Asian Option Pricing with Monte Carlo

Anne-Sophie Karmel

May 30, 2017

Introduction

The purpose of this report is to present the results of Asian option pricing based on Monte Carlo simulations. The robustness and accuracy of the pricing method is discussed for different pay-off functions (fixed and average) and both types of non-weighted averages (arithmetic and geometric).

An Asian option is an exotic derivative (strong path dependence) even though it is considered as a vanilla product on crude oil trading. The pay-off function is dependent of the average price of the underlying during the option life (part of only for forward starting option):

- $C(T) = max(S(T) - A(0,T), 0)$ and $P(T) = max(A(0,T) - S(T), 0)$ for a floated strike
- $C(T) = max(A(T,0) - E, 0)$ and $P(T) = max(E - A(T,0), 0)$ for a fixed strike

They are expected to be cheaper than European options as they are by design less sensitive to the volatility of the underlying. While this is particularly true for continuous average, in practice all computational methods are based on discretization of time and therefore an Asian option may have an important sensibility to the volatility (vega). Continuous time is simulated through small time steps (one trading day for the present implementation). Still, this approximation triggers a certain type of model error (as detailed later in the report). The average functions are described below:

- Continuous algorithmic average

  $$A(T,0) = \frac{1}{T}\int_0^T S(t)dt$$

  and for discrete time

  $$A(T,0) = \frac{1}{N}\sum_{i=1}^{N} S(t_i)$$

- Continuous geometric average

  $$A(T,0) = \frac{1}{T}\int_0^T \ln(S(t))dt$$

  and for discrete time

  $$A(T,0) = \exp\left(\frac{1}{N}\sum_{i=1}^{N} \ln(S(t_i))\right)$$

Monte Carlo methods are well-designed for multi-dimensional pay-off functions as they simulate multiple pay-off and compute the discounted value of the mean. However, they require intense processing time and struggle to address early exercise and other path-dependent events for which finite difference models (e.g. Crank Nicholson methods) seem to be more effective.

For this report, experimentations are conducted on a call option with an underlying that follows a Geometric Brownian Motion. Both fixed and floated strikes will be simulated for arithmetic average. However, geometric average call will be priced by an exact formula as a closed form solution exists (detailed later).

Results and observations are presented for two model error types:

- number of simulations for sampling the expected value of pay-off,
- number of time steps for simulating the underlying path prices.

Observed impacts on variation of model parameters are also mentioned nonetheless further explorations on other parameters are not covered in this report.

Numerical procedures

The price of the option is defined as :

$$V(t) = \mathbb{E}^{\mathbb{Q}}\left[e^{-\int_t^T r_\tau d\tau}\mathbf{Payoff}(S_T)\right]$$

with the $\mathbf{Payoff}(S_T)$ function as defined previously, i.e. for a call option $C(T,A) = max(S(T) - A(0,T), 0)$ (floated strike) and $C(T,A) = max(A(T,0) - E, 0)$ (fixed strike).
The underlying asset price S is assumed to follow a Geometric Brownian Motion, i.e. : $dS = \mu S dt + \sigma S dW(t)$ where μ is the asset price drift, σ the asset price volatility and $W(t)$ represents a random variable that follows a Brownian motion.
Consequently, the option price is obtained by derivating Black-Scholes formulae (for example through a replication portfolio that is dynamically hedged):

$$\Pi = V(S(t), I(t), t) - \Delta S(t)$$

where $I(t) = \int_0^t f(S,\tau)d\tau$ is the newly introduced path-dependent variable, and $dI(t) = f(S,t)dt$
From Itô :

$$dV = \frac{\partial V}{\partial t}dt + \frac{\partial V}{\partial S}dS + \frac{\partial V}{\partial I}dI + \frac{1}{2}\frac{\partial^2 V}{\partial S^2}dS^2$$

$$dV = \left(\frac{\partial V}{\partial t} + \mu S\frac{\partial V}{\partial S} + \frac{1}{2}\sigma^2 S^2\frac{\partial^2 V}{\partial S^2}\right)dt + \frac{\partial V}{\partial I}dI + \sigma S\frac{\partial V}{\partial S}dW$$

Few more derivation steps and assumptions on dynamic hedging are required to obtain the following equation [Wilmott 2006]:

$$dV = \frac{\partial V}{\partial t} + f(S,t)\frac{\partial V}{\partial I} + \frac{1}{2}\frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0$$

Geometric average and closed form solution

For the geometric average Asian option price, the BS equation becomes :

$$dV = \frac{\partial V}{\partial t} + log(S)\frac{\partial V}{\partial I} + \frac{1}{2}\frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0$$

The geometric average is lognormal distributed therefore a closed form solution of Black -Scholes equation for geometric average Asian option exists and a call price is defined as [Wilmott 2006]:

$$C_{float}(S,A,t) = SN(d_1) - A^{t/T} S^{(T-t)/T} e^{\frac{-\left(r+\frac{\sigma^2}{2}\right)(T^2-t^2)}{2T} + \frac{\sigma^2}{6}\frac{T^3-t^3}{T^2}} N(d_2)$$

with

$$d_1 = \frac{tln\frac{S}{A} + (r+\frac{\sigma^2}{2})\frac{T^2-t^2}{2}}{\sigma\sqrt{\frac{T^3-t^3}{3}}}$$

and

$$d_2 = \frac{tln\frac{S}{A} - (r+\frac{\sigma^2}{2})\frac{T^2-t^2}{2}}{\sigma\sqrt{\frac{T^3-t^3}{3}}} = d_1 - \sigma\sqrt{\frac{T^3-t^3}{3}}$$

Therefore, Asian geometric call option price is computed as an exact price and the following numerical method is used for pricing arithmetic Asian call option only.

The general approach is as follow:

- Generate N underlying price paths
- Compute pay-off function for each of the paths
- Calculate the expected value of the pay-off and discount it at time t_0
- Estimate the mean error and evaluate the 95% confidence interval

Generation of underlying prices

First, we need to generate random variable from the standardized Normal distribution. Then we need to update the asset price at each time step using these random increments.
With the Euler Maruyama method, each price increment is calculated as follow :

$$\delta S = rS\delta t + \sigma S\sqrt{\delta t}\phi$$

where ϕ is drawn from the standardized normal distribution. The discretization method has an error of $O(\delta \mathrm{t})$
A more precise approach is to add the Itô second derivative term (i.e. Milstein correction) :

$$S_{t+\delta t} = S_t e^{(r-\frac{1}{2}\sigma^2)\delta t + \sigma\phi\sqrt{\delta}t}$$

Both approaches are implemented in C++. The standard normal random variable is generated with a Box-Miller algorithm.

Discounted Expected Pay-off computation

Then the discounted expected value of the N pay-off is calculated as follow:

- For the floating strike :

$$C_{float,simulationn} = \mathbb{E}(e^{-rT}C(T)) = e^{-rT}\mathbb{E}((S(T) - A(0,T))^{+})$$

- For the fixed strike :

$$_{float,simulationn} = \mathbb{E}(e^{-rT}C(T)) = e^{-rT}\mathbb{E}((A(T) - K)^{+})$$

This computation is to be processed for each Monte Carlo simulation, i.e. up to 10^6 times for the current analysis.

Model errors

Two kind of errors that may question the convergence or the accuracy of the model are evaluated:

- the size of of the time step when simulating asset prices
- the number of Monte Carlo simulations to assess expected discounted pay-off mean

Results and observations

The initial parameter values are :

$$T = 1\text{year}; \sigma = 0.2; S_0 = 100; Strike = 100; r = \ 0.05$$

Geometric average Asian option

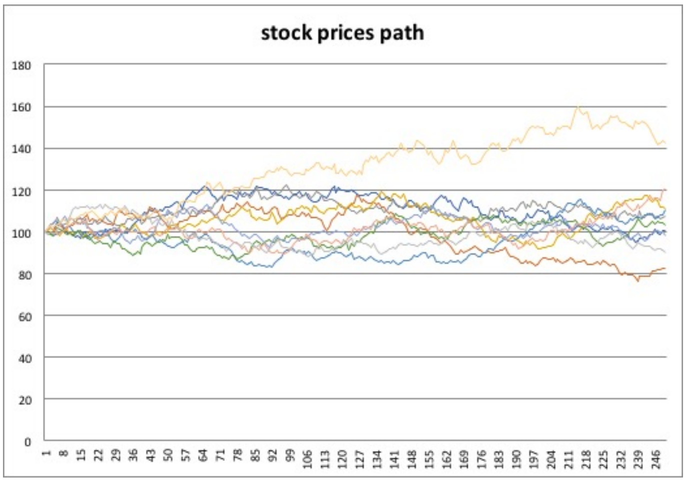
First the result of the geometric Asian call closed form solution are presented below:

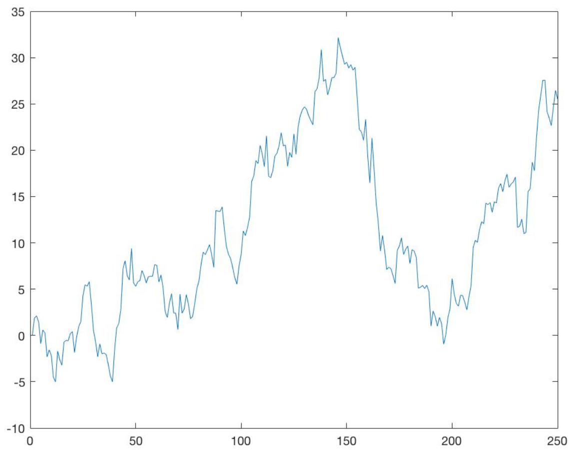
Geometric Fixed Asian Call	Geometric Floated Asian Call
5.51753	4.31556

Underlying prices

By simulating 10 times the stock price paths with daily time steps (250) using the Milstein correction, we obtain the following graph.

The absolute difference of asset prices between Euler-Maruyama method and the Milstein correction are presented below.
As expected, the observed differences are random as they represent the inclusion of the second order derivation of Itô lemma that is simulated with $\sqrt{t}\phi$.





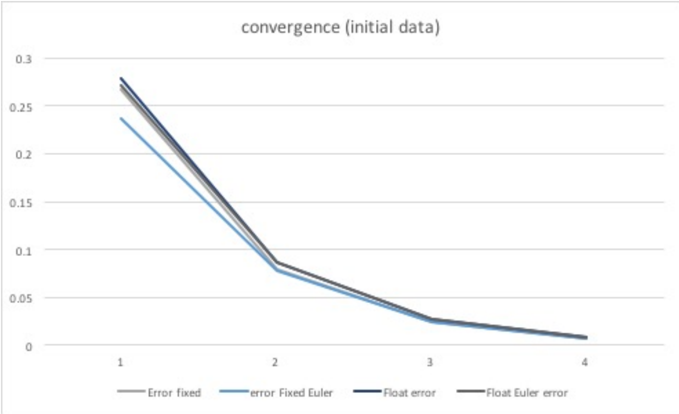
Arithmetic average Asian option

Then, the results of the arithmetic Asian call prices for 10 to 10^6 Monte Carlo simulations are presented below:

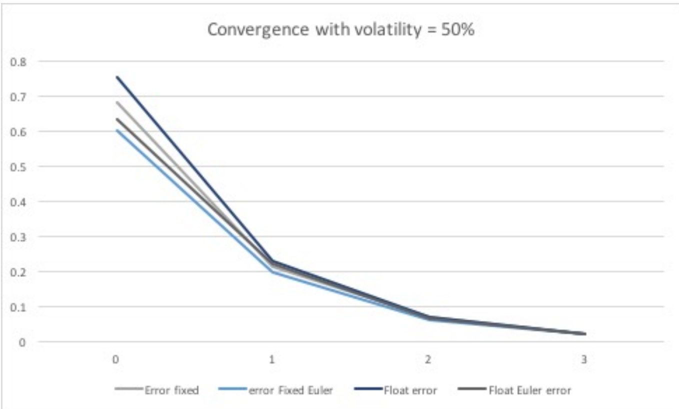
nb simulations	Fixed Call	confidence interval.95	Floated Call	confidence interval.95
10	6.91239	[1.4483; 12.3765]	7.27563	[1.30653; 13.2447]
100	6.72322	[4.99643; 8.45001]	7.08764	[5.31163; 8.86365]
1,000	5.76665	[5.27143; 6.26187]	6.2392	[5.67397; 6.80443]
10,000	5.48882	[5.33703; 5.6406]	6.14972	[5.97798; 6.32146]
100,000	5.5195	[5.47113; 5.56787]	6.16768	[6.11367; 6.22169]
1,000,000	5.49484	[5.47955; 5.51013]	6.10766	[6.09073; 6.12458]

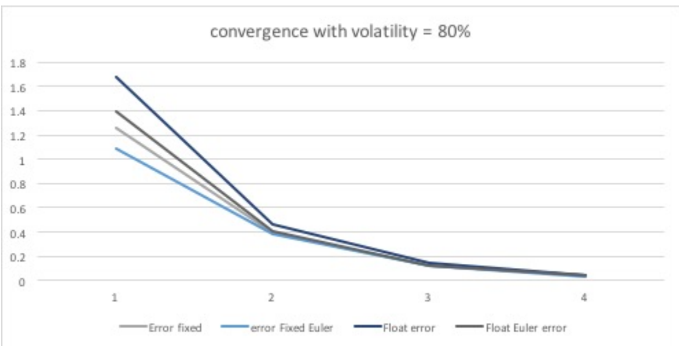
Model errors

The convergence of the model errors for the initial parameters is presented below :

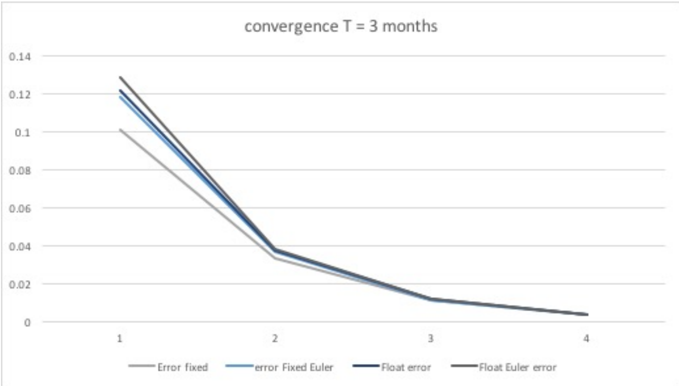


We observe a slower convergence when the volatility of the underlying asset increase :

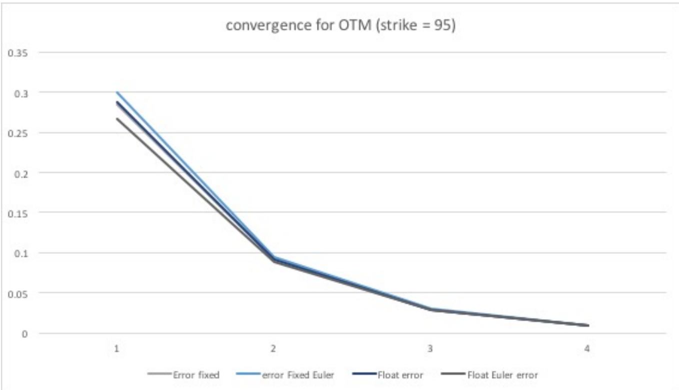




However, when decreasing the expiry to 3 months, the convergence is not affected:



Lastly, for an out-of-the-money option, the strike being set at 95, the convergence results also doesn't seem to be affected :



These last results tend to emphasize certain limits of the Monte Carlo simulation approach.
Firstly on the maturity side: a 3-month call and a 1-year call would not be traded with the same liquidity and their prices are likely to be affected by this missing parameter in Monte Carlo and Black-Scholes methods. Moreover, the longer the expiry the more the assumption of a constant risk-free interest rate becomes irrelevant. Moreover, one can wonder how the real drift μ impacts the price in the long term.
Secondly, on the volatility side : if the underlying asset price is highly volatile, we may expect σ to follow a random walk (maybe with some jumps) instead of being constant. Then some behavioral finance elements may interfer and impact the option price (e.g. anchoring, when the agents expect the underlying price to come back to an initial level regardless its actual volatility)[Wilmott 2017].

C++ implementation

The numerical methods are implemented in C++ in order to cope with the intense processing time required for Monte Carlo simulations. Some design pat-

terns have been used although much more can be done to improve the reusability, flexibility (e.g. introduction of new stochastic variables or change of pay-off function) and processing time of the program [Joshi 2004].
To speed the randow walk generation, other variance reduction approaches can be implemented too, e.g. control variates (the error is deemed similar to the error observed for a Vanilla option where both Monte Carlo and closed form solution can be implemented) or antithetic variates (the number of generated random variable is doubled by simply taking the opposite value). Using multiple approaches and boundary controls is also recommended to check the consistency of the implementation.

Conclusion

Numerical methods for exotic option pricing can be implemented through multiples ways, Monte Carlo simulations, finite differences and various approximation of closed form solutions.
Monte Carlo simulations require intensive processing time for robust results therefore, closed form solutions (or even approximation of) must be preferred when possible.
Still, all these pricing models - including Black-Scholes closed form solutions - are limites by the strong and irrealistic assumptions they carry. Non-model-capturated parameters like liquidity, agent psychologic biais, commercial relationship, target deadline, etc. would need to be considered when assessing the price of an exotic derivatives.

References

"C++ Design Patterns and Derivatives Pricing", 2nd edition, 2004, Mark S. Joshi, Cambridge University Press.
"Paul Wilmott on Quantitative Finance", 2000, Paul Wilmott, Wiley.
"The Money Formula", 2017, Paul Wilmott and David Orrell, Wiley.