# Fast Valuation of Derivative Contracts by Simulation

Jin-Chuan Duan    Geneviève Gauthier    Jean-Guy Simonato*

(May 28, 1999)

**Abstract**

Monte Carlo simulations are commonly used for valuing derivative securities, particularly for the contracts with a high degree of path dependency. The Monte Carlo method is, however, known to be computationally intensive. Accelerating the Monte Carlo method has thus long become an issue of significant academic interest and of great practical relevance. In this article a combination of two existing ways of accelerating simulation is proposed. Incorporating the recently developed empirical martingale adjustment into the increasingly popular quasi-Monte Carlo simulation (also known as the low-discrepancy simulation) is found to deliver by far the best performance.

*Duan is at the Department of Finance, Hong Kong University of Science and Technology. Gauthier and Simonato are at École des Hautes Études Commerciales, Montréal. We acknowledge the financial support from the Research Grants Council of Hong Kong, the Social Sciences and Humanities Research Council of Canada (SSHRC) and Les Fonds pour la Formation de Chercheurs et l'Aide à la Recherche du Québec (FCAR). Duan also acknowledges the support received as Senior Wei Lun Fellow at the Hong Kong University of Science and Technology. The program FINDER for generating the Sobol sequence is provided to us by A. Papageorgiou, courtesy of Columbia Innovation Enterprise, Columbia University. The earlier version of this paper was titled, "Monte Carlo Simulation - A Quick Fix."

# 1 Introduction

Many derivative securities exhibit some degree of path dependency. The financial market abounds with such contracts; for example, Asian options, lookback options, knock-in and -out options, collateralized mortgage obligations, among many others. This type of path dependency is created by contract specifications which explicitly call for the payout to be a function of the past prices of the underlying instrument. To value a derivative security, one also needs to make a choice of pricing framework. The adopted framework may rely on a price dynamic that is highly path dependent. The increasingly popular GARCH model is one such example. Since path dependency arises from the underlying asset price, it therefore stays with us regardless of whether the derivative contract is path dependent or not. In short, path dependency is often intrinsic to the tasks facing analysts. Some of path dependency can be simplified and dealt with in the lattice or other numerical valuation frameworks. Inevitably, there are valuation problems with a high degree of path dependency, and the Monte Carlo method becomes the method of last resort.

Using Monte Carlo simulations to value derivatives is not without problems. The method can be computationally demanding if a reasonably high level of pricing accuracy is required. This is, of course, due to the well-known fact that the rate of convergence for a Monte Carlo estimate is proportional to the inverse of the square root of the number of simulated sample paths. In other words, reducing the standard error of a Monte Carlo estimate by a factor of two requires the number of sample paths to increase by a factor of four. There are many well-known methods for error reduction, including antithetic simulation, control variate simulation and importance sampling simulation. Recently, Duan and Simonato (1998) proposed a simulation adjustment method which takes advantage of an inherent property of the asset price dynamic. Since the asset price is supposed to be a (semi-) martingale under the risk-neutralized pricing measure, their method sets out to incorporate this theoretical feature into the simulation algorithm. A multiplicative adjustment was used by them to ensure the martingale property in simulation. They demonstrated that the empirical martingale simulation (EMS) substantially accelerates the convergence of Monte Carlo price estimates.

A more drastic departure from the Monte Carlo tradition is the adoption of quasi-Monte

Carlo simulation method for pricing derivatives; for example, Brotherton-Ratcliffe (1994), Paskov and Traub (1995), Joy, Boyle and Tan (1996). The quasi-Monte Carlo method is also known as the low-discrepancy simulation. The idea is to generate a sequence of finite-dimensional, say $k$-dimensional, vectors that fill the $k$-dimensional hypercube more evenly (or having a lower discrepancy) than the $k$-dimensional uniformly distributed independent random vectors. Many sequences are known to possess such a property, and they are commonly referred to as low-discrepancy numbers. If a low-discrepancy sequence is used, in stead of the standard uniformly distributed random numbers, in evaluating a derivative security (technically, computing an integral), the rate of convergence has been shown to be in the order of $(\ln n)^k/n$ where $n$ is the number of sample paths. In comparison to $1/\sqrt{n}$, the rate of convergence for the standard Monte Carlo, it is clear that the low-discrepancy simulation method asymptotically dominates regardless of how large is $k$. In practice, however, the numerical performance of the quasi-Monte Carlo method is greatly affected by the dimensionality of the problem. If $k$ is large but $n$ is not large enough, the quasi-Monte Carlo method may perform poorly.

In this article, we propose a marriage of the EMS and the quasi-Monte Carlo simulation as a way of accelerating Monte Carlo convergence. Using the EMS or quasi-Monte Carlo simulation individually yields a substantial gain in computational efficiency. Combining these two methods together generates a measurable improvement over their individual applications. We carry out a trade-off analysis using a randomly generated test pool of one thousand Asian options. Our results confirm that this new way of approaching derivatives valuation is indeed promising.

# 2 The empirical martingale quasi-Monte Carlo simulation

To facilitate the discussion of our proposed empirical martingale quasi-Monte Carlo (EMQMC) simulation, we provide below a brief review of the EMS and the quasi-Monte Carlo simulation. We first discuss the EMS and then move on to the quasi-Monte Carlo simulation.

The EMS is based on an observation that a general feature shared by all derivatives pricing models is the martingale property. A good simulation method should therefore reproduce such a property for the simulated sample if possible. The martingale property for derivatives pricing essentially states that the discounted asset price must be a martingale under the risk-neutralized pricing system. In other words, the discounted average future asset price under the risk-neutralized probability measure should equal its current value. This statement can be formalized as follows:

$$e^{-rt}E^Q\left[S_t|\mathcal{F}_0\right] = S_0, \tag{1}$$

where $S_0$ denotes the current stock price, $r$ the risk free interest rate, $\mathcal{F}_0$ the information known at time 0 and $E^Q(\cdot|\mathcal{F}_0)$ the conditional expectation operator in a risk-neutralized economy. For typical Monte Carlo simulations, the martingale condition stated in (1) almost certainly fails for any finite simulated sample. For a simulated sample of size $n$, the Monte Carlo estimate for $e^{-rt}E^Q\left[S_t|\mathcal{F}_0\right]$ is

$$\widehat{S}_0 \equiv e^{-rt}\frac{1}{n}\sum_{i=1}^{n}\widehat{S}_{i,t}, \tag{2}$$

where $\widehat{S}_{i,t}$ is the $i$-th simulated value for time $t$. But, $\widehat{S}_0$ will be different from its theoretical value $S_0$ with probability one. The difference is the so-called "simulation error," which decreases if the sample size is increased.

Duan and Simonato (1998) had come up with a multiplicative adjustment to ensure that the martingale property is respected in simulation. For the following discussion, we use a geometric Brownian motion asset price dynamic to describe their empirical martingale adjustment. For a more general exposition, readers are referred to Duan and Simonato (1998). In a standard Monte Carlo simulation, the $i$-th sample path is generated using:

$$\widehat{S}_{i,0} = S_0 \text{ and } \widehat{S}_{i,t} = S_0 \exp\left((r - \frac{\sigma^2}{2})t + \sigma\epsilon_{i,t}\sqrt{t}\right) \tag{3}$$

where $r$ is the risk free rate, $\sigma$ the asset volatility rate, and $\epsilon_{i,t}$ the $i$-th element of a sequence of independent and identically distributed standard normal random numbers. The empirical martingale adjustment modifies the simulated sample paths as follows:

$$S^*_{i,t} = S_0 \exp(rt) \, \frac{\widehat{S}_{i,t}}{\frac{1}{n}\sum_{i=1}^{n} \widehat{S}_{i,t}}, \tag{4}$$

The above adjustment is essentially a multiplication by the ratio of the expected time-$t$ stock price, $S_0 \exp(rt)$, over the average obtained from the simulated sample, $\frac{1}{n}\sum_{i=1}^{n} \widehat{S}_{i,t}$. This ratio is expected to be very close to one if the simulation sample is large. It can be easily verified that the adjusted stock price satisfies $\exp(-rt)\frac{1}{n}\sum_{i=1}^{n} S^*_{i,t} = S_0$. This in turn implies that the simulated sample after the multiplicative adjustment satisfies the martingale property stipulated by the pricing theory.

To understand the quasi-Monte Carlo simulation method, one must first comprehend the notion of discrepancy of a sequence. We will skip a formal definition of discrepancy and opt for a more intuitive understanding instead. A two-dimensional example should suffice for such a purpose. Consider a square defined by $[0,1] \times [0,1]$ and a sequence of two dimensional vectors of numbers with each entry taken from $[0,1]$. In a nutshell, discrepancy measures the maximum absolute difference between the size of the area defined by the subset $[0,x] \times [0,y]$, where $x$ and $y$ are arbitrary but must be confined to $[0,1]$, and the percentage of the points from this sequence falling inside this subset, after reaching, say $n$. If the discrepancy goes to zero when $n$ goes to infinity, the sequence is uniformly distributed. The rate of convergence to zero varies among different uniformly distributed sequences, however. The independent and identically distributed uniform random numbers actually have a discrepancy in the order of $1/\sqrt{n}$. Interestingly, this order of convergence has nothing to do with the dimension of the vectors in the sequence, and the dimension only affects the convergence constant. There are many sequences known to have a smaller discrepancy; for example, Sobol, Faure and Halton sequences. These sequences have a discrepancy in the order of $(\ln n)^k/n$, which depends on $k$, the dimension of the vectors in the sequence. This property suggests that for an extremely large $n$, the low-discrepancy sequence is certain to be more uniformly distributed than the independent, identically distributed uniform random sequence. How large is considered large? The answer depends on the value of $k$, the dimension of the vector. If $k$ is large, then $(\ln n)^k/n$ may still be greater than $1/\sqrt{n}$ for fairly large $n$, which implies that up to this particular sample size, the low-discrepancy sequence is still less uniformly distributed.

The quasi-Monte Carlo method replaces the independent and identically distributed uniform random numbers by a low-discrepancy sequence. The discrepancy of a sequence and the pricing error for derivatives is linked together by the Koksma-Hlawka inequality, which states that the quasi-Monte Carlo pricing error is bounded by a value that is proportional to the discrepancy of the uniformly distributed sequence employed. It is therefore evident that the use of a low-discrepancy sequence will asymptotically dominate the use of the independent and identically distributed uniform random numbers. Indeed, several applications in finance such as Brotherton-Ratcliffe (1994), Paskov and Traub (1995), Joy, Boyle and Tan (1996) have confirmed this theoretical prediction.

The practical relevance of asymptotic dominance is quite a different matter, however. When a problem has a large dimension $k$ and the sample size $n$ is restricted by computing considerations, the relative merit between two types of sequences become murky. Our proposed EMQMC simulation is designed for such a situation.

What is the EMQMC simulation? It is simply the empirical martingale simulation described in (3) and (4) with $\epsilon_{i,t}$ being generated by a low-discrepancy sequence instead of the independent and identically distributed uniform random numbers. Since $\epsilon_{i,t}$ is supposed to be normally distributed, we need to transform the low-discrepancy uniformly distributed sequence into a normally distributed sequence. We use the algorithm proposed by Moro (1995) to perform this transformation.

# 3 A performance analysis

How does the EMQMC perform numerically? The answer to this question lies with a numerical performance analysis. The trade-off relationship between the pricing accuracy and computing time determines the relative performance of the competing methods. The trade-off relationship is established using a test pool of Asian options whose payout is based on the path average of the underlying asset prices over the entire life of the option contract. The choice of the type of options in the test pool is dictated by two factors. First, we want the options to be path dependent. Second, there must be a closed-form solution to facilitate the checking of pricing accuracy. As a result, we choose to use Asian options with the payout based on a discretely sampled geometric average of the prices over the contract life. The valuation framework is the Black and Scholes (1973) geometric Brownian motion setup. This combination allows us to use the analytical formula established by Turnbull and Wakeman (1991) as the benchmark.

The frequency of discrete sampling in the payout calculation determines the dimension of the simulation problem; for example, a daily sampling of prices for an option with a six month to maturity is equivalent to having an approximate dimension of 180. In terms of our notation, $k = 180$. As our earlier discussion revealed, a higher dimension is likely to impede the performance of the quasi-Monte Carlo simulation. We assume for simplicity that the path average is calculated on a weekly sampling basis. A similar analysis can be easily conducted for different frequencies.

Our test pool of options is constructed in a spirit similar to Broadie and Detemple (1996). A test pool of $1,000$ Asian options is created using a random selection of parameters based on pre-determined distributions. The following distributions are used for the parameter values: the volatility rate $\sigma$ is distributed uniformly between 0.1 and 0.6; the maturity $T$ is also distributed uniformly between 0.1 and 1 year; the initial stock price $S_0 = 100$; the exercise price $K$ is uniform between 70 and 130; the risk-free rate of interest $r$ is, with probability 0.8, uniform between 0 and 0.1, and with probability 0.2, equal to 0. Each parameter value is drawn independently of the others.

For each simulation method, we record the total computing time and an aggregate measure of pricing errors for the test pool of options. The computing time is measured by the average time (in seconds) taken to complete the calculations for the test pool using the matrix programming language Matlab and a standard desk-top computer. The aggregate pricing error measure is the root mean squared error:

$$RMS(m,n) = \sqrt{\frac{1}{m}\sum_{i=1}^{m} e_i^2(n)}, \tag{5}$$

where $e_i(n) = |\widehat{C}_i(n) - C_i|/C_i$ with $C_i$ the $i^{th}$analytical price and $\widehat{C}_i(n)$ is the $i^{th}$ estimated option price using one of the Monte Carlo methods with $n$ sample paths. The variable $m$ stands for the size of the test pool, and it is set equal to $1,000$. We eliminate the cases that $C_i < 0.50$ to avoid large relative errors caused by a small divider. We control the test pool to be the same for different Monte Carlo methods using common random numbers. In the Monte Carlo calculation of option prices, different simulation methods also share a common set of random numbers. For each value of $n$, we produce a pair – the computing time and the relative pricing error to study the trade-off.

We examine the time-pricing accuracy trade-off for four different simulation methods. The straightforward Monte Carlo simulation is referred to as the crude Monte Carlo simulation. We also consider the EMS method and the quasi-Monte Carlo simulations with and without the empirical martingale adjustment. The specific low-discrepancy sequence used in the quasi-Monte Carlo simulation is the Sobol sequence. The uniform Sobol numbers are generated using the program FINDER, a software developed at Colombia University.

Figure 1 is generated by varying the value of $n$ from $1,000$ to $20,000$ with an increment of $1,000$. Since the horizontal axis is the time used and the vertical axis the aggregate measure of pricing errors, a better method should lead to a trade-off relationship that is closer to the origin of the graph. As the results indicate, the crude Monte Carlo is easily dominated by either the EMS or the quasi-Monte Carlo method (labeled as Sobol). The EMQMC method (labeled as Sobol-EMS) outperforms all other simulations methods. The efficiency gain by incorporating the EMS adjustment into the quasi-Monte Carlo method is quite pronounced when the time allocated for calculations is small. The gain becomes marginal when more computing time is used to reduce pricing errors. In the above analysis, the time taken to generate the Sobol numbers and transform them into normally distributed ones is not included in the average time spent. Since Sobol numbers are not random, we have first stored them in memory to avoid repeatedly generating them. Excluding their computing time is thus more in line with the actual scenario in applications.

Figure 2 plots the aggregate pricing error as a function of the number of sample paths used. Although a larger number of sample paths certainly increases the computing time for a given simulation method, it need not represent a one-to-one relationship to the computing time when comparing across different simulation methods. This graph depicts the number of sample paths needed to achieve a given level of pricing precision; for example, to achieve a $RMS$ of 0.03, approximately $2,000$ samples paths are needed using the EMQMC simulation method. Using the quasi-Monte Carlo method without the empirical martingale adjustment would require approximately $4,000$ sample paths. The results have a similar meaning as those in Figure 1. For a given level of pricing accuracy, the EMQMC method does best. The empirical martingale adjustment improves the quasi-Monte Carlo method measurably when the sample size is small. In a way, the EMQMC method has the best of two worlds. It has taken advantage of the asymptotic dominance of the quasi-Monte Carlo simulation, and it is at the same time greatly enhanced by the empirical martingale adjustment.

Our results will, of course, be affected by the dimension of the valuation problem. If we change the discrete sampling from weekly to daily, the dimension $k$ increases substantially. According to the order of convergence described earlier, this change is likely to yield an outcome that is even more in favor of making the EMS adjustment to the quasi-Monte Carlo method.

# 4 Conclusion

We have proposed a straightforward marriage of two established simulation enhancement methods. The quasi-Monte Carlo simulation is known to deliver an excellent performance when the sample size is large, but the method's performance can be questionable when the path dependency (dimension) is high and the sample size is not very large. Since the sample size is not expected to be very large for many day-to-day applications, the quasi-Monte Carlo method is thus not all that desirable from a practical standpoint. We have shown in this article that the quasi-Monte Carlo simulation method can be greatly enhanced by incorporating the empirical martingale adjustment. This combination results in a simple simulation method that effectively turbo-charges Monte Carlo simulations.

## References


[1] Black, F. and M. Scholes, 1973, The Pricing of Options and Corporate Liabilities, *Journal of Political Economy* 81, 637-654.

[2] Broadie, M. and J. Detemple, 1996, American Option Valuation: New Bounds, Approximations, and a Comparison of Existing Methods, *Review of Financial Studies* 9, 1211-1250.

[3] Brotherton-Ratcliffe, R., 1994, Monte Carlo Motoring, *Risk* (December), 53-57.

[4] Paskov, S. and J. Traub, 1995, Faster Valuation of Financial Derivatives, *Journal of Portfolio Management*, 113-120.

[5] Duan, J.-C. and J.-G. Simonato, 1998, Empirical Martingale Simulation for Asset Prices, *Management Science* 44, 1218-1233.

[6] Joy, C., P. Boyle and K.S. Tan, 1996, Quasi Monte Carlo Methods in Numerical Finance, *Management Science* 42, 926-936.

[7] Moro, B., 1995, The Full Monte, *Risk*, February, 57-58.

[8] Turnbull, S. and L. Wakeman, 1991, A Quick Algorithm for Pricing European Average Options, *Journal of Financial and Quantitative Analysis* 26, 377-389.

Figure 1: Computational efficiency of Monte Carlo methods

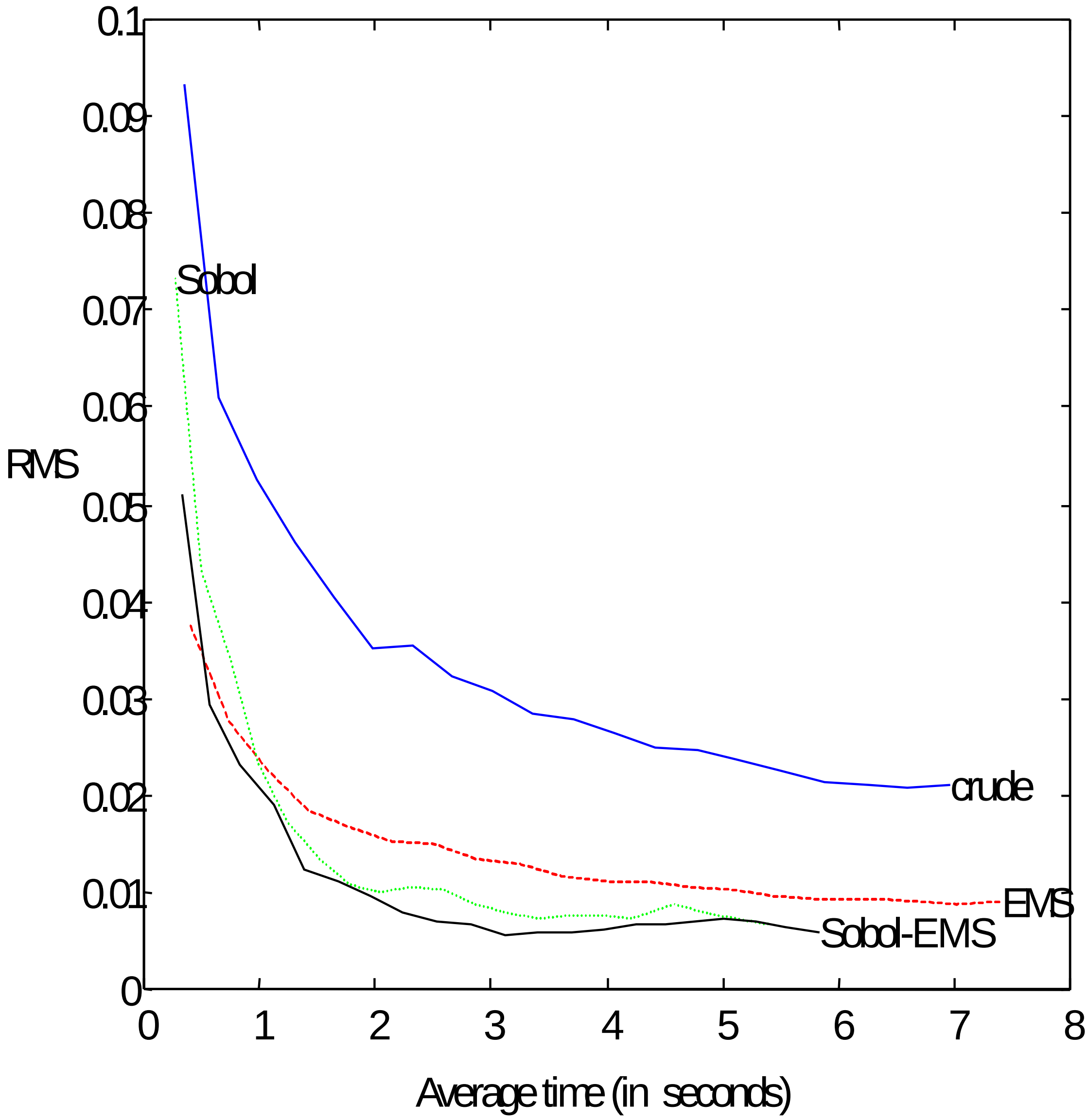

Figure 2: Trade-off between pricing accuracy and sample size