Imperial College London

Simulation Methods for Barrier and Look-back Options

Simulation Methods for Finance
under the supervision of Professor Harry Zheng

Authors:
Thomas Espel
Konstantin Kulak
Callum MacIver
Zhentian Qiu
Vera Zhang

March 11, 2018

Contents

I Basic Task **1**

1 Generate Random Number **1**
1.1 Presentation 1
1.2 Generating the uniform distribution 1
1.3 Generating the normal distribution 1

2 European Option **2**
2.1 Presentation 2
2.2 Likelihood Ratio Greeks 3
2.3 Pathwise Derivative Greeks 3

II Main Task **6**

3 Barrier Option **6**
3.1 Presentation 6
3.2 Another Method using Rayleigh Distribution 6

4 Look-back Option **8**

Appendix **10**

A Closed-formed formula for Barrier Options **10**

B App User Guide **11**

C Package Manual **12**

Part I
Basic Task

1 Generate Random Number

1.1 Presentation

There are two methods we used to generate random numbers. The first one is to generate a uniform distribution and then transform it into Standard Normal Distribution. The second method we tried is the system build-in function `Random`, which can directly generate a variable follows Standard Normal Distribution.

1.2 Generating the uniform distribution

To generate a uniform distribution we tried two ways: one can use the system build-in methods `rand`. The function will return a number between 1 and $2^{15} - 1$ randomly. The range is around 30,000, way below 100,000, the sample size we plan to generate. In other words, when we use `rand` to simulate 100,000 entries, there will be numbers appear more than once, which will create bias. We prefer the second way to generate a uniform distriubtion: linear congruential generator.

$$n_i = (an_{i-1})\ mod\ m$$

for $i = 1, 2, ..., 10,000$, and we let $a = 7^5$, and $m = 2^{31} - 1$, which gives about 2 billion points. We run 100,000 times and will only use 0.005% of all points. Theoretically, no pattern should appear.

1.3 Generating the normal distribution

To transform the uniform distribution we got into Standard Normal Distribution, we have tried three methods. By Central Limit theory,

$$Z_n = \frac{\sum_{i=1}^{n} X_i - n\mu}{\sqrt{n}\sigma}$$

where X_i is from the uniform distribution we generated previously. Z_n converges in distribution to SND, however, it requires n, the number of uniform distribution, to be sufficiently large. Therefore this method requires significant large simulations and speed is slow consequently.

We also tried Box-Muller methods [1]

$$Z_1 = \sqrt{-2lnX_1}sin(2\pi X_2),\ \mathbb{Z}_2 = \sqrt{-2lnX_1}cos(2\pi X_2)$$

Since its simulation involves computation of *sine* and *cosine*, the speed is slow. We prefer the last method, Marsaglia Polar method.

$$let\ V_1 = 2U_1 - 1,\ V_1 = 2U_2 - 1.$$

where U_1 and U_2 are two independent uniform distribution. Let $W = V_1^2 + V_2^2$. If $W > 1$, return to the beginning. Otherwise,

$$N_1 = \sqrt{\frac{(-2logW)}{W}}V_1,\ N_2 = \sqrt{\frac{(-2logW)}{W}}V_2$$

As the computation does not involve sine and cosine, it is generally faster than Box-Muller.

Method	Mean	Variance	Time
***rand* + CLT**	1.77 e-5	0.999909	90.34
***rand* + Box-Muller**	-6.80 e-5	1.000703	8.64
***rand* + Marsagilia Polar**	1.70 e-5	1.000472	6.40
LCG + CLT	1.52 e-5	0.999973	236.3
LCG + Box-Muller	-1.27 e-4	0.999797	11.59
LCG + Marsagilia	7.36 e-5	0.999979	10.52
`random`	3.58 e-4	1.000210	1.498

Table 1: We note that the Marsaglia method is faster compared to the other methods. Note that the run time depends on the computer used for the simulation (here a 3.4 GHz Intel Core i7). We have generated 100,000,000 variables to obtain these results.

The following is the simulation result of Standard Normal distribution and time used by different methods from generating 100,000,000 random variables.

As shown in the table above, when using CLT generating standard normal distribution, the time needed is significantly longer than the rest methods. We use the combination of linear congruential generator and Marsagilia methods to simulate random variables. This was also the opportunity for us to have full control over the generation of random numbers, as it is a critical step towards efficient simulations [1]. It also features very good results in terms of statistical parameters and time.

2 European Option

2.1 Presentation

The asset price in a risk neutral probability space $(\Omega, \mathcal{F}, (\mathcal{F}_t)_{0\leq t\leq T}, P)$ follows Geometric Brownian Motion,

$$dS_t = rS_t dt + \sigma S_t dW_t,\ 0 \leq t \leq T$$

with initial price $S_0 = S$, where r is riskless interest rate, σ volatility, and W_t the standard Brownian motion. A European call option price at time t with maturity time T is given by

$$C_t = E[e^{-r(T-t)}(S_T - K)^+ | F_t]$$

For the basic task, we let $S_0 = 100$, $K = 100$, interest rate $r = 0.05$, volatility $\sigma = 0.4$, maturity time $T = 1$. We use Monte Carlo method to simulate the path of

$$S_T = S_0 e^{(r-\frac{1}{2}\sigma^2)T+\sigma\sqrt{T}Z}$$

and get sample distribution of $(S_T - K)^+$. By Black-Scholes formula,

$$C_{bs}(S_0, K) = N(d_1)S_0 - N(d_2)Ke^{-rT}$$

where

$$d_1(S_0, K) = \frac{1}{\sigma\sqrt{T}}[ln(\frac{S_0}{K}) + (r + \frac{\sigma^2}{2})T],\ d_2(S_0, K) = d_1 - \sigma\sqrt{T}$$

The closed-form Greeks are calculated the following way:

$$\delta_{bs} = \frac{\partial C}{\partial S_0} = \Phi(d_1), \quad \gamma_{bs} = \frac{\partial^2 C}{\partial S_0^2} = \frac{\Phi'(d_1)}{S_0\sigma\sqrt{T}}, \quad \nu_{bs} = \frac{\partial C}{\partial \sigma} = \Phi'(d_1)\sqrt{T}$$

2.2 Likelihood Ratio Greeks

To calculate the Greeks from simulation, we compare Likelihood ratio method and Pathwise method. As the Call option price is given by

$$C = e^{-rT}E[(S_T - K)^+] = e^{-rT}\int (S_T - K)^+ h_{S_0}(S_T)dS_T$$

where $h_{S_0}(S_T)$ is the probability density function of $(S_T - K)^+$. Then by Likelihood ratio methods, the partial dirivative of C with respect to S_0 is

$$\frac{\partial C}{\partial S_0} = \int (S_T - K)^+ \frac{d}{dS_0} h_{S_0}(S_T)dS_T = E[(S_T - K)^+ \frac{h'_{S_0}(S_T)}{h_{S_0}(S_T)}]$$

And the lognormal density function of S_T is given by

$$h(x) = \frac{1}{x\sigma\sqrt{T}}\phi(\xi(x)), \quad \xi(x) = \frac{ln(x/S_0) - (r - \frac{1}{2}\sigma^2)T}{\sigma\sqrt{T}}$$

Therefore [1],

$$Delta_{LR} = \frac{\partial C}{\partial S_0} = E[e^{-rT}(S_T - K)^+ \frac{Z}{S_0\sigma\sqrt{T}}]$$

where Z $N(0,1)$.

$$Gamma_{LRLR} = \frac{\partial^2 C}{\partial S_0^2} = E[e^{-rT}(S_T - K)^+ (\frac{Z^2 - 1}{S_0^2\sigma^2 T} - \frac{Z}{S_0^2\sigma\sqrt{T}})]$$

$$Vega_{LR} = \frac{\partial C}{\partial \sigma} = E[e^{-rT}(S_T - K)^+ \left(\frac{Z^2 - 1}{\sigma} - Z\sqrt{T}\right)]$$

2.3 Pathwise Derivative Greeks

By pathwise methods, the greeks are given as following[1] [1], with $Z \sim N(0,1)$:

$$Delta_{PW} = \frac{\partial C}{\partial S_0} = E[e^{-rT}\mathbb{1}_{S_T > K}\frac{S_T}{S_0}]$$

$$Gamma_{LRPW} = \frac{\partial^2 C}{\partial S_0^2} = E[e^{-rT}\mathbb{1}_{S_T > K}\frac{KZ}{S_0^2\sigma\sqrt{T}}]$$

$$Gamma_{PWLR} = \frac{\partial^2 C}{\partial S_0^2} = E[e^{-rT}\mathbb{1}_{S_T > K}\frac{S_T}{S_0^2}\left(\frac{Z}{\sigma\sqrt{T}} - 1\right)]$$

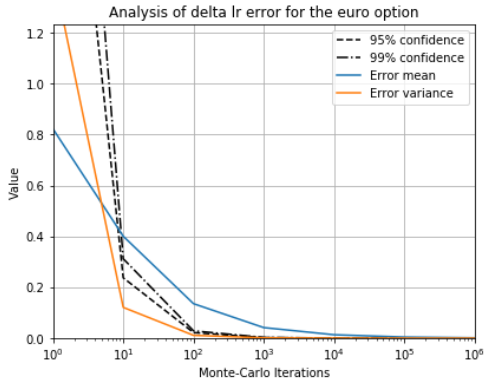
$$Vega_{PW} = \frac{\partial C}{\partial \sigma} = E[e^{-rT}\mathbb{1}_{S_T > K}S_T(\sqrt{T}Z - \sigma T)]$$

The results calculated from the closed-form formulae and the simulations are rempresented in Table 2.

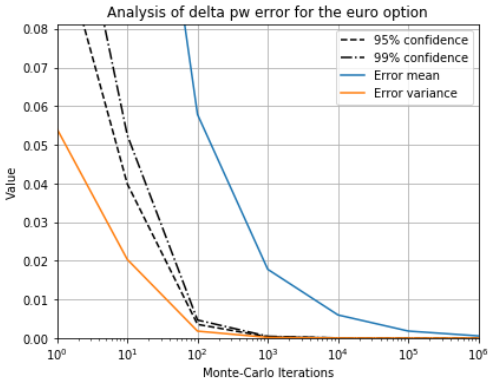
We can see from above table that simulation results improve and get closer to the theoretical value as simulation number increase. Hence to compare the results from different methods, we will only analyze the simulation result from the most simulation number - 100,000 in this case. This is even more clear on the graphs (Figure 1).

We now have a closer look at 100,000 simulations, which is the industry standard and which is above the 1,000 simulations threshold we have noticed on the graphs. For delta, although the fastest methods is LR, it has a significantly lower accurany. Gamma LRLR is the most

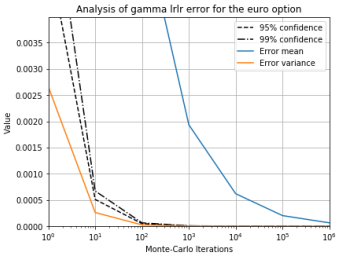
[1]For gamma, there are two different methods.



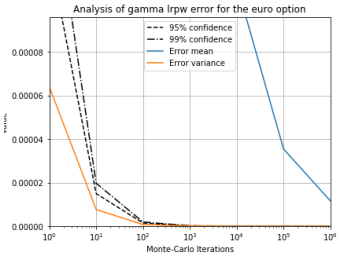
(a) Delta with LR method



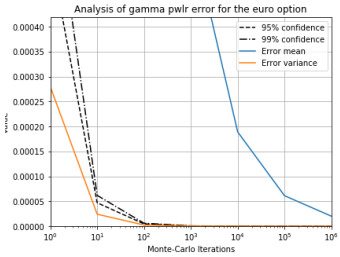
(b) Delta with PW method



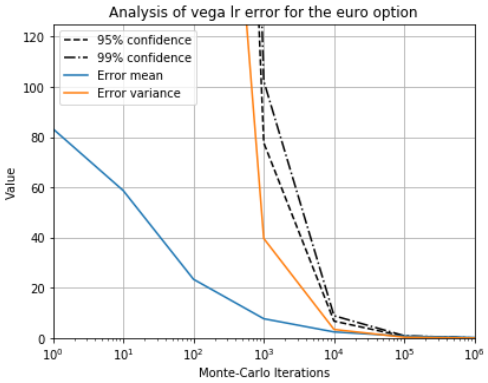
(c) Gamma with LR-LR method



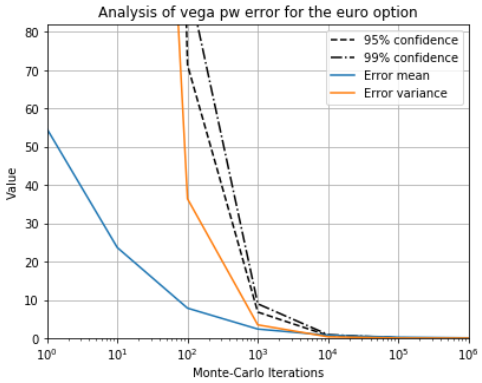
(d) Gamma with LR-PW method



(e) Gamma with PW-LR method



(f) Vega with LR method



(g) Vega with PW method

Figure 1: A graphical analysis of the error of the greeks depending on the number of simulations. We notice that there is a significant increase above 1000 simulations in all cases. This is very clear on our vega which tends to have a very high error variance below 1000 simulations. In order to generate these graphs, we have donne one thousand simlations for each value, at every power of 10.

		Black -Scholes	Monte Carlo Simulation		
			1,000	**10,000**	**100,000**
Option Price		18.023	19.0162	17.6732	18.0014
Delta	**LR**	0.627409	0.662561	0.612362	0.627889
	PW		0.64485	0.623239	0.627453
Gamma	**PW LR**	0.0094605	0.0101213	0.00919597	0.00945029
	LR PW		0.00994418	0.00930474	0.00944593
	LR LR		0.00975465	0.00918059	0.0095502
Vega	**LR**	37.842	39.0186	36.7224	38.2008
	PW		40.4851	36.7839	37.8012

Table 2: Table with the result of our simulations for different Monte-Carlo iterations. We can clearly see the improvement between 1,000 and 10,000 simulations.

1 000 000 MC simulations	**Error Mean**	**Error Variance**	**Time (seconds)**
Option Price	na	na	5.0 e-6
Delta LR	4.1 e-3	9.7 e-6	1.4 e-3
Delta PW	1.8 e-3	1.8 e-6	9.9 e-4
Gamma PWLR	6.1 e-5	2.0 e-9	1.4 e-3
Gamma LRPW	3.5 e-5	7.4 e-10	1.4 e-3
Gamma LRLR	1.9 e-4	2.1 e-8	4.7 e-3
Vega LR	7.7 e-1	3.3 e-1	4.6 e-3
Vega PW	2.4 e-1	3.2 e-2	1.6 e-3

Table 3: Sample from our simlation dataset with 100 000 Monte-Carlo simlations. Note that the run time depends on the computer (here a 3.1 GHz Intel Core i5). We computed the absolute error and since the option price is computed using the closed form formula no error is represented.

accurate but it is the slowest, so gamma PWLR seems to be a good compromise. Vega PW is both faster and more accurate than vega LR.

For the following section, PW methods will be infeasible to do for Barrier Option. Hense, we will use Likelihood ratio method when calculating the greeks.

Part II
Main Task

3 Barrier Option

3.1 Presentation

Let T denote option expiration time and $[0, T]$ lookback period. For $T_0 \leq t \leq T$ denote by

$$m_0^T = \underset{0\leq t\leq T}{min} S_t, \quad M_0^t = \underset{0\leq t\leq T}{max} S_t$$

For an up-and-out barrier call option $A_T = (S_T - K)^+ 1_{max 0\leq t\leq T\ S_t\leq B}$, where B is a barrier leve and 1_S is an indicator function.

We stimulate the path of S_t by taking the maturity time T into 1,000 steps, and we simulate 5,000 such paths. As standard normal distribution is symmetrically distributed, 5,000 paths can be treated as 10,000 paths by adding negative sign and creating the other 5,000. We have an M_0^T for each path. However, this method is bordensome as we need to generate each step for each path. The simulation process is long.

3.2 Another Method using Rayleigh Distribution

Hense, we tried second method using Rayleigh Distribution to simulate the distribution of M_0^t directly. The maximum of a standard Brownian motion starting at the origin to be at b at time 1 over period [0, T] has the Rayleigh distribtuion [2, 3, 4]

$$F(x) = 1 - e^{-2x(x-b)},\ x \geq b.$$

solving the equation $F(x) = u, u \in (0, 1)$ has roots

$$x = \frac{b}{2} \pm \frac{\sqrt{b^2 - 2log(1-u)}}{2}$$

Hense, at time T with S_T,

$$M^T = \frac{S_T + \sqrt{S_T^2 - 2TlogU}}{2}$$

And we simulate S_T by Black-Scholes formula

$$S_T = S_0 e^{(r-\frac{1}{2}\sigma^2)T+\sigma\sqrt{T}Z}$$

Therefore,

$$M^T = S_0 e^{\frac{1}{2}log(\frac{S_T}{S_0})+\sqrt{log(\frac{S_T}{S_0})^2-2\sigma^2 TlogU}}$$

With this method, we only need to generate one S_T for each path and one random variable for U to get the M^T. It reduces the amount of simulation by 500 times (as before, 1,000 S_t need to be generated for each path).

As we mentioned previously, it is impossible to find the partial direvative of the indicator function $1_{M_0^T\leq B}$ with respect to S_0. Hense pathwise method is eliminated by us. For the likelihood ratio method, we find the differentiation of joint cdf of S_T and the M_T to be

$$f_{uo}(x, m, T) = \frac{1}{\sqrt{T}}\left(\phi(\frac{x-\mu T}{\sqrt{T}}) - e^{2m\mu}\phi(\frac{x-2m-\mu T}{\sqrt{T}})\right)$$

100 000 MC simulations	**Error Mean**	**Error Variance**	**Time (seconds)**
Option Price	na	na	..
Delta LR	1.036 e-2	3.029 e-5	9.284 e-2
Gamma LRLR	1.895 e-4	1.895 e-8	1.475 e-1
Vega LR	7.507 e-1	3.220 e-1	1.097 e-1

(a) Error statistics and computation time for the "classical" method. We computed the absolute error and since the option price is computed using the closed form formula no error is represented.

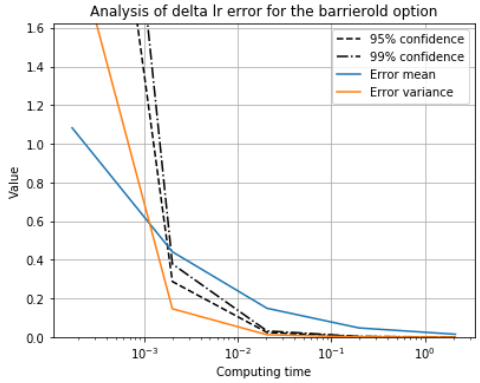
1 000 000 MC simulations	**Error Mean**	**Error Variance**	**Time (seconds)**
Option Price	PLEASE PASTE NEW	na	..
Delta LR	1.036 e-2	3.029 e-5	9.284 e-2
Gamma LRLR	1.895 e-4	1.895 e-8	1.475 e-1
Vega LR	7.507 e-1	3.220 e-1	1.097 e-1

(b) Error statistics and computation time for the Rayleight method. We computed the absolute error and since the option price is computed using the closed form formula no error is represented.

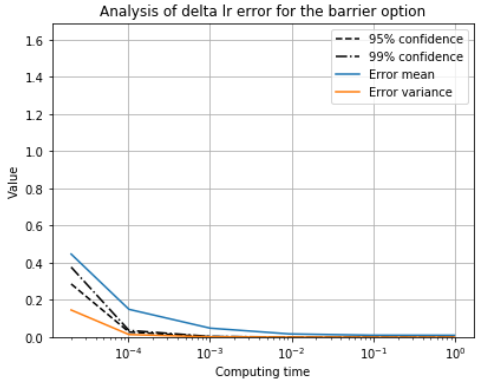
Table 4: Sample from our simulation dataset with the new fast method for barrier option simulation. It is clear that the results have significantly improved compared with the previous method. note that the run time depends on the computer used (here a 3.4 GHz Intel Core i7).

where $x = \frac{1}{\sigma} ln \frac{x}{S_0}$, $\mu = \frac{1}{2}(r - \frac{\sigma^2}{2})$ and $m = \frac{1}{\sigma} ln \frac{M_T}{S_0}$. Then we can find the greeks accordingly, the detail of the computation is in Appendix A.

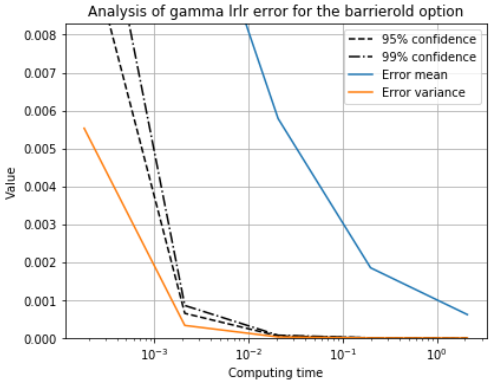
To analyze further the accuracy of our simulation, we plot the simulated option prices and greaks versus their theoretical value as barrier increases 2.



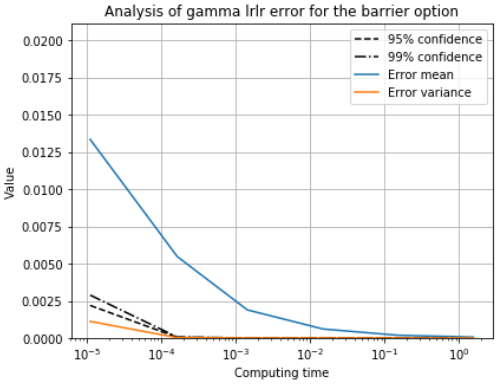
(a) Delta with LR method



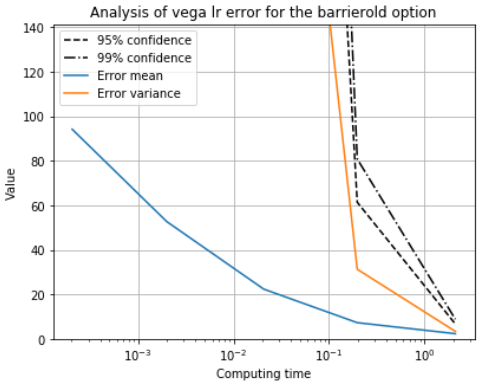
(b) Delta with Rayleight method



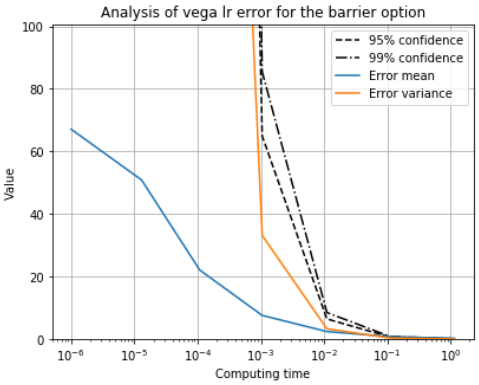
(c) Gamma with LRLR method



(d) Gamma with Rayleight method



(e) Vega with LR method



(f) Vega with Rayleight method

Figure 2: A graphical analysis of the error of the greeks depending on the computation time. We notice that the method with Rayleight distribution is between 100 and 1,000 times faster. In order to generate these graphs, we have donne one thousand simlations for each value, at every power of 10. However, due to the extremely slow computation time for LR methods, we only computed up to 100,000 samples compared to 1,000,000 for the Rayleight method.

4 Look-back Option

Lookback call option with fixed strik price K has payoff $(M_0^T - K)^+$. The call option price at time t is

$$c(S_0, K, t) = e^{-r(T-t)} E[(max(M_0^t, M_t^T) - K)^+ | \mathcal{F}_t]$$

	Closed-form Formula	Mean	Error	Variance	Time
Option Price					
Delta					
Gamma					
Vega					

Table 5

The closed-form formula for fixed strike lookback call option at time 0 [1, 4] is

$$c(S_0, K, 0) = C_{bs}(S_0, K) + \frac{S_0\sigma^2}{2r}\{\Phi[d_2(S_0, K)] - e^{-rT}\frac{S_0}{K}^{-\frac{2r}{\sigma^2}}\Phi[-d_1(K, S_0)]\}$$

The pdf of the distribution of Maximum S_t for standard Brownian motion for the period of (0, T) is

$$f = \frac{2}{\sqrt{T}}\phi\left(\frac{m - aT}{\sqrt{T}}\right) - 2ae^{2am}\Phi\left(\frac{-m - aT}{\sqrt{T}}\right)$$

and the cdf is

$$F = \Phi\left(\frac{m - aT}{\sqrt{T}}\right) - e^{2am}\Phi\left(\frac{-m - aT}{\sqrt{T}}\right)$$

We used Newton-Raphson method to solve for the above cdf F = U. Hense, for each random number we generate from [0,1], we receive one m in by solving for F. And to get M^T of stock price follows geometric Browninan motion with starting price S_0, we let $M^T = S_0e^{\sigma m}$.

References

[1] Harry Zheng (Pr.), *Simulation Methods for Finance*, Imperial College London, London, 2018.

[2] Diogo Monteiro da Costa Soares Justino, *Hedging of Barrier Options*, Instituto Universitário de Lisboa, Lisbon, 2010.

[3] Paul Glasserman (Pr.), *Monte Carlo Methods in Financial Engineering*, Springer Science, New York, 2013.

[4] Peter G. Zhqng (Pr.), *Exotic Options, A Guide to Second Generation Options*, World Scientific Publishing, Hong Kong, 1998 (second edition).

Appendix

A Closed-formed formula for Barrier Options

$$UOC(S_0,K,B) = 1_{B>K}\{C_{bs}(S_0,K) - C_{bs}(S_0,B) - (B-K)e^{-rT}\Phi[d_1(S_0,B)] \\ - \frac{B}{S_0}^{\frac{2v^2}{\sigma^2}}\left[C_{bs}(\frac{B^2}{S_0},K) - C_{bs}(\frac{B^2}{S_0},B) - (B_0-K)e^{-rT}\Phi[d_1(S_0,B)]\right]\} \quad (1)$$

Where C_{bs} and d_1 are as stated in the Black-Scholes formula (1) and (2), and $v = r - \frac{\sigma^2}{2}$.

closed form for DOC price is

$$C_{DO}(S_0,K,B) = C_{bs}(S_0,K) - (\frac{S_0}{B})^{-2\frac{v}{\sigma^2}}C_{bs}(\frac{B^2}{S_0},K)$$

where $v = r - \frac{\sigma^2}{2}$.

$$\begin{aligned}\delta_D OC =& \delta_{BS}(S_0,K) - \delta_{BS}(S_0,B) - \frac{B-K}{\sigma S_0\sqrt{T}}e^{-rT}\Phi(d_2(S_0,B)) \\ &+ \left(\frac{2v}{\sigma^2 S_0}\left(\frac{B}{S_0}\right)^{2v/\sigma^2}\right) \\ &\times \left(C_{BS}\left(\frac{B^2}{S_0},K\right) - C_{BS}\left(\frac{B^2}{S_0},B\right) - (B-K)e^{-rT}\Phi(d_2(B,S_0))\right) \\ &- \left(\frac{B}{S_0}\right)^{2v/\sigma^2}(\left(\frac{-B}{S_0}\right)^2\delta_{BS}\left(\frac{B^2}{S_0},K\right) + \left(\frac{B}{S_0}\right)^2\delta_{BS}\left(\frac{B^2}{S_0},B\right) \\ &+ \left(\frac{B-K}{\sigma S_0\sqrt{T}}\right)e^{-rT}\Phi(d_2(B,S_0)));\end{aligned}$$

$$\begin{aligned}\delta_{DOC} =& \Phi\left(\frac{\log\frac{S_0}{K} + (r+\frac{\sigma^2}{2})T}{\sigma\sqrt{T}}\right) - \left(\frac{B}{S_0}\right)^{r/\sigma^2-1} \\ &\times\left(-\frac{B}{S_0}^2\Phi\left(\frac{log\frac{B^2}{S_0K} + vT}{\sigma\sqrt{T}} + \sigma\sqrt{T}\right) - \frac{2vC_{BS}(B^2/S_0,K)}{(S_0\sigma^2)}\right)\end{aligned}$$

$$\gamma_{DOC} = \frac{\phi(d_2)}{S_0\sigma\sqrt{T}} - \left(\frac{B}{S_0}\right)^{2v/\sigma^2}\left(\frac{4v^2+2v\sigma^2}{S_0^2\sigma^4}C_{BS}(B^2/S_0,K) + \gamma_{bs} - \frac{4v\delta_{bs}}{S_0\sigma^2}\right)$$

$$\nu_{DOC} = S_0\phi\left(\frac{log(\frac{S_0}{K}) + (r+\frac{\sigma^2}{2})T}{\sigma\sqrt{T}}\right)\sqrt{T} - \left(\frac{B}{S_0}\right)^{\frac{2v}{\sigma^2}}\left(\nu_{bs} - \frac{4rC_{bs}(\frac{B^2}{S_0},K)log(\frac{B}{S_0})}{\sigma^3}\right)$$

B App User Guide

C Package Manual