Stochastic Methods for Finance

Report 6: Montecarlo and Euler-Maruyama Simulation

Lorenzo Magnino

May 12, 2023

Abstract:

In this report, we want to analyse and compare methods for the simulation of financial objects. In particular, we will use the Monte Carlo and Euler-Maruyama methods to simulate the trajectories of a geometric Brownian motion, the price of a European and Asian call and put option. Finally, we will give a generalisation for the path-dependent payoff price by studying the case of a look-back option.

Introduction:

A geometric Brownian motion (GBM) process is a stochastic process that is widely used to model market price movements and other natural phenomena. Its mathematical form is given by the stochastic differential equation:

$$dS_t = \mu S_t dt + \sigma S_t dW_t$$

where S(t) represents the price of the asset at time t, μ represents the drift rate (expected growth rate of the asset), σ represents the volatility of the asset and dW(t) represents a random increment of a Wiener process (also called Brownian motion) with zero mean and unit variance.
Given the initial condition and under the risk-neutral measure we know that the solution takes the following form:

$$S_t = S_0 \exp\left[\left(r - \frac{\sigma^2}{2}\right)t + \sigma W_t\right],$$

where r is the risk-free interest rate.
The simulation of a GBM requires the numerical solution of this SDE. Two commonly used methods for simulating SDEs are the Monte Carlo method and the Euler-Maruyama scheme.

The Monte Carlo method for simulating a GBM requires the generation of random numbers to sample the solution of the stochastic differential equation. Specifically, the simulation of the Wiener process (Brownian motion) is performed by generating random numbers from a standard normal distribution and integrating the stochastic differential equation.

The Euler-Maruyama scheme is an iterative method that divides the process into time intervals and estimates the solution in each interval. Specifically, the Euler-Maruyama scheme for simulating a GBM requires dividing time into small intervals and using iterative formulas to calculate the asset increment in each interval.

In finance, the simulation of a GBM is used to model the price movements of financial instruments, e.g. shares, bonds, currencies and commodities. A GBM simulation can also be used to evaluate the price of a financial option, such as a call or put option. For example, the Monte Carlo method can be used to simulate the price of the underlying asset and calculate the option price, while the Euler-Maruyama scheme can be used to assess the risk of an investment portfolio. In addition, the simulation of a GBM can be used to build forecasting models and analyse financial market dynamics.

Results obtained with Excel's VBA:

- **Parameters Settings:**

We simulate an underlying asset for the development of our models and analyses. These are the parameters:

Parameters setting:	
S_0 (Initial stock price)	110
σ (Volatility)	0,2
r (Risk-free interest rate)	0,01
T (Time to maturity)	1
K (Exercise price)	104,5
dt (discretisation of time: 1 day)	0,003968253968

Where dt represents the discretisation of 1 year divided into 252 steps (i.e. how many working days per year).

- **Simulation of N=100 trajectories of GBM:**

We used the Euler-Maruyama method to simulate these trajectories. It is based on the fact that Brownian motion is a Gaussian process and the geometric Brownian motion is a log-normal process. We have:

1. $W_0 = 0$
2. For any $t \geq 0, \delta \in (0, +t)$, $W_{t+\delta} - W_t \sim N(0,\delta)$
3. Forward increments are independent

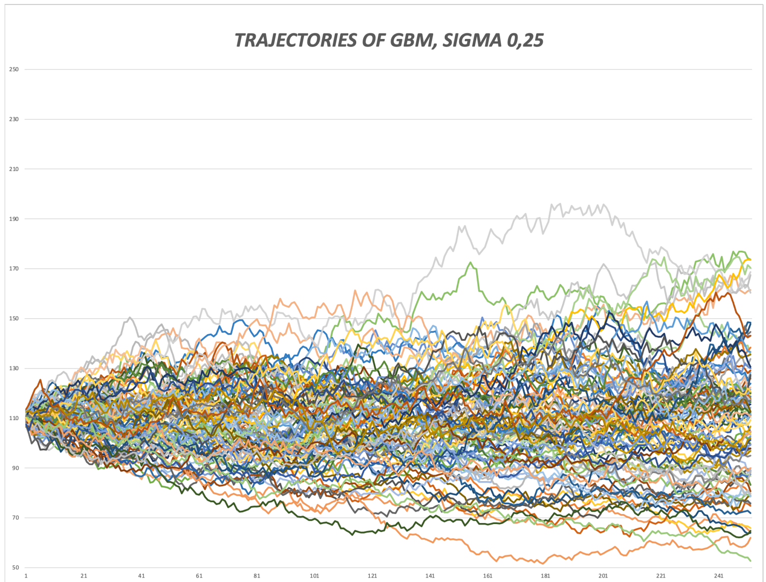
So on the interval $[0,T]$ we create a partition $0 = t_0 \leq t_1 \ldots \leq t_n = T$ so that using property 2. we can write:

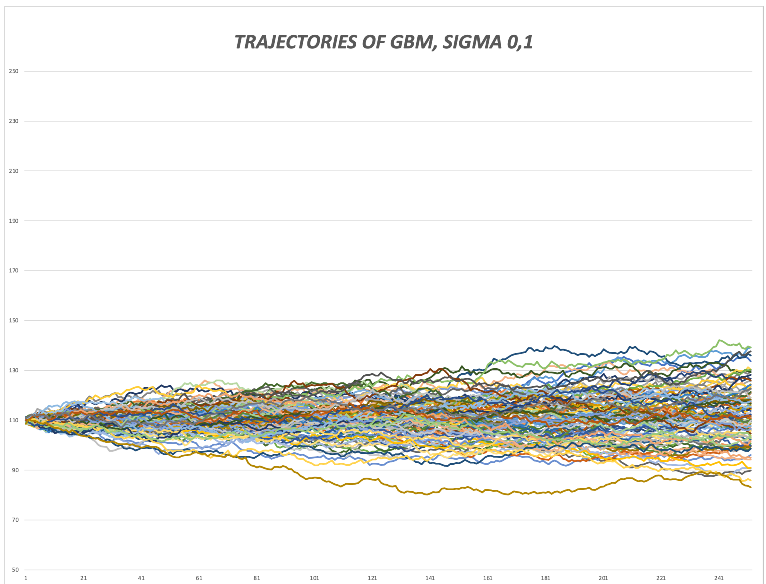
$$W_T = W_{t_n} = \sqrt{t_n - t_{n-1}}N_{n-1} + \ldots + \sqrt{t_1 - t_0}N_0$$

where $(Nj)_j \sim N(0,1)$:
In our VBA code, we have *WorksheetFunction.NormInv(Rnd(),0,1) to give values at Gaussian.*
In particular, in terms of stock price, we have this kind of simulation of each trajectory:

$$S_{t+1} = S_t + S_t(r\Delta t + \sigma\sqrt{\Delta t}N(0,1))$$

We have reported two simulations of 100 trajectories, which differ only in the volatility value for sigma = 0.25 and sigma = 0.1 respectively.





- **Pricer of vanillas (Call/Put) through 1-step simulation and through multi-step Euler-schemed (N=1200).**

The objective is to price a European call and put option using the Monte Carlo method and the Euler-Maruyama method and compare the two methods.
In particular, what we are going to do is to simulate the following value numerically:

$$price_t = e^{-rT} E[payoff_T]$$

To do this we should generate N (=1200) values of the price S_T using the above methods and then calculate the following payoffs for each S_T generated (approximating the trajectory of GBM doing only one jump from 0 up to maturity T):

$$Payoff_T^{call} = (S_T - K)^+ \qquad\qquad Payoff_T^{Put} = (K - S_T)^+$$

Finally, we should average these payoffs to calculate the associated candidates.

Montecarlo:

CallPrice (Black -Scholes)	12,1662309
Montecarlo simulation	12,4864058
Absolute Error	0,3201749
PutPrice (Black - Scholes)	5,62643857
Montecarlo simulation	5,18316228
Absolute Error	0,44327629

Euler-Maruyama:

CallPrice (Black -Scholes)	12,1662309
Euler-Maruyama	11,70193004
Absolute Error	0,46430086
PutPrice (Black - Scholes)	5,62643857
Euler-Maruyama	5,07028001
Absolute Error	0,55615856

- **Pricer of Asian (Call/Put) through multi-step Euler-schemed:**

An Asian option is a type of financial option that has a unique feature compared to traditional options. Unlike European or American options, where the strike price is determined on the expiry date, in an Asian option the strike price depends on the average price of the underlying asset during a period of time.
In particular, the strike price of an Asian option is determined as the average (mean integral) of the price of the underlying asset during the observation period. This observation period is usually a continuous time interval that can be one day, one week, one month or any other time interval specified at the time the option is issued. The main objective of an Asian option is to reduce the volatility risk of the underlying asset during the observation period. In other words, the Asian option provides greater stability in the strike price than traditional options, which can have a highly variable strike price during the observation period.

In particular, the payoff of an Asian Call and an Asian Put becomes:

$$Payoff_T^{As-call} = \left(\frac{1}{T} \int_0^T S_t dt - K \right)^+ \qquad Payoff_T^{As-Put} = \left(K - \frac{1}{T} \int_0^T S_t dt \right)^+$$

We want to use the same steps as in the previous step for the Euler-Maruyama method. However, we must deal with the presence of the integral on all trajectories, up to maturity, of S_t. To do this, through numerical approximation, we exploit the discretisation of the integral as a sum and store the generated trajectories to make their arithmetic mean.
Finally, we repeat the same steps as before, deducing:

CALL PRICE	PUT PRICE
8,806482128	2,349433332

- **Generalized Payoff: Path-dependent Option - LookBack Option**

A lookback option is a type of financial option that gives the owner the right, but not the obligation, to buy or sell an underlying asset at a specified price, which is determined according to the highest or lowest price of the underlying asset during the option's term.

One of the main reasons for using lookback options is their ability to limit investment risk. Since the strike price is based on the highest or lowest price of the underlying asset, the option owner can obtain a more favourable strike price than with traditional options, which reduces the risk of losing the investment. Moreover, lookback options are particularly useful in highly volatile markets, as they allow the option owner to take advantage of fluctuations in the price of the underlying asset.

$$Payoff_T^{LB-call} = S_T - min\{S_t, t \leq T\} \qquad Payoff_T^{LB-Put} = max\{S_t, t \leq T\} - S_T$$

Here, too, we perform the steps previously done using the discretised model of S_t:

$$S_{t+1} = S_t + S_t(r\Delta t + \sigma\sqrt{\Delta t}N(0,1))$$

Different to before, we would have to calculate the payoffs in the manner set out above and then average them out. Finally, this value will be multiplied by e^{-rT}*. We Found:*

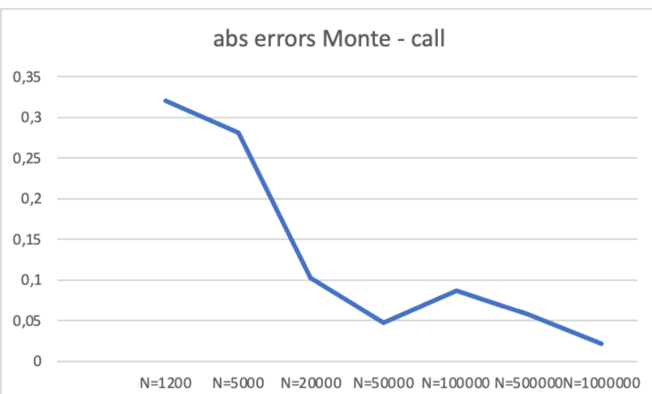
CALL Price	PUT Price
16,93240081	17,09149528

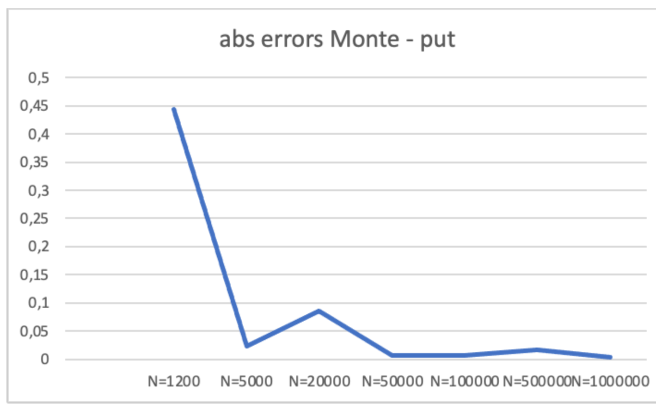
Conclusions:

It is evident from the first graphs that as volatility increases, the noise within the geometric Brownian motion trajectories increases. In fact, for a volatility of 10% the trajectories are contained in the interval [85,145], while increasing the volatility by 15% (thus reaching 25%) the trajectories oscillate much more, being contained in a much larger interval, [50,230].

Concerning the pricing of a European option by simulation both the 1-step Monte Carlo method and the multi-step Euler method are commonly used approaches for pricing European options.
Both methods have advantages and disadvantages, and the choice depends on the specific needs of the application. The 1-step Monte Carlo method is often simpler to implement and requires less calculation time than the multi-step Euler method. However, Euler's multi-step method may be more accurate in some situations, such as when the underlying asset has high volatility or when the option has a long-term maturity.
In general, the best method depends on the complexity of the option, the characteristics of the underlying asset and the time and resources available for the simulation.
Here we have seen that the Montecarlo Simulation is more suitable for our model since it is more precise and requires less complexity to be implemented and computed.

Finally to better understand the speed of convergence of Euler's method, we have simulated several trials by varying the number of simulations used (clearly, the errors are those of the specific trials. in fact, they change each time a trial is re-run)





We can see how the velocity of convergence for these particular simulations is very different between a call and a put.