^{n-1} + f_{i-1}^{n-1}}{\Delta S^2} + \mathcal{O}(\Delta S^2)$$

with $f_i^n = f(n\Delta t, i\Delta S)$

So $\partial_t f + rS\partial_s f + \frac{1}{2}\sigma^2 S^2 \partial_{ss} f - rf = 0$ is written as:

$$\frac{f_i^n - f_i^{n-1}}{\Delta t} + \frac{ri}{2}(f_{i+1}^{n-1} - f_{i-1}^{n-1}) + \frac{\sigma^2 i^2}{2}(f_{i+1}^{n-1} - 2f_i^{n-1} + f_{i-1}^{n-1}) - rf_i^{n-1} = 0$$

$$f_i^n = f_i^{n-1}(1 + \sigma^2 i^2 \Delta t + r\Delta t) + f_{i+1}^{n-1}(-ri - \sigma^2 i^2)\frac{\Delta t}{2} + f_{i-1}^{n-1}(ri - \sigma^2 i^2)\frac{\Delta t}{2}$$

Finally

$$f_i^n = a_i f_{i-1}^{n-1} + b_i f_i^{n-1} + c_i f_{i+1}^{n-1}$$

with

$$a_i = (ri - \sigma^2 i^2)\frac{\Delta t}{2}$$
$$b_i = (1 + \sigma^2 i^2 \Delta t + r\Delta t)$$
$$c_i = (-ri - \sigma^2 i^2)\frac{\Delta t}{2}$$

So

$$f^n = Af_{n-1}$$

$$f_n = \begin{bmatrix} f_1^n \\ f_2^n \\ \vdots \\ f_i^n \\ \vdots \\ f_n^{M-1} \end{bmatrix} A = \begin{pmatrix} b_1 & c_1 & 0 & 0 & \cdots & 0 \\ a_2 & b_2 & c_2 & 0 & \cdots & 0 \\ 0 & a_3 & b_3 & c_3 & \cdots & 0 \\ \vdots & 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & 0 & \ddots & \ddots & c_{M-2} \\ 0 & 0 & \cdots & 0 & a_{M-1} & b_{M-1} \end{pmatrix}$$

Crank-Nicholson

We denote

$$\begin{aligned} f_i^n &= f(n\Delta t, i\Delta S) \\ f_i^{n-\frac{1}{2}} &= f((n-\frac{1}{2})\Delta t, i\Delta S) \end{aligned}$$

$$\partial_t f = \frac{f_i^n - f_i^{n-1}}{\Delta t} + \mathcal{O}(\Delta t^2)$$

$$\begin{aligned} \partial_s f &= \frac{1}{2}(\partial_s f_i^{n-1} + \partial_s f_i^n) \\ &= \frac{1}{2}(\frac{f_{i+1}^{n-1} - f_{i-1}^{n-1}}{2\Delta S} + \frac{f_{i+1}^n - f_{i-1}^n}{2\Delta S}) + \mathcal{O}(\Delta S^2) \end{aligned}$$

$$\begin{aligned} \partial_{ss} f &= \partial_{ss} f_i^{n-\frac{1}{2}} \\ &= \frac{1}{2}(\partial_{ss} f_i^{n-1} + \partial_{ss} f_i^n) \\ &= \frac{1}{2}(\frac{f_{i+1}^{n-1} - 2f_i^{n-1} + f_{i-1}^{n-1}}{\Delta S^2} + \frac{f_{i+1}^n - 2f_i^n + f_{i-1}^n}{\Delta S^2}) + \mathcal{O}(\Delta S^2) \end{aligned}$$

So $\partial_t f + rS\partial_s f + \frac{1}{2}\sigma^2 S^2 \partial_{ss} f - rf = 0$ is written as:

$$\frac{f_i^n - f_i^{n-1}}{\Delta t} + rS\frac{1}{2}(\frac{f_{i+1}^{n-1} - f_{i-1}^{n-1}}{2\Delta S} + \frac{f_{i+1}^n - f_{i-1}^n}{2\Delta S}) + \frac{1}{2}\sigma^2 S^2 \frac{1}{2}(\frac{f_{i+1}^{n-1} - 2f_i^{n-1} + f_{i-1}^{n-1}}{\Delta S^2} + \frac{f_{i+1}^n - 2f_i^n + f_{i-1}^n}{\Delta S^2}) - r\frac{f_i^n + f_i^{n-1}}{2} = 0$$

$$(-\frac{ri}{4} + \frac{\sigma^2 i^2}{4})f_{i-1}^n + (\frac{1}{\Delta t} - \frac{r}{2} - \frac{\sigma^2 i^2}{2})f_i^n + (\frac{ri}{4} + \frac{\sigma^2 i^2}{4})f_{i+1}^n = (\frac{ri}{4} - \frac{\sigma^2 i^2}{4})f_{i-1}^{n-1} + (\frac{1}{\Delta t} + \frac{r}{2} + \frac{\sigma^2 i^2}{2})f_i^{n-1} + (-\frac{ri}{4} - \frac{\sigma^2 i^2}{4})f_{i+1}^{n-1}$$

So

$$a_i f_{i-1}^n + b_i f_i^n + c_i f_{i+1}^n = -a_i f_{i-1}^{n-1} + d_i f_i^{n-1} - c_i f_{i+1}^{n-1}$$

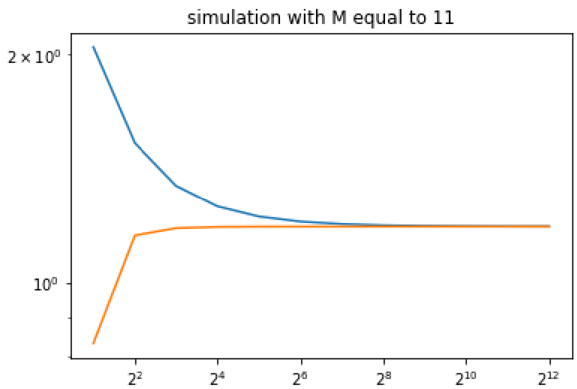
with

$$\begin{aligned} a_i &= (-\frac{ri}{4} + \frac{\sigma^2 i^2}{4}) \\ b_i &= (\frac{1}{\Delta t} - \frac{r}{2} - \frac{\sigma^2 i^2}{2}) \\ c_i &= (\frac{ri}{4} + \frac{\sigma^2 i^2}{4}) \\ d_i &= (\frac{1}{\Delta t} + \frac{r}{2} + \frac{\sigma^2 i^2}{2}) \end{aligned}$$

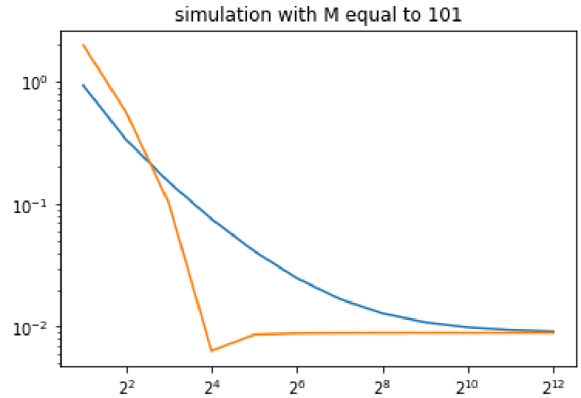
finally,

$$Af_n = Bf_{n-1}$$

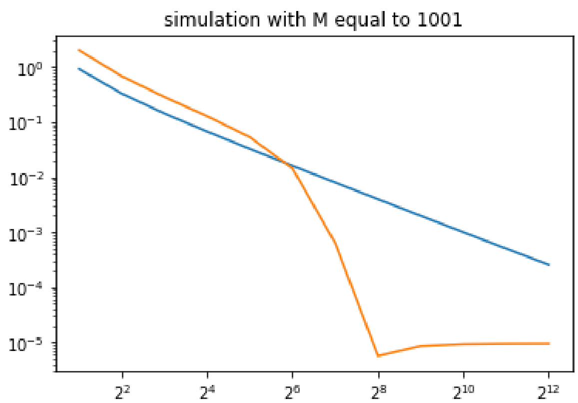
$$f_n = \begin{bmatrix} f_1^n \\ f_2^n \\ \vdots \\ f_n^i \\ \vdots \\ f_n^{M-1} \end{bmatrix} A = \begin{pmatrix} b_1 & c_1 & 0 & 0 & \cdots & 0 \\ a_2 & b_2 & c_2 & 0 & \cdots & 0 \\ 0 & a_3 & b_3 & c_3 & \cdots & 0 \\ \vdots & 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & 0 & \ddots & \ddots & c_{M-2} \\ 0 & 0 & \cdots & 0 & a_{M-1} & b_{M-1} \end{pmatrix} B = \begin{pmatrix} d_1 & -c_1 & 0 & 0 & \cdots & 0 \\ -a_2 & d_2 & -c_2 & 0 & \cdots & 0 \\ 0 & -a_3 & d_3 & -c_3 & \cdots & 0 \\ \vdots & 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & 0 & \ddots & \ddots & -c_{M-2} \\ 0 & 0 & \cdots & 0 & -a_{M-1} & d_{M-1} \end{pmatrix}$$



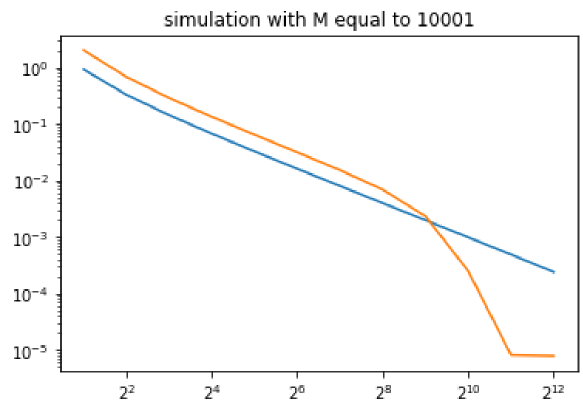
(a)



(b)



(c)



(d)

The error:

$$\epsilon = c_1 \Delta t^p + c_2 \Delta S^q \leq TOL$$
$$c_1 \Delta t^p = \frac{TOL}{2}$$
$$c_2 \Delta S^q = \frac{TOL}{2}$$

So

$$\Delta t = (\frac{TOL}{2c_1})^{\frac{1}{p}}$$
$$\Delta S = (\frac{TOL}{2c_2})^{\frac{1}{q}}$$

c1 and c2 are derived using the Richardson extrapolation The principle is as follow

$$\begin{cases} \bar{X}_{\Delta t} = X + a_1.\Delta t^p + a_2.\Delta s^r + h.o.t \\ \bar{X}_{\frac{\Delta t}{2}} = X + a_1.(\frac{\Delta s}{)}{}^p + a_2.(\frac{\Delta s}{2})^r + h.o.t \end{cases} \tag{1}$$

$$a_1 \approx \frac{X_{\Delta t} - X_{\frac{\Delta t}{2}}}{\Delta t^p - (\frac{\Delta t}{2})^p}$$

Using the same principle we can derive an approximation for the space constant and we get

$$a_2 \approx \frac{X_{\Delta s} - X_{\frac{\Delta s}{2}}}{\Delta s^p - (\frac{\Delta s}{2})^p}$$

b.
We want to prove that

$$Work(MC) = \mathcal{O}(TOL^{-3})$$
$$Work(PDE) = \mathcal{O}(TOL^{-\frac{d}{q}-\frac{1}{p}})$$

where d is the number of stock .
For MC Euler estimator:

$$\tilde{\pi} = e^{-rT} M^{-1} \sum_{m=1}^{M} g(\bar{S}_{\Delta t}^{(m)})$$

Where $\bar{S}_{\Delta t}$ is the numerical solution of the SDE by using the Euler method with time step Δt.
So the MC error: $\epsilon_{MC}(M, \Delta t) = \mathcal{O}(M^{\frac{-1}{2}} + \Delta t)$
The asymptotic average work for MC Euler is obtained by solving the minimization problem for a given tolerance $TOL > 0$

$$M^*, \Delta t^* = \arg\min_{M,\Delta t} M(\Delta t)^{-1}$$
$$\text{s.t} \quad M^{\frac{-1}{2}} + \Delta t < TOL$$

The Lagrangian with $\lambda > 0$

$$\mathcal{L}(M, \Delta t, \lambda) = M(\Delta t)^{-1} + \lambda(M^{\frac{-1}{2}} + \Delta t - TOL)$$
$$\frac{\partial \mathcal{L}}{\partial \Delta t} = -M(\Delta t)^{-2} + \lambda = 0 \Longrightarrow M = \lambda(\Delta t)^2$$
$$\frac{\partial \mathcal{L}}{\partial M} = (\Delta t)^{-1} - \frac{1}{2}\lambda M^{\frac{-3}{2}} = 0 \Longrightarrow \Delta t = \frac{2}{\lambda}(M)^{\frac{3}{2}}$$
$$\frac{\partial \mathcal{L}}{\partial M} = (\Delta t)^{-1} - \frac{1}{2}\lambda M^{\frac{-3}{2}} = 0 \Longrightarrow \Delta t = \frac{2}{\lambda}(M)^{\frac{3}{2}}$$
$$\frac{\partial \mathcal{L}}{\partial \lambda} = M^{\frac{-1}{2}} + \Delta t - TOL = 0$$

$\Longrightarrow$

$$\Delta t^* = \mathcal{O}(TOL)$$
$$M^* = \mathcal{O}(TOL^{-2})$$

So $Work^*(MCE) = M^*(\Delta t^*)^{-1} = \mathcal{O}(TOL^{-3})$

For the PDE:
The asymptotic average work for the PDE approach is obtained by solving the minimization problem for a given tolerance $TOL > 0$

$$\Delta S^*, \Delta t^* = \arg\min_{\Delta S,\Delta t} (\Delta S)^{-d}(\Delta t)^{-1}$$

s.t

$$\Delta t^p + \Delta S^q < TOL$$
$$\Delta t^p < \Delta S^q$$

The Lagrangien with $\lambda > 0$

$$\mathcal{L}(\Delta t, \Delta S, \lambda) = (\Delta S)^{-d}(\Delta t)^{-1} + \lambda(\Delta t^p + \Delta S^q - TOL)$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = (\Delta t^p + \Delta S^q - TOL) = 0$$

$\Longrightarrow$

$$\Delta S = \mathcal{O}(TOL^{\frac{1}{q}})$$
$$\Delta t = \mathcal{O}(TOL^{\frac{1}{p}})$$

So the asymptotic average work is

$$(\Delta S^*)^{-d}(\Delta t^*)^{-1} = \mathcal{O}(TOL^{\frac{-d}{q}})\mathcal{O}(TOL^{\frac{-1}{p}})$$

$$Work(PDE) = \mathcal{O}(TOL^{-\frac{d}{q}-\frac{1}{p}})$$

We can see that the work by the PDE approach suffers from the dimentionality curse as the work grows exponentially with d.
So when $d >> 1$ MC Euler method has a better assymptotic accuracy because $\frac{d}{q} + \frac{1}{p} > 3$
c.
MLMC
Let$\Delta t_\ell = \Delta t_0 2^{-\ell}, \ell = 0, \ldots, L$ and $\bar{X}\left(T; \Delta t_\ell\right)$ be a F. Euler approximation computed with uniform step Δt_ℓ and $g_\ell = g\left(\bar{X}\left(T; \Delta t_\ell\right)\right)$. We write the telescopic sum

$$\mathbb{E}\left[g_L\right] = \mathbb{E}\left[g_0\right] + \sum_{\ell=1}^{L} \mathbb{E}\left[g_\ell - g_{\ell-1}\right]$$

The Forward Euler has weak order 1

$$|E[g_\ell - g]| \leq \Delta t_\ell$$

andthe bias form considering levels up to L is to ensure $\left(E\left[g_L - g(X(T))\right] = \mathcal{O}(TOL)\right)$

$$TOL \leq \Delta t_L = \Delta t_0 2^{-L}$$
$$L = \mathcal{O}\left(\log_2\left(\frac{\Delta t_0}{TOL}\right)\right)$$

We use $L+1$ batches, each with M_ℓ independent realizations, $\ell = 0, \ldots, L$ to create the estimator

$$A = \frac{1}{M_0}\sum_{j=1}^{M_0} g_0\left(\omega_j^0\right) + \sum_{\ell=1}^{L}\frac{1}{M_\ell}\sum_{j=1}^{M_\ell}\left(g_\ell - g_{\ell-1}\right)\left(\omega_j^\ell\right)$$

And we compute its variance:

$$\operatorname{Var}(A) = \frac{\operatorname{Var}\left(g_0\right)}{M_0} + \sum_{\ell=1}^{L}\frac{\operatorname{Var}\left(g_\ell - g_{\ell-1}\right)}{M_\ell}$$

Since F. Euler has strong order 1/2 we have

$$\operatorname{Var}(A) = \frac{\mathcal{O}(1)}{M_0} + \sum_{\ell=1}^{L}\frac{\mathcal{O}\left(\Delta t_\ell\right)}{M_\ell}$$

Now we take

$$\frac{\Delta t_\ell}{M_\ell} = \frac{\Delta t_0}{M_0}, \ell = 1, \ldots, L$$

This implies $M_\ell = M_0 2^{-\ell}$ and

$$\operatorname{Var}(A) = \frac{\mathcal{O}(1)}{M_0}\left(1 + \Delta t_0 L\right)$$

To achieve accuracy TOL we need $\mathrm{Var}(A) = \mathcal{O}\left(TOL^2\right)$ and thus

$$\begin{aligned} M_0 &= \mathcal{O}\left(TOL^{-2}\left(1 + \Delta t_0 L\right)\right) \\ &= \mathcal{O}\left(TOL^{-2}\left(1 + \Delta t_0 \log_2\left(\Delta t_0/TOL\right)\right)\right) \end{aligned}$$

So the computational work of computing the estimator A to achieve a given accuracy TOL

$$\begin{aligned} Work &= \sum_{\ell=1}^{L} \frac{M_\ell}{\Delta t_\ell} \\ &= (1+L)\frac{M_0}{\Delta t_0} \\ &= \mathcal{O}\left((1+\left(\log_2\left(\frac{\Delta t_0}{TOL}\right)\right))(1+\left(\log_2\left(\frac{\Delta t_0}{TOL}\right)\right)\frac{TOL^{-2}}{\Delta t_0})\right) \\ &= \mathcal{O}\left(\left(\log_2\left(\frac{\Delta t_0}{TOL}\right)TOL^{-2}\right)\right) \end{aligned}$$

So the Multi Level F. Euler Monte Carlo method improves the cost from $\mathcal{O}(TOL^{-3})$ to $\mathcal{O}((log(TOL)TOL^{-1})^2)$