Monte Carlo Simulation

Daniel Gutierrez Velez

Summary

This text covers the Monte Carlo Simulation approach to price options. We start by introducing a crucial component for the Monte Carlo simulations for option pricing: the Geometric Brownian motion, which is used in almost all implementations. Then the basic one-step Monte Carlo approach for European option pricing is described. From that we build up to the Euler-schema which simulates whole paths of prices. From that we then steer to Asian options which take the average of the path rather than the final price. Another variation is evaluated, lookback, where we take the min or max of the whole series as input for the payoff of the option. Finally, we implement a “worst of” 3 assets method and compare them all and the classic Black-Scholes Implementation.

Introduction

The Monte Carlo simulation involves repeatedly sampling from a probability distribution to observe the behavior of a specific phenomenon across many scenarios. This technique is extensively utilized in finance and insurance. In finance, it helps model uncertain components of projected cash flows, providing a range of Net Present Values (NPVs) to assess the financial volatility of companies. In insurance, it is used to estimate potential losses from events like earthquakes to appropriately price policies. A particularly significant application discussed here is in derivative pricing, specifically for options. This approach uses the risk-neutral valuation method, where the price of an option is determined based on its expected value under the assumption that the underlying asset (a non-dividend-paying stock) follows a Geometric Brownian Motion trajectory. The focus is on generating price paths for the underlying asset using this model. Then evaluating the generated price paths compared to the strike value and give a expected price for the option.

Geometric Brownian Motion

The Geometric Brownian Motion **(GBM)** is a type of stochastic process where the log of a randomly changing variable follows a Brownian motion (Wiener process) with a consistent drift. This concept is critical in stochastic differential equations, often utilized in financial mathematics to model fluctuating stock prices, notably within the Black-Scholes framework.

A stochastic process of interest for us is the path that follows a stock S_t, which follows a GBM if the following stochastic differential equation is satisfied:

$$dS_t = \mu S_t dt + \sigma S_t dW_t \quad (1)$$

Where W_t is a Winner process, μ is the percentage drift and σ is the percentage volatility of the process.

This is based on a continuous domain, for practical application the focus is changed for discrete domain.
The discretized version of Geometric Brownian Motion (GBM) simplifies the continuous model into steps that can be calculated at discrete intervals, making it practical for numerical simulations like Monte Carlo methods. In this version, stock prices at each time step are estimated using the previous step's price, adjusted by a drift term proportional to the time step, and a random shock derived from the normal distribution scaled by volatility and the square root of the time step. Here is the formula used for coding a VBA function:

$$S_{t+\Delta t} = S_t \exp\{\left(\mu - \frac{1}{2}\sigma^2\right)\Delta t + \sigma\sqrt{\Delta t}\,\epsilon\} \quad (2)$$

Where:

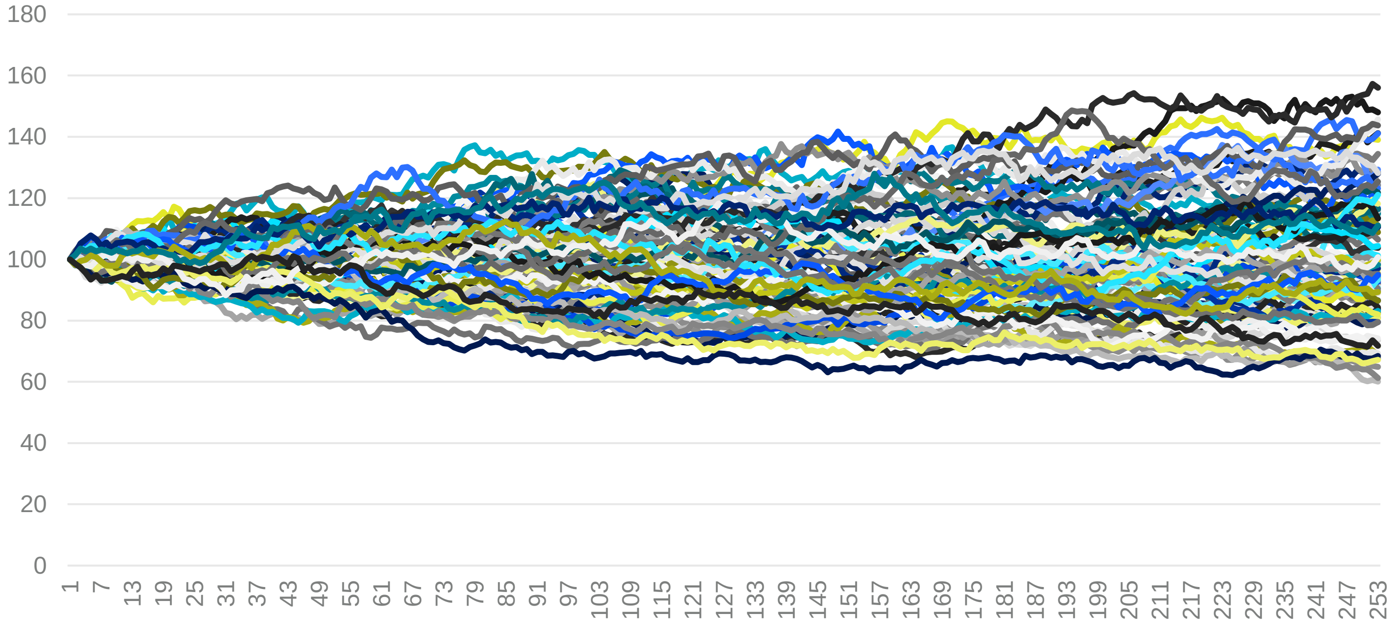
- Δ_t is the time increment.
- μ is the drift coefficient.
- σ is the volatility shock.
- ϵ is a standard normal random variable generated using the Box-Muller method.

The results for 100 GBM are presented below.

Monte Carlo Simulation

Daniel Gutierrez Velez

GBM Trajectories



GBM Simulation details

For the 100 trajectories above, the following inputs were given:

- Initial price S0 = 100
- μ drift coefficient = 1%
- σ volatility shock = 20%
- Time = 1 year
- Δ_t time step = 1/252 days in a trading year

Monte Carlo Inputs

For the Monte Carlo pricing approaches these are the inputs given:

- S_0 as initial price of the underlying.
- ***K*** as strike price.
- S_T as price at maturity of the underlying.
- ***T*** as time to maturity.
- ***r*** as risk-free rate.
- σ as the volatility of the underlying.
- Δ_t as time steps in ***T.***
- ***N*** as number of random draws.

Vanilla Monte Carlo Pricing

A VBA code was implemented to build a function to price either a call or a put option using the Monte Carlo simulation method. It simulates different scenarios of stock price movements at the option's expiration, based on Geometric Brownian Motion. The function takes as parameters, the initial stock price, strike price, time to expiration, risk-free rate, volatility, and the number of simulations. It calculates the stock price at maturity for each simulation, determines the payoff for each scenario (depending on whether it is a call or put), and then calculates the average payoff across all simulations. This average payoff is discounted to present value to determine the option's price.

The price is then given by:

$$Option\ Price = e^{-rT}\frac{1}{N}\sum_{i=1}^{N}(Payoff_i) \quad (3)$$

Where the payoff of a call is $(S_T - K)^+$ and the payoff of the put is $(K - S_T)^+$

Monte Carlo Simulation

Daniel Gutierrez Velez

Euler Scheme

The Euler-Maruyama method to price European call or put options via Monte Carlo simulation. This method updates the price path by taking small steps from an initial stock price to an expiration, incorporating randomness and drift.

The stock price update is calculated with equation (1). And the price of the option is given again by:

$$Option\ Price = e^{-rT}\frac{1}{N}\sum_{i=1}^{N}(Payoff_i) \quad (3)$$

Where the payoff of a call is $(S_T - K)^+$ and the payoff of the put is $(K - S_T)^+$

Asian Options

For the Asian options, a VBA code was implemented using the Monte Carlo simulation approach, where the option payoff is based on the average price of the underlying asset over a specified period, rather than its price at maturity alone.

The Asian options have analytical solution if we take the continuous time geometric average of the path, but for the sake of this written, the arithmetic average of the path was taken.

We simulate again the stock price at each time step by updating its price using equation **(1).**

The average stock price over the simulation period is then calculated. The option's payoff for each simulation depends on the type of option:

$$Option\ Price = e^{-rT}\frac{1}{N}\sum_{i=1}^{N}(Payoff_i) \quad (3)$$

Where the payoffs are given by:

- Call payoff = $\left([\frac{1}{N}\sum_{i=1}^{N}(S_t)] - K\right)^+$ (4)
- Put payoff$\left([K - \frac{1}{N}\sum_{i=1}^{N}(S_t)]\right)^+$ (5)

Lookback Options

For the lookback a code was designed to price lookback options using Monte Carlo simulation. Lookback options are exotic options where the payoff depends on the maximum or minimum price of the underlying asset during the option's life.

The function simulates multiple paths of an asset's price using Geometric Brownian Motion (GBM), calculating the maximum and minimum prices reached across the duration for each simulation. Depending on whether it's pricing a call or a put, the payoff is determined by the difference between these extremums and the strike price. The final price of the option is the average of these discounted payoffs across all simulations.

The underlying follows the price path given by equation **(1).**

While the price of the option varies slightly from the other previous implementations, specifically in the payoff of the call or put option.

$$Option\ Price = e^{-rT}\frac{1}{N}\sum_{i=1}^{N}(Payoff_i) \quad (3)$$

Where the payoffs are given by:

- Call payoff = $(\max(S_t) - K)^+, \forall t \in [0, T]$ (6)
- Put payoff = $(K - \min(S_t)^+, \forall t \in [0, T]$ (7)

Worst of Certificate

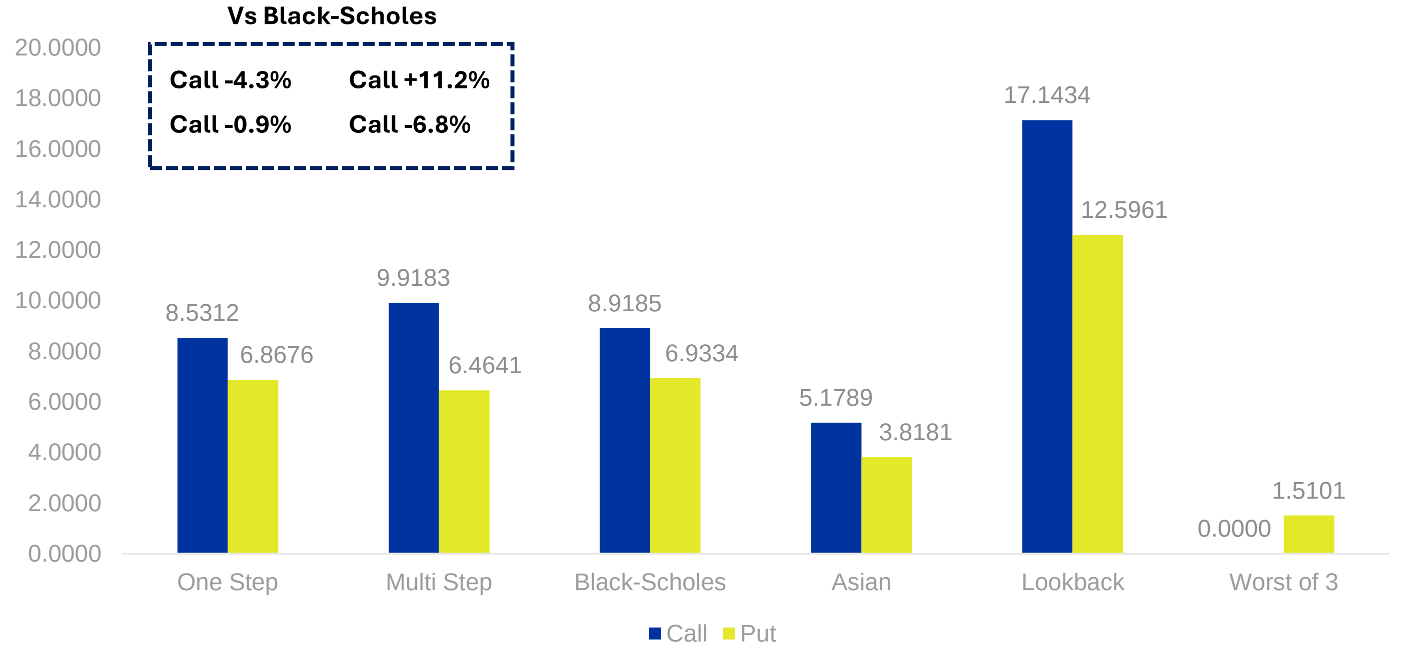
For the final VBA function, a function for a "Worst Of" was coded for prices options on a basket of three assets, calculating the option payoff based on the worst-performing asset. This approach uses Monte Carlo simulation to model the path of each asset using Geometric Brownian Motion (GBM).
The price paths of the 3 assets are given by equitation (1), while the price is given by equation (3).
Where the payoffs are given by:

- Call payoff = $\left(\min(S_{i,T}) - K\right)^+, \forall i \in [1,2,3]$ (8)
- Put payoff = $\left(K - \min(S_{i,T}\right)^+, \forall i \in [1,2,3]$ (9)

Option Greeks

Daniel Gutierrez Velez

Results



Methods Input

For all approaches the following inputs were given:

Input	Value
S0	100
K (strike)	99
T (years)	1
r (risk free rate)	1.0%
Vol σ	20%
Steps Δ_t	252
N iters	1,000

Keeping in mind that each method constructs a GBM set, where the drift is obtained using the volatility of the underlying, the time to maturity and the risk-free rate.

The drift is given by:

$$\mu = \left(r - \frac{1}{2}\sigma^2\right)T$$

Conclusions

We see that the one-step and multi-step (Euler scheme) implementations both yield different prices compared to the Black-Scholes model. This discrepancy can be minimized by increasing the number of iterations, leveraging the law of large numbers. Asian options are the cheapest because their prices are based on averages, which tend to be less volatile than just the value of the underlying at expiry. Lookback options are the most expensive, as they allow the buyer to select the optimal historical price, either the min or max, likely better than the price at expiry. Finally, the 'worst of' option is the least expensive among those discussed, because it is based on the worst-performing asset in a basket of three, leading to a lower likelihood of high payoffs in Monte Carlo simulations. Consequently, the discounted price of these averaged simulations results in a lower price.

Daniel Gutierrez
Msc. Data Science Student
University of Padova
daniel.gutierrezvelez@studenti.unipd.it