

Report for HW3

Part 1 Implementation of algorithm:

For a), the formula is exactly the same as in HW1. The Black-Scholes formula is as below:

$$C(S, t) = N(d_1) S - N(d_2) K e^{-r(T-t)},$$
$$d_1 = \frac{\ln\left(\frac{S}{K}\right) + \left(r + \frac{\sigma^2}{2}\right)(T-t)}{\sigma\sqrt{T-t}}$$
$$d_2 = \frac{\ln\left(\frac{S}{K}\right) + \left(r - \frac{\sigma^2}{2}\right)(T-t)}{\sigma\sqrt{T-t}} = d_1 - \sigma\sqrt{T-t}.$$

The function is from Joshi code.

For b), the random generator is replaced with Park Miller random generator from Joshi's code. Park Miller sequence are uniform distributed random numbers. We get the inverse normal function to get back a normally distributed random numbers and use that as the input of path generation. The code is borrowed from Joshi. He build a class randomParkMiller and build it as an extension of the RandomBase class an abstract class where all basic methods, including getGaussian, are defined. This getGaussian method is the most important one and is what is used in the ExoticEngine classes to generate spot path. It is also called in either ExoticBSEngine, which as the name suggests is an extension of ExoticEngine class.

For c), what I did was to take off the Antithetic methods in previous question. Antithetic is an extended class of RandomBase method. It specifies the sampling as alternating sequence where the second one is the negation of the first. Basically, it overrides the methods defined in RandomBase, especially the GenUniform method. I put it on in this question and this is only a line in the main function but it actually takes a whole class in the code.

For d), I created another class SobolGenerator with similar log but implementing Sobol sequence instead of Park Miller sequence. I also overrode the methods inherited from RandomBase and eliminated the Inner class of Park Miller class. One thing to note is that I still call the GenUniform function GenUniforms because the class is called in ExoticBSEngine through this function. The name seems confusing as I didn't name it GenSobol but if I do that, I will have to rewrite a new ExoticBSEngine_Sobel class again. I don't want to do that.

Part 2 Benchmark(Note that this is not for the question about changing Barrier price. There will be another chart showing that.)

	My Code with standard 10000 simulations (mean/std-dev)	Haug's VBA Code
Closed-form Vanilla call option price	10.0201	10.0201
MC vanilla call price with Park-Miller uniforms	9.95932/20.3121	NA
MC vanilla call price with Park-Miller uniforms and antithetics	9.9824/20.2609	9.9861
QMC vanilla call price with Sobol sequence	9.99624/19.4387	10.0151

It appears the Park Miller uniforms with antithetic method is not as good as QMC in this case. So, QMC do improves the result in this case. But, as I test another with the strike price changed to 50 for comparison, here is result table:

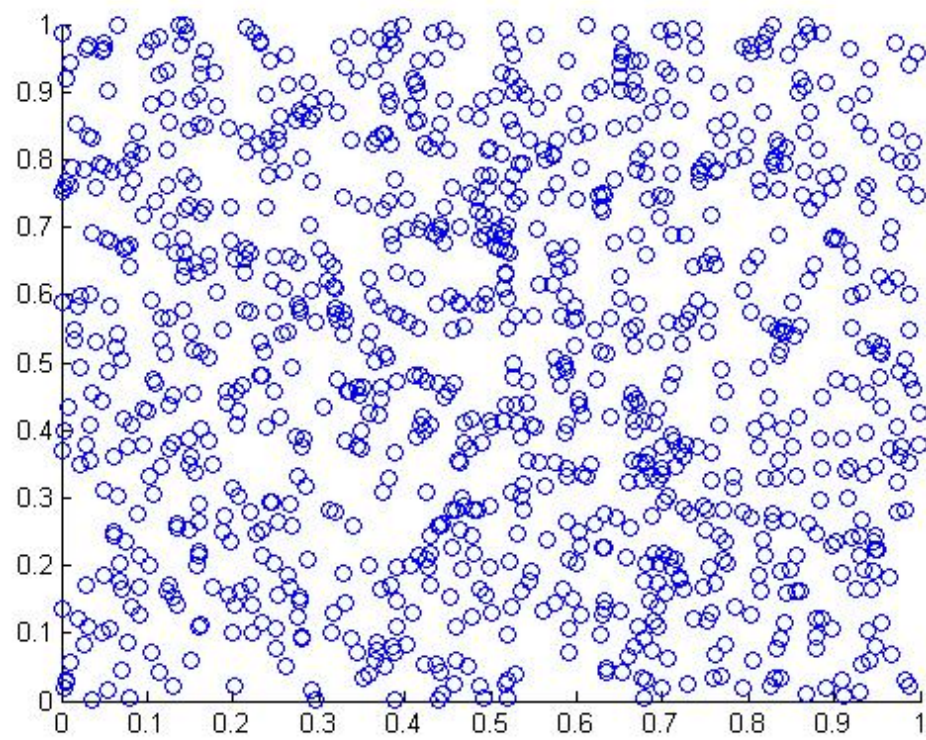
	My Code with standard 10000 simulations	Haug's VBA Code
Closed-form Vanilla call option price	52.4826	52.4826

MC vanilla call price with Park-Miller uniforms	52.487/31.2106	NA
MC vanilla call price with Park-Miller uniforms and antithetics	52.4596/30/9912	52.4826
QMC vanilla call price with Sobol sequence	52.455/30.56	52.3881

In this case, MC with antithetic and plain MC is almost as good as Sobol sequence. So, we can not say QMC is panacea for everything. We can only say QMC gives nice result in some cases.

Part 3 Answers to the report Questions:

1. The model is designed to put more weight on certain area in order to best fit the distribution the process is modeling. In our case, it is normal distribution. So, we have to apply transformation to make it so. The Sobol sequence is essentially providing a close sequence to uniform distribution. So, if we apply this sequence directly, we would have assumed the process is uniform rather than normal. This violates the model assumption.
2. The graphs are attached below: The first one is Park Miller pairs and the other is Sobol in 2 dimension.



3.

