Imperial College London

MATH70113 - Simulation Methods for Finance

Imperial College London

Department of Mathematics

Coursework n°1

Authors:
Matthieu Baboulène (02299325)
Antoine Balouka (02305271)
Max Dedieu (02289032)
Rayan Messekher (02299112)
Quentin Varrier (02280080)

Lecturer:
Harry Zheng

May 9, 2023

Contents

1 Introduction

Financial modeling and simulation are increasingly important in finance to predict market trends and price financial instruments. However, traditional methods may not accurately reflect the behavior of financial instruments in complex and dynamic markets. This has made advanced modeling and simulation techniques necessary. The Black-Scholes model, which assumes constant volatility, has been widely used in options pricing since 1973. However, this assumption may not perfectly reflect the behavior of financial instruments in the market. Stochastic volatlity models, which allow for volatility to vary over time, have emerged as a good alternative. This coursework aims to explore the pricing of European, geometric Asian, and arithmetic Asian options under the Black-Scholes model and then the assumption of stochastic volatility with the Heston model (1993). We will also use simulation methods to estimate the sensitivity of option prices like the quotient method, the likelihood ratio method, and the pathwise method.

2 Base Model and Task

2.1 The Black-Scholes-Merton Model

In this first part we focus on the most known model : the Black-Scholes-Merton model. Under the risk neutral mesure $\mathbb{Q}$ we modelize the evolution of a stock S_t using the following stochastic differential equation (SDE)

$$dS_t = rS_t dt + \sigma S_t dW_t \tag{1}$$

This SDE is a Geometric Brownian Motion where W_t is a brownian motion under Q and r the risk free interest rate. Using Ito's formula applied on $x \to ln(x)$ we obtain 2

$$S_t = S_0 \exp\left(\left(r - \frac{\sigma^2}{2}\right)t + \sigma\sqrt{t}Z\right) \quad Z \rightsquigarrow N(0,1) \tag{2}$$

Using the Risk Neutral Pricing formula 3 we can price a call option with maturity T and strike price K at time t :

$$C_{BS}(t,T,K) := E^Q\left[e^{-r(T-t)}\left(S_T - K\right)^t\right] \tag{3}$$

Using equations 2 and 3 a basic calcul of expectation shows that the price of the call is determined by equation 4 :

$$C(S_t,t) := \Phi(d_1)S_t - \Phi(d_2)Ke^{-r(T-t)} \qquad \text{Where } \Phi \text{ is the C.D.F of } \mathcal{N}(0,1) \tag{4}$$

Where d_1 and d_2 are defined as follows :

$$d_1 := \frac{1}{\sigma\sqrt{T-t}}\left[\ln\left(\frac{S_t}{K}\right) + \left(r + \frac{\sigma^2}{2}\right)(T-t)\right] \qquad d_2 := d_1 - \sigma\sqrt{T-t} \tag{5}$$

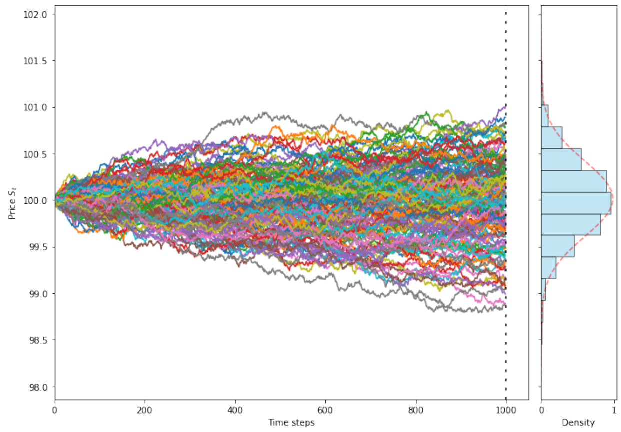
2.2 Code Structure

2.2.1 Price trajectories generation

We will mostly rely on the Monte Carlo method to carry out this coursework. We thus need to generate price trajectories taking into account different market parameters, such as the price at time $t = 0$ S_0, the risk-free interest rate r, the volatility σ, and the time horizon T.
Instead of working in continuous time, we discretize the time horizon T in N time-steps, such that we have the infinitesimal time $dt = \frac{T}{N}$. For each time increment, we compute the asset price increment dS_t, which will be given by the market model (here either Black-Scholes-Merton or Heston).

We can visualize the paths generated by our class to better understand how market parameters affect the price trajectories. We can also use a simpler model to simulate the asset price, for instance Bachelier's model. We use a 0 drift model, equating to a normal behavior for the asset's price. We use parameters $\sigma = 0.4, S_0 = 100, T = 1$, $r = 0.05$ for Black-Scholes-Merton, and $r = 0.00$ for Bachelier's simulation.

(a) Price trajectories with parameters $\sigma = 0.4$, $r = 0.05$, $S_0 = 100$, $T = 1$ For B.S.M's model



(b) Price trajectories with parameters $r = 0$ for Bachelier's model

2.2.2 Derivative Payoff

Now that we can successfully generate price trajectory, another important component of the Monte Carlo method is to compute the payoff of the derivative at the final time $t = T$. To do so, we implement the class PAYOFF, described in more precision in the appendix. To ensure the class is working as expected, we simulate the payoff for both type of European options, the call with payoff $(S_T - K)^+$, and the put with payoff $(K - S_T)^+$. We choose the strike price $K = 100$.

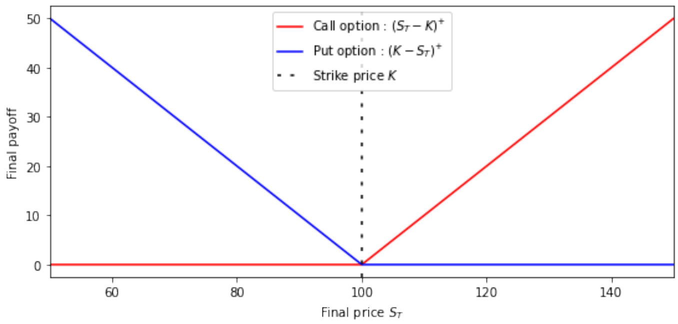


Figure 1: Simulated Payoff for the European call and put with strike price $K = 100$

2.2.3 Monte Carlo Pricer

We have now the ability to generate price trajectories, and compute the payoff of a given derivative. The last step is to wrap these two processes together so we can use Monte Carlo's method. To recall the principle of Monte Carlo's method :

- Select a payoff function F that returns the payoff of the derivative at time T to a price trajectory : $F(S_{t\in\{0..T\},i}) = P_i$
- Create a PAYOFF object using the payoff function F
- Simulate M price trajectories with given market and model parameters, using the PATHGENERATOR class

- Compute the discounted mean (by a factor of e^{-rT}) of the simulated payoffs computed on all the price trajectories : $e^{-rT}\sum_{i=1}^{M} P_i$, where r is the risk-free interest rate

- Assign the price at time $t = 0$ of the derivative to the computed mean

The MonteCarloPricer class automates the process described above. We can dive deeper in the structure of the code with this extended UML diagram in Annex : 6.

2.3 Results

We now have the essential building blocks to work on our first task. We wish to price a European call option using Monte Carlo's method, and estimate the relative greeks. We also want to study how the number of steps and number of paths influence the simulation's precision and time of execution.
We will study the behavior of a call option with strike price $K = 100$ in the Black-Scholes-Merton model with market parameters $S_0 = 100, r = 0.05, \sigma = 0.4, T = 1$.

2.3.1 Call Price

As described above, the call price is obtained by averaging the discounted payoff of many simulations. To have a reference price to measure accuracy, we compute the closed-form solution to this problem, using 4. We find $C = 18.023$. Since the European call options are path-independent, meaning they are not influenced by the values taken by the stock for $t \in [0, T - dt]$, we can compute the simulation for only 1 time step. This mean the steps simulated are in $\{0, T\}$. This is computationally efficient since we do not generate non-relevant data, and this helps us save memory so we can generate more paths. We have below the results of the simulation for an array of price trajectories.
We note that we can compute the error with a 95% confidence interval for the Monte Carlo estimation, using below formulas :

$$CI = [\bar{X} - 1.96\frac{\sigma}{\sqrt{N}}, \bar{X} + 1.96\frac{\sigma}{\sqrt{N}}] \tag{6}$$

$$\bar{X} = \frac{1}{N}\sum_{i=0}^{N} X_i \qquad \hat{\sigma} = \frac{1}{N-1}\sum_{i=1}^{N}(X_i - \bar{X})^2 \tag{7}$$

Number of paths	$\hat{C}$	Execution time (s)	95% Confidence error	Relative error
100	19.35	0.001	± 10.3	-10.2%
1000	17.60	0.009	± 3.47	4.4%
10000	18.34	0.048	± 1.03	-2.3%
100000	18.09	0.6	± 0.34	0.72%
1000000	18.08	2.75	± 0.11	0.54%

Table 1: Monte Carlo simulation results (1 step) for the Black-Scholes-Merton model for a call option with strike price $K = 100$ and market parameters $r = 0.05, \sigma = 0.4, T = 1, S_0 = 100$p

As we can see, the estimated call price $\hat{C}$ is getting more precise with the number of paths increasing (the relative error is decreasing). This comes at a cost of the execution time, which increases more or less linearly.

To get a better idea of how the values evolve with the increase of number of paths, we can plot the estimated price call $\hat{C}$, the confidence interval and the time of execution. We have below results in 2.

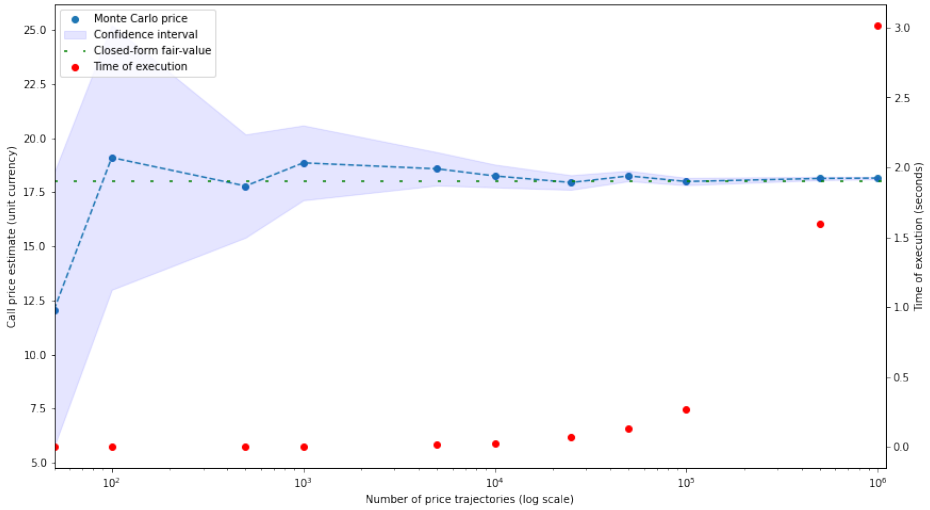


Figure 2: Call price estimation and time of execution of the Monte Carlo methods with 1 step and variable number of price trajectories using Black-Scholes-Merton model with $r = 0.05, \sigma = 0.4, T = 1, K = 100, S_0 = 100$

We see what appears to be an exponential increase in time of execution, since the x-scale is logarithmic, we deduce the increase is indeed linear.
We have a decent result from 10^5 price trajectories, that yield the results in a less than a second. This is the best compromise between time and accuracy for these settings.

2.3.2 The greeks : closed-form, finite difference, pathwise and likelihood method

We now wish to estimate sensitivy of the call price to some of its parameters. The partial derivative of the call price to its input parameters (market settings) are called greeks, in this simulation we focus on the 3 main greeks :

$$\nu = \frac{\partial C}{\partial \sigma}(S_0, \sigma, r, T) \quad \Delta = \frac{\partial C}{\partial S_0}(S_0, \sigma, r, T) \quad \gamma = \frac{\partial^2 C}{\partial S_0^2}(S_0, \sigma, r, T) \tag{8}$$

Under Black-Scholes-Merton model assumptions, we know that the call price can be expressed as in 4. We can derive the closed-form expression of the greeks using this formula, this yields :

$$\nu = \Phi(d_1)\sqrt{T}S_0 \quad \Delta = \Phi(d_1) \quad \gamma = \Phi(d_1)\frac{1}{S_0\sigma T} \tag{9}$$

We have several methods to estimate the greeks in this setting. We can use the finite difference method (also called quotient method), the pathwise method, and the likelihood method [2].
Starting with the finite difference method, we note θ a parameter (volatility σ^2, starting stock price S_0), we compute the expected payoff w.r.t a random variable Y : $\mathbf{E}[Y(\theta)]$. We compute M realizations of $Y(\theta)$, such that we have $\bar{Y}(\theta) = \sum_{i=1}^{M} Y_i(\theta)$, that is an unbiased estimator of $\mathbf{E}[Y(\theta)]$ according to the Strong Law of Large Number (S.L.L.N).
Considering a small increment h of the parameter θ, we have :

$$\Delta_C = \frac{\bar{Y}(\theta + h) - \bar{Y}(\theta - h)}{2h} \tag{10}$$

This is the central difference estimator for $\frac{d}{d\theta}\mathbf{E}[Y(\theta)]$. This estimator is biased, such that $Bias(\Delta_C) \simeq o(h^2)$, and has variance $Var(\Delta_C) = o(h^{-2})$. This means that a small value for h will reduce the bias, but will also increase the variance. We cannot choose h arbitrarily small, we must pick a good compromise.
The finite difference estimator can be used to approximate all derivatives, under assumptions of smoothness of the function.

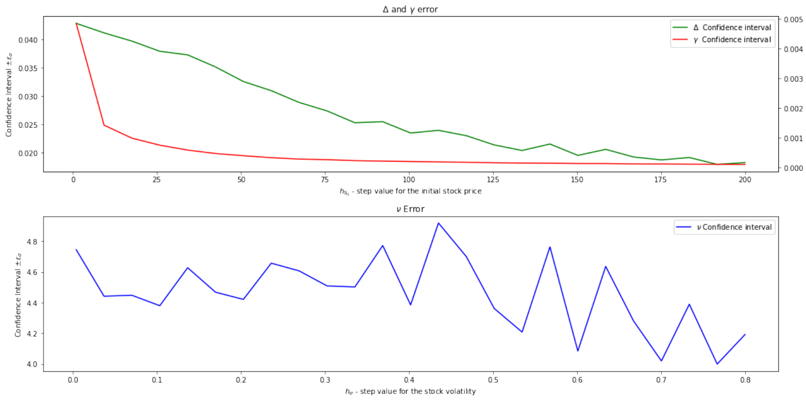


Figure 3: Impact of the step-size h_σ and h_{S_0} in the half-width of the confidence interval of the Monte-Carlo simulation (Black-Scholes-Merton model with $r = 0.05, \sigma = 0.4, T = 1, K = 100, S_0 = 100$)

As we can see above, the choice of the step-size parameter h is very important. In practice, we try to find a step-size parameter that reduces the width of the confidence interval, for a fixed number of price trajectories and simulation steps, and fixed market parameters. As we can see, for Δ and γ we observe a clear convergence (γ is more visible since h is squared in the formula), but it is less obvious for ν.

The pathwise method uses a different method. We have put emphasis in the mathematical proof in the appendix. We use the chain-rule to compute the partial derivative, and then use Monte Carlo method to get an estimate. For instance, let $Y = e^{-rT}(S_T - K)^+$, where $S_T = S_0 e^{(r-\frac{\sigma^2}{2})T - \sigma\sqrt{T}Z}$ and $Z \sim \mathcal{N}(0,1)$. We note $C = \mathbf{E}[\mathbf{Y}]$, by the chain-rule, we have :

$$\frac{dY}{dS_0} = \frac{dY}{dS_T}\frac{dS_T}{dS_0} \quad \text{and} \quad \mathbf{E}[\frac{dC}{dS_0}] = \mathbf{E}[\frac{dY}{dS_0}]$$

By the S.L.L.N, we can find an estimator $\hat{\Delta}$ of Δ, by generating M price trajectories with final value $S_{T,i}$. We then have : $\hat{\Delta} = \frac{1}{M}\sum_{i=1}^{M} e^{-rT}\mathbf{1}_{\{S_{T,i}-K\geq 0\}}\frac{S_{T,i}}{S_0}$

With the same notation as before, we derive the unbiased estimator for ν, noted

$$\hat{\nu} = \frac{1}{M}\sum_{i=1}^{M} e^{-rT}\mathbf{1}_{\{S_{T,i}-K\geq 0\}}(log(\frac{S_{T,i}}{S_0}) - (r + \frac{\sigma^2}{2})T)\frac{1}{\sigma}$$

However, we cannot derive the pathwise estimator for γ, since the payoff function is not smooth enough.

The likelihood estimator is derived from the MLE for the partial derivative of C. We note $g(x) = \frac{d}{dx}P(S_T \leq x)$ the probability density function of S_T. In the Black-Scholes-Merton model, S_T follows a log-normal law. In the Black-Scholes-Merton model, this is equivalent to :

$$g(x) = \frac{1}{x\sigma\sqrt{T}}\phi(\xi(x)) \quad \xi(x) = \frac{log(x/S_0) - (r - \sigma^2/2)T}{S_0\sigma^2 T}$$

We can thus derive the expression for Δ, γ, ν as expectation of a random variable (respective derivative of the price of the option) :

$$\frac{\partial C}{\partial S_0} = \mathbf{E}[e^{-rT}(S_T > K)^+\frac{Z}{S_0\sigma\sqrt{T}}] \qquad \frac{\partial^2 C}{\partial {S_0}^2} = \mathbf{E}[e^{-rT}(S_T > K)^+(\frac{Z^2-1}{S_0^2\sigma^2 T} - \frac{Z}{S_0^2\sigma\sqrt{T}})]$$

$$\frac{\partial C}{\partial \sigma} = \mathbf{E}[e^{-rT}(S_T > K)^+(\frac{Z^2-1}{\sigma} - \sqrt{T}Z)]$$

To estimate these values, we use the Monte Carlo method. This yields an estimator that converges with a $\sqrt{N}$ speed, where N is the number of price trajectories.

We compile the results of all these simulation in below table 1 :

	Closed-Form	Finite difference		Pathwise		Likelihood	
		Value	Error	Value	Error	Value	Error
Delta	0.627	0.632	± 0.01	0.635	± 0.001	0.626	± 0.003
Gamma	0.009	0.005	± 0.0001	N/A	N/A	0.009	± 0.0015
Vega	37.842	37.859	± 0.189	37.392	± 0.179	37.431	± 0.593

Table 2: Values and errors obtained for the greeks using the different simulation methods with $r = 0.05, \sigma = 0.4, T = 1, K = 100, S_0 = 100$

As we can see, there is no "best" method to estimate the greeks. Some methods are closer to the closed form solution, but with a confidence interval that is larger. The Pathwise method produce convincing results, but cannot estimate the value of γ.

3 Geometric Asian options and Stochastic Volatility

3.1 The asian options in the Black-Scholes-Merton model

Unlike European and American options, the Asian options that we will study here depend on the price of the underlying S at time T, but also on all the prices between 0 and T. To modelize an Asian option, we need first to discretize the interval $[0, T]$. Then we compute the (geometric or arithmetic) mean of the S_{t_i} for $t_i = hi$ and $h = T/n$.

A geometric asian call will have the following payoff : $A_T = ((\prod_{i=1}^n S_{t_i})^{1/n} - K)^+$. The payoff of an arithmetic asian call will be : $A_T = (\frac{1}{n}\sum_{i=1}^n S_{t_i} - K)^+$.
In practice, it is a computationnal challenge to compute $\bar{S}^{\Pi} = (\prod_{i=1}^n S_{t_i})^{1/n}$ because of float overlow. Indeed, say we have a price trajectorie spanning in the order of $1,000$ for $100,000$ time steps, the value $\bar{S}^{\Pi}$ is of the order $(1e^8)^{1/5}$, which would be converted to a NP.INF in python. We use the *log* transformation so the numbers we deal with are smaller : $\bar{S}^{\Pi} = exp(\frac{1}{n}\sum_{i=1}^n log(S_{t_i}))$.

We create a new PAYOFF object that simulate the payoff of the geometric and arithmetic Asian options for a price trajectory.

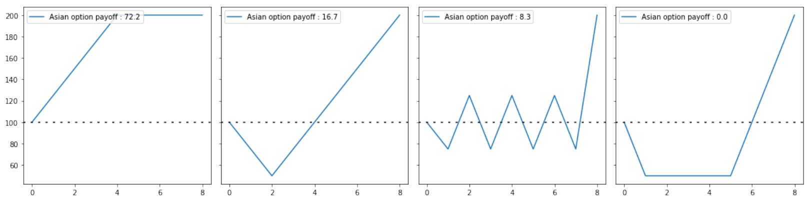


Figure 4: Payoff of the arithmetic asian option for different price trajectories in the Black Scholes Merton model with parameters $r = 0.05, \sigma = 0.4, T = 1, K = 100, S_0 = 100$

As we can see, the payoff is path dependant, meaning the option's payoff can be different even if the final price S_T is the same.

3.1.1 Pricing Asian options

To compute the fair value of the asian options, we can use the previously used Monte Carlo method. We us the Black-Scholes-Merton model : $dS_t = rS_t dt + \sigma S_t dW_t, \quad 0 \le t \le T$

To price the arithmetic and the geometric Asian option, we use the standard class MONTECARLOPRICER defined above. We now have to compute the path of the stock before pricing it, so the number of steps in simulation could have an impact of the precision of the results. To see things more clearly, we study below figure 5.

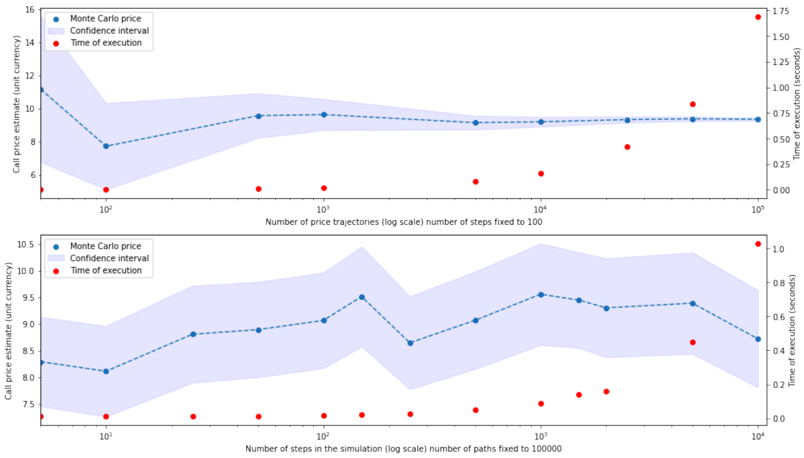


Figure 5: Convergence of the price of the geometric Asian option for fixed number of path and variable number of steps and vice-versa (Black-Scholes-Merton with parameters : $r = 0.05, \sigma = 0.4, T = 1, K = 100, S_0 = 100$)

As we can see, the number of steps influences very little the convergences of the final price, for a fixed number of price trajectories. The number of paths influences much more the shrinking of the 95%-confidence interval, by the strong law of large number. Including more steps would only lengthen the computation time, which is less than ideal.
We fix the number of steps to 100, which for $T = 1$ "year" would be equivalent to simulating around two prices every week. We wish to price the arithmetic and geometric call options with a growing number of paths.We can observe the price yielded by the Monte Carlo method in below table 3.

Number of paths	Arithmetic Option			Geometric Option		
	Price	Error	Exec Time	Price	Error	Exec Time
100	10.75	4.05	0.012	12.0	3.0	0.02
1,000	10.23	1.02	0.096	9.75	0.973	0.034
10,000	10.0	0.32	0.1	9.39	0.297	0.365
100,000	10.1	0.1	3.0	9.34	0.09	4.27
1,000,000	10.1	0.03	16.5	9.35	0.030	18.3

Table 3: Price, 95%-confidence interval and execution time for the Arithmetic and Geometric Asian option for the Black-Scholes-Merton $r = 0.05, \sigma = 0.4, T = 1, K = 100, S_0 = 100$

We see above that the greater the number of paths, the more precise the solution, but the longer the execution time. At around 1,000,000 paths, we see the time of executions starts to be unacceptable in a "real-time" setting.
Furthermore, the arithmetic option is systematically priced higher than the geometric option. This is a result of this theorem :

$$(\prod_{i=1}^{N} X_i)^{\frac{1}{N}} \leq \frac{1}{N}\sum_{i=1}^{N} X_i \qquad \text{, since } (a+b)^2 \geq 0 \Rightarrow \frac{a+b}{2} \geq \sqrt{ab}$$

As an arithmetic Asian call A is not log-normal, performing some operations can be difficult. For example, an arithmetic Asian option doesn't have a closed-form pricing formula. To solve this problem, we can use Lévy's approximation. The key idea behind the Levy approximation is to find a lognormal distribution that has the same first two moments as the underlying Levy distribution.

Once this is done, the option price can be approximated by the option price under the lognormal distribution.

To do so, we create a new random variable Z such that $Z = e^{a+bX}$ with $a, b \in \mathbb{R}$ and $X \sim \mathcal{N}(0,1)$.
Then, to compute the values of a and b we solve the following system in the appendices :

$$\begin{cases} \mathbb{E}[A] = \mathbb{E}[Z] & \iff & m_1 = e^a e^{b^2/2} \\ \mathbb{E}[A^2] = \mathbb{E}[Z^2] & \iff & m_2 = e^{2a+2b^2} \end{cases}$$

Solving for a and b gives:

$$a = \log\left(\frac{m_1^2}{\sqrt{m_2}}\right) \quad \text{and} \quad b = \sqrt{\log\left(\frac{m_2}{m_1^2}\right)} \text{ (more details in part B of the appendices)}$$

Once we have determined a and b, we can use the lognormal distribution $Z = e^\wedge\{a + bX\}$ to approximate the distribution of A_T. We can then use the standard Black-Scholes formula to price the European fixed strike Asian call option, using Z instead of A_T.

Finally, using the Monte-Carlo Method on several realisations of a standard normal random variable and the values of a and b, we can approximate the fair price value of an arithmetic asian call. We can also derive the following closed-form formula [1] :

$$C[S(t), A(t), t] = e^{-r(T-t)} \{m_1 N(d_1) - [K - A(t)(m+1)/(N+1)]N(d_2)\}$$

where

$$d_1 = \frac{\frac{1}{2}\ln m_2 - \ln[K - A(t)(m+1)/(N+1)]}{b} \quad \text{and} \quad d_2 = d_1 - b$$

The values obtained can be seen in table 4.

Methods	Results	
	Value	Error
Monte Carlo on Arithmetic Average	10.282	0.010
Monte-Carlo on Lévy Approximation	10.296	0.313
Closed-form formula	10.266	N/A

Table 4: Fair-price obtained using Lévy's approximation for an arithmetic Asian option with market parameters $r = 0.05, \sigma = 0.4, T = 1, K = 100, S_0 = 100$

3.1.2 Estimating the greeks

Geometric Average Asian Options

The methods used for computing the greeks will be the quotient method and the pathwise method. However, we cannot use the pathwise method for the gamma as there is not enough smoothness of the integrand in the expectation calculus.

We use the same steps as in the previous section to compute the greeks with the quotient method.
For the pathwise method, the estimators are the following :

$$\begin{cases} \hat{\Delta} = \frac{d(\bar{S_T}-k)^+}{dS_0} \approx \mathbb{E}\left[Y'(S_0)\right] = e^{-rT}\mathbb{E}\left[\frac{\mathbb{1}_{\bar{S_T}-K>0}(\bar{S_T}-K)^+}{S_0}\right] \\ \hat{\nu} = \frac{d(\bar{S_T}-k)^+}{d\sigma} \approx \mathbb{E}\left[Y'(\sigma)\right] = e^{-rT}\mathbb{E}\left[\frac{1}{n}\prod_{i=1}^n(S_T)\sum_{i=1}^n(W_{t_i} - \sigma t_i)\prod_{i=1}^n(S_{t_i}^{1/n-1})\right] \end{cases}$$

Arithmetic Average Asian Options

Again we cannot estimate the pathwise estimator for the gamma greek.
The estimators for the pathwise method are the following :

$$\hat{\Delta} = e^{-rt}\mathbb{E}\left[\mathbf{1}_{\{\bar{S}\geq K\}}\frac{\bar{S}}{S_0}\right], \qquad \hat{\nu} = e^{-rt}\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}(S_T\frac{(\log(S_t/S_0)-(r+\sigma^2)t)}{\sigma}\right]$$

with n the number of steps in each path.

The values obtained for both types of options can be seen in table 5.

Arithmetic	Quotient		Pathwise	
	Value	Error	Value	Error
Delta	0.556	0.010	0.561	0.012
Gamma	0.017	0.0005	-	-
Vega	21.180	0.901	21.793	0.924

Geometric	Quotient		Pathwise	
	Value	Error	Value	Error
Delta	0.536	0.010	0.645	0.012
Gamma	0.016	0.0005	-	-
Vega	18.148	0.792	17.822	0.852

Table 5: Greeks obtained for the Asian Options

3.2 Stochastic volatility : the Heston model

One of the limitations of the Black-Scholes model is that volatility is assumed to be constant. This is why other, more complex models have been developed. The Heston model (1993) that we will use in this part is one of them, and assumes that volatility is stochastic. In the Heston model, the price of an asset S is determined by the following stochastic differential equations:

$$dS_t = rS_tdt + S_t\sqrt{v_t}dW_t, \text{ and } dv_t = \kappa(\theta - v_t)dt + \tilde{\sigma}\sqrt{v_t}dB_t$$

with B and W Brownian motions with correlation ρ and r, κ, θ and $\tilde{\sigma}$ positive constants.

To generate price trajectories, we still use class PATHGENERATOR but with a new price move function based on the Heston model which returns for given price S and volatility v at t and increment dt the price and the volatility at $t + dt$. This PRICE_MOVE_HESTON function uses the previous equations to compute the new stochastic volatility, and then the new price with two correlated normal random variables to modelize the two Brownian Motions.

3.2.1 Implied parameters for Heston model

One major problem of Heston model is the important number of parameters, which have no intuitive values. To solve this problem, we used market data to compute implied parameters. We collected the prices of 73 call options on AAPL and fit the parameters using the market price of these options. Then, we solved the optimization problem described by equation 11 by minimizing the loss function between the market price and Heston model price.

$$\sum_{i=1}^{n}(C_{Heston,i}(\kappa,\theta,\sigma_0,\theta,\rho) - C_{market,i})^2 \tag{11}$$

We chose AAPL because of its high correlation with SPY, hoping to extend the implied parameters for other assets. The solution is noisy due to the significant number of parameters, and the result depends on the initial values used in the gradient descent. To solve this problem, we used "classical" Heston parameters as initial values. The implied parameters we get (table 6) are good enough to provide an idea of the parameters that can be used in the Heston model, and the generated paths confirm that these values make sense.

Parameters	σ_0	κ	$\tilde{\sigma}$	θ	ρ
Implied Value	0.09990	2.09570	0.00230	0.09770	-0.09965

Table 6: Implied parameters of Heston model for AAPL call options

We wish to study how the Heston model differs from the Black-Scholes model when pricing the Asian options, with similar market parameters ($r = 0.05, \sigma_0 = 0.4, T = 1, K = 100, S_0 = 100, \kappa = 1.5, \theta = 0.5, \hat{\sigma} = 0.3, \rho = 0.1$). We see the results in table 7 for $n_{steps} = 10^2$ and $n_{paths} = 10^6$.

Option	Arithmetic (Heston)	Geometric (Heston)	Arithmetic (BS)	Geometric (BS)
Fair-value	12.930 ± 0.044	11.315 ± 0.039	10.135 ± 0.032	9.321 ± 0.030

Table 7: Prices of the geometric and arithmetic Asian options under BSM and Heston model

We can notice that the fair prices obtained with the Heston model are higher than the ones obtained with the Black-Scholes model. This is due to the fact that the Heston model slightly surestimates the volatility.

3.2.2 Estimating the greeks

In order to implement Heston model we have chosen the following values for the parameters. Subsequently, we discuss about finding implied parameters using the market price of call options (for more detail see 3.3.2). The objective is to compute the greeks for both Arithmetic and Geometric Asian options using different methods.

Parameters	r	σ_0	κ	$\tilde{\sigma}$	θ	ρ
Value	0.05	0.4	1.5	0.3	0.5	0.1

Table 8: Values of Heston parameters

Arithmetic Average Asian Options

The steps to compute the greeks with the quotient method are the same as before. The estimator for the delta with the pathwise method is as follow :

$$\frac{d(\bar{S_T} - k)^+}{dS_0} \approx \mathbb{E}\left[Y'(S_0)\right] = e^{-rT}\mathbb{E}\left[\frac{\mathbb{1}_{\bar{S_T}-K>0}(\bar{S_T} - K)^+}{S_0}\right], \; \bar{S_T} = \frac{1}{n}\sum_{i=1}^{n} S_{t_i}$$

The result obtained can be seen in table 9. When available we display the estimator error for the different method.

Geometric Average Asian Options

The steps to compute the greeks with the quotient method are the same as before. The estimator for the delta with the pathwise method is as follow :

$$\frac{d(\bar{S_T} - k)^+}{dS_0} \approx \mathbb{E}\left[Y'(S_0)\right] = e^{-rT}\mathbb{E}\left[\frac{\mathbb{1}_{\bar{S_T}-K>0}(\bar{S_T} - K)^+}{S_0}\right], \; \bar{S_T} = \prod_{i=1}^{n}(S_{t_i})^{1/n}$$

The result obtained can be seen in table 9. When available we display the estimator error for the different method.

Geometric	Quotient		Pathwise	
	Value	Error	Value	Error
Delta	0.571	0.004	0.603	0.013
Gamma	0.012	0.00014	-	-
Vega	12.888	0.211	-	-

Arithmetic	Quotient		Pathwise	
	Value	Error	Value	Error
Delta	0.532	0.004	0.623	0.012
Gamma	0.012	0.00013	-	-
Vega	10.734	0.178	-	-

Table 9: Results of computing the greeks of the Asian Options

Appendices

A The Greeks

The pathwise method uses a different method. We use the chain-rule to compute the partial derivative, and then use Monte Carlo method to get an estimate. For instance, let $Y = e^{-rT}(S_T - K)^+$, where $S_T = S_0 e^{(r-\frac{\sigma^2}{2})T - \sigma\sqrt{T}Z}$ and $Z \sim \mathcal{N}(0,1)$. We note $C = \mathbf{E}[\mathbf{Y}]$, by the chain-rule, we have :

$$\frac{dY}{dS_0} = \frac{dY}{dS_T}\frac{dS_T}{dS_0} \quad \text{and} \quad \mathbf{E}[\frac{dC}{dS_0}] = \mathbf{E}[\frac{dY}{dS_0}]$$

We can compute $\frac{dY}{dS_T} = e^{-rT}\mathbf{1}_{\{S_T - K \geq 0\}}$ and $\frac{dS_T}{dS_0}$ since we have the closed-form expression. We then have :

$$\mathbf{E}[\frac{dC}{dS_0}] = \mathbf{E}[\frac{dY}{dS_0}] = \mathbf{E}[e^{-rT}\mathbf{1}_{\{S_T - K \geq 0\}}\frac{S_T}{S_0}]$$

By the S.L.L.N, we can find an estimator $\hat{\Delta}$ of Δ, by generating M price trajectories with final value $S_{T,i}$. We then have : $\hat{\Delta} = \frac{1}{M}\sum_{i=1}^{M} e^{-rT}\mathbf{1}_{\{S_{T,i} - K \geq 0\}}\frac{S_{T,i}}{S_0}$

In the same manner, we can derive the pathwise estimator for ν. We have

$$\frac{dS_T}{d\sigma} = (\sigma T - \sqrt{T}Z)S_T = S_T(log(\frac{S_T}{S_0}) - (r + \frac{\sigma^2}{2})T)\frac{1}{\sigma}$$

Following this, we have that :

$$\mathbf{E}[\frac{dC}{d\sigma}] = \mathbf{E}[\frac{dY}{d\sigma}] = \mathbf{E}[e^{-rT}\mathbf{1}_{\{S_T - K \geq 0\}}S_T(log(\frac{S_T}{S_0}) - (r + \frac{\sigma^2}{2})T)\frac{1}{\sigma}$$

With the same notation as before, we derive the unbiased estimator for ν, noted

$$\hat{\nu} = \frac{1}{M}\sum_{i=1}^{M} e^{-rT}\mathbf{1}_{\{S_{T,i} - K \geq 0\}}(log(\frac{S_{T,i}}{S_0}) - (r + \frac{\sigma^2}{2})T)\frac{1}{\sigma}$$

However, we cannot derive the pathwise estimator for γ. The lack of smoothness of $x \rightarrow \mathbf{1}_{\{x-K \geq 0\}}$ does not allow us to interchange the derivative and the expectation operator in :

$$\frac{\partial^2}{\partial S_0^2}\mathbf{E}[Y] = \frac{\partial}{\partial S_0}(\frac{\partial}{\partial S_0}\mathbf{E}[Y]) = \frac{\partial}{\partial S_0}\mathbf{E}[\frac{\partial Y}{\partial S_0}] = \frac{\partial}{\partial S_0}\mathbf{E}[e^{-rT}\mathbf{1}_{\{S_T - K \geq 0\}}\frac{S_T}{S_0}]$$

B Lévy's approximation for an arithmetic Asian option

$$\begin{cases} \mathbb{E}[A] = \mathbb{E}[Z] \\ \mathbb{E}[A^2] = \mathbb{E}[Z^2] \end{cases}$$

which is strictly equivalent to :

$$\begin{cases} \mathbb{E}[Z] = \mathbb{E}\left[e^{a+bX}\right] = e^{a+\frac{1}{2}b^2} \\ \mathbb{E}[A] = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[S_{t_i}\right] = \frac{1}{n}\sum_{i=1}^{n} Se^{rt_i} \end{cases}$$

and :

$$\begin{cases} \mathbb{E}\left[Z^2\right] = \mathbb{E}\left[e^{2a+2bX}\right] = e^{2a+2b^2} \\ \mathbb{E}\left[A^2\right] = \frac{1}{n^2}\sum_{i,j=1}^{n}\mathbb{E}\left[S_{t_i}S_{t_j}\right] = \frac{1}{n^2}\mathbb{E}\left[\sum_{i=1}^{n} S_{t_i}^2 + 2\sum_{1\leq i<j\leq n} S_{t_i}S_{t_j}\right] \end{cases}$$

In the Black-Scholes model, we get :

The first equation can be rewritten as:

$$e^a e^{b^2/2} = \frac{1}{n} \sum_{i=1}^{n} S e^{r t_i}$$

The second equation can be rewritten as:

$$e^{2a+2b^2} = \frac{1}{n^2} \sum_{i=1}^{n} S^2 e^{\left(2r+\sigma^2\right) t_i} \left(1 + 2 \sum_{j>i}^{n} e^{r(t_j - t_i)}\right)$$

C Code

We create class PATHGENERATOR which takes as input the market and model parameters. The core function of this method is the GENERATE_PATH method which, given N the number of time steps and M the number of price trajectories, returns a $(N+1)\mathrm{x}M$ matrix that contains the generated prices.

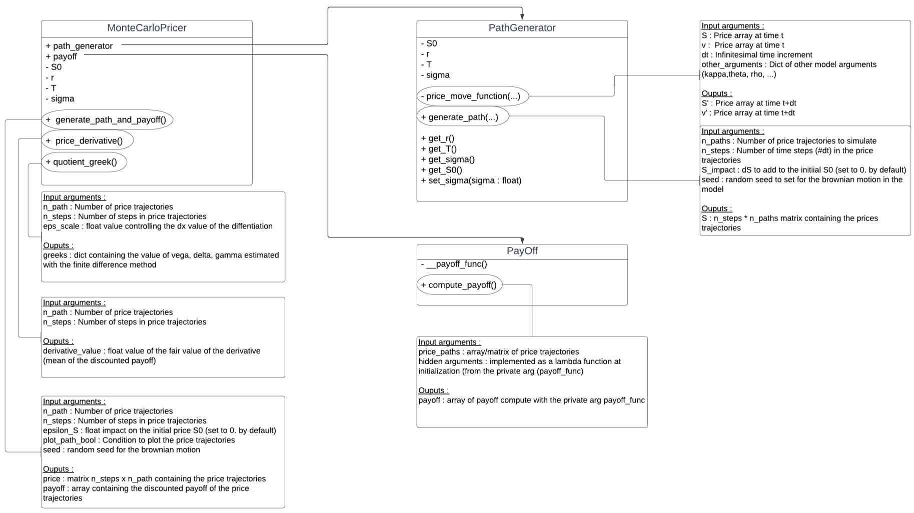


Figure 6: Extended UML diagram of the code structure

References

[1] Aleksandra Mraovic and Qian Zhang. *Valuation of Asian Options with Levy Approximation.* 2014. URL: https://lup.lub.lu.se/luur/download?func=downloadFile&recordOId=4301159&fileOId=4301160.

[2] Harry Zheng. "MATH70113 - Simulation Methods for Finance ". In: *Imperial College London - Department of Mathematics* (2022).