Monte Carlo and Finite Difference Methods in pricing financial derivatives

Group Number: Group 1 (6)

SID: 520360707

Link to code file:
https://drive.google.com/file/d/1XVi9DxLLkG1LnhFE9C1MMs0x_6J_wzWm/view?usp=sharing

Table of Contents

1. Introduction (Executive Summary)

This report will go through the comparison of Monte Carlo and Finite Difference methods, with a focus on their application in financial modeling. The insights provided will be targeted towards quantitative analysts and also professionals with moderate understanding of mathematics. Both Monte Carlo and Finite Difference methods have different charms.

Monte Carlo methods are widely used in finance for risk analysis, option pricing, and portfolio management. They are flexible and simple to understand but might be computationally expensive. As for the Finite Difference Method, they usually have higher precision and smooth problems but it is more complex and might be more computationally expensive for high dimensional settings. This report will explore and compare the different usages and different charms of these two methods.

2. Brief background on Monte Carlo and Finite Difference Methods and it's application to derivative pricing.

Both methods are powerful numerical approaches widely used in financial models, particularly in derivative pricing. They both have their own strengths and weaknesses, but they are still integral tools in pricing derivatives and managing financial risk.

Monte Carlo Methods relies on statistical sampling to simulate the behavior of underlying assets and assess potential outcomes of financial derivatives. By generating numerous random scenarios (simulations) of asset price movements, Monte Carlo allows for the estimation of the expected value of the derivative, which is often used to determine its fair price. Monte Carlo is known to be versatile and is suitable for high dimensional problems, but might require a large number of simulations for a more accurate result.

Finite Difference Methods solve partial equations that describe the price evolution of derivatives over time. In financial modeling , this method is commonly applied to price options by solving the Black-Scholes equation. This method discretized both time and asset prices into a grid and approximates the solution using iterative calculations. The Finite Difference Method is very accurate when dealing with well defined boundary conditions, such as European or American options, but gets more and more complex and less efficient for higher dimensional problems like options on multiple assets.

3. Brief background on derivatives (options), and why this contracts are important

Derivatives, especially options, play a critical role in modern finance by enabling market participants to hedge risks, speculate on future asset prices, and enhance portfolio management. Options give the holder the right (but not the obligation) to buy or sell an asset at a predetermined price, providing flexibility in managing price volatility.

These financial instruments help institutions and investors manage risks associated with currency fluctuations, interest rates, and commodity prices. By offering a mechanism for price discovery and risk transfer, derivatives contribute to market efficiency and liquidity. However, as highlighted by the 2008 financial crisis, improper use or excessive leverage of these instruments can lead to systemic risks. Therefore, strong regulatory frameworks are necessary to mitigate these risks and ensure the stability of financial markets *(Joanna B³ach, 2011).*

They help with ***risk management***, allowing institutions and investors to hedge against price fluctuations in assets like stocks, currencies, and commodities. Options also contribute to a more ***accurate market pricin***g by reflecting expectations of future volatility and asset values. Moreover, derivatives ***enhance liquidity*** by allowing more trading strategies. These factors improve the overall market efficiency and stability.

4. Monte Carlo Method

4.1 Basic Idea

Definition:
The Monte Carlo method uses random sampling to estimate numerical results. Iit simulates asset price paths to estimate the value of derivatives. The core idea is that by averaging the outcomes of a large number of simulations, one can approximate the expected value of a financial instrument like an option. It works well for complex models, especially those with path dependencies.

Why Monte Carlo Works:
Monte Carlo works as it is flexible and easy to implement to complex and high dimensional problems, though can be computationally expensive. It is also slow but steady, and requires many simulations to get closer to precise and accurate value. It improves as number of simulation increases

Numerical Example:
We will use Monte Carlo to estimate the value of pi. By the help of the library Random, we are able to apply Monte Carlo in Python. We can achieve it by resampling it on 100, 1000, 10000, and 100000 simulations. The plot below provides an illustration that shows how as the number of simulations increases, it approaches the value of pi.

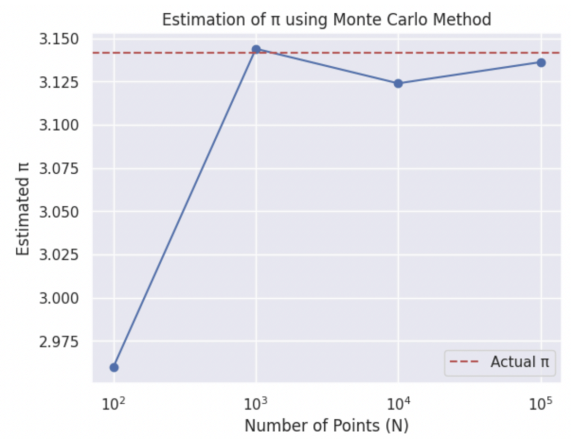


Figure 1 Estimation of pi with Monte Carlo

Factors Table: Monte Carlo Method

Metric	**Monte Carlo**
Efficiency	Moderate to low, depending on complexity
Convergence	Slow, more simulation more accurate
Precision	Improve as simulation increases
Scalability	Highly scalable to high dimension problems
Applicability	Effective for complex, path-dependent models
Computational Cost	HIgh (large number of simulations)

Figure 2 General characteristics of Monte Carlo

4.2 Monte Carlo for European Call Option

4.2.1 Standard Monte Carlo for European Call

The S&P 500 MINI SPX OPTIONS INDEX (^XSP) is chosen as the underlying asset with an initial stock price of $552.89 and a strike price of $330. The option has a maturity of 1.04 years, calculated from the 380 days between September 4, 2024, and September 19, 2025. The risk-free interest rate is 4.38%. These values are used to simulate future stock prices and estimate the price of a European call option. We will use Monte Carlo to show how the option price estimation changes to increased simulations.

Parameter Description (Short)

- Initial Stock Price (S_0) : $552.89
- Strike Price (K) : $330
- Time to Maturity (T) : 1.04 years
- Risk-Free Interest Rate (r) : 4.38% (0.0438)
- Volatility (σ) : 54% (0.54)
- Number of Simulations (N) : 100, 500, 1,000, 5,000, 10,000, 50,000, 100,000
- Payoff Calculation : $\max(S_T - K, 0)$, discounted by e^{-rT}
- Black-Scholes Price (bs_price) : Calculated for comparison

Geometric Brownian Motion for Simulating Terminal Stock Price

$$S_T = S_0 \cdot \exp\left(\left(r - \frac{\sigma^2}{2}\right) T + \sigma\sqrt{T} Z\right)$$

Substitute these values into the equation

$$S_T = 552.89 \cdot \exp\left(\left(0.0438 - \frac{0.54^2}{2}\right) \cdot 1.04 + 0.54 \cdot \sqrt{1.04} \cdot 0\right)$$

Simplify the term inside the exponential

$$0.0438 - \frac{0.54^2}{2} = 0.0438 - 0.1458 = -0.102$$

Then:

$$S_T = 552.89 \cdot \exp(-0.102 \cdot 1.04) = 552.89 \cdot \exp(-0.10608)$$

Calculate

$$\exp(-0.10608) \approx 0.8993$$

Finally:

$$S_T = 552.89 \cdot 0.8993 \approx 240.69$$

The plot below shows the computation of the performance of estimation from monte carlo increases as simulation increases.

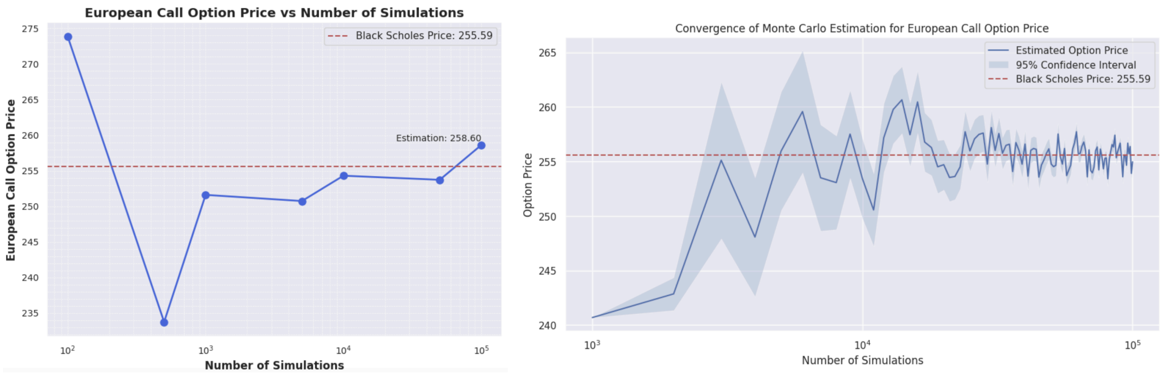


Figure 3&4 Estimate European Call Option price as simulation increases & convergence

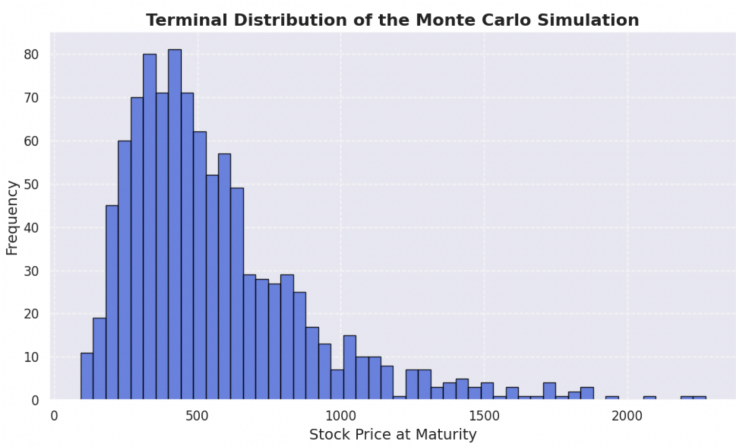


Figure 5 Distribution of MC estimates on option price

It shows that as simulation increases, monte carlo slowly approaches accurate value with smaller variations and tighter confidence intervals. This shows the importance of increasing the number of simulations to reduce variance in Monte Carlo estimates. The terminal distribution of the Monte Carlo simulation indicates that distribution is right-skewed, with most stock prices clustering around the lower values, reflecting the random walk nature of the Geometric Brownian Motion used in the simulation. These figures highlight the effectiveness of Monte Carlo methods in pricing options and the convergence behavior of such simulations.

4.2.2 Monte Carlo Antithetic for European Call

Using the same parameters, we estimate the European call option price using the antithetic variates method. Here is how we can do it:

We simulate two terminal stock prices using the random variables:

$$S_T^+ = S_0 \cdot \exp\left(\left(r - \frac{\sigma^2}{2}\right)T + \sigma\sqrt{T}Z\right)$$

$$S_T^- = S_0 \cdot \exp\left(\left(r - \frac{\sigma^2}{2}\right)T + \sigma\sqrt{T}(-Z)\right)$$

The payoff for a European call option is the maximum of the stock price minus the strike price, or zero.

$$\text{Payoff}^+ = \max(S_T^+ - K, 0)$$

$$\text{Payoff}^- = \max(S_T^- - K, 0)$$

The Monte Carlo estimator using the antithetic variates method averages the payoffs from and discounts them to the present:

$$C_{\text{antithetic}} = e^{-rT} \cdot \frac{1}{2}\left(\frac{1}{N} \sum_{i=1}^{N} \left(\text{Payoff}_i^{+} + \text{Payoff}_i^{-}\right) \right)$$

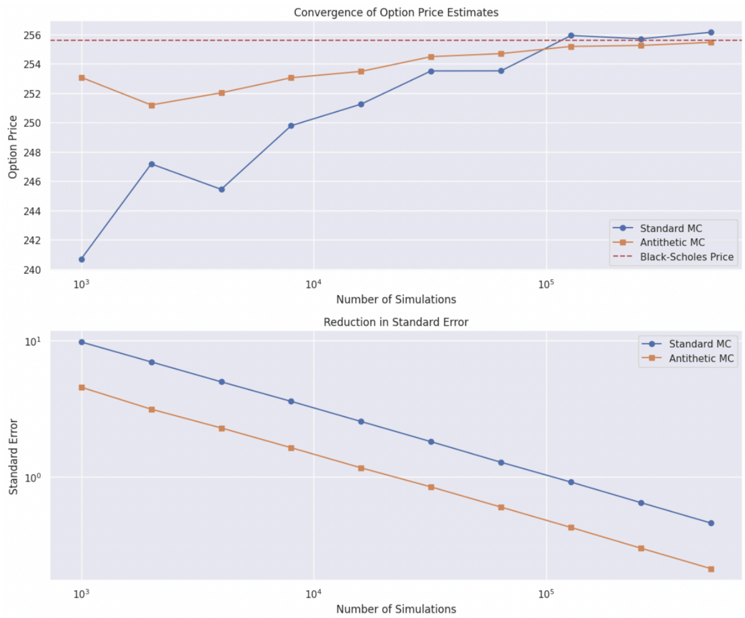


Figure 6 Comparison of Antithetic and Standard MC

The top graph shows that the antithetic method consistently produces option price estimates closer to the Black-Scholes price (255.59) and converges more quickly than the standard MC as the number of simulations increases. The antithetic method is more stable and accurate with less simulations. In the bottom graph, the antithetic method also demonstrates a significant reduction in standard error compared to the standard MC method. This lower error translates to more efficient convergence. This is because the antithetic variates approach reduces the variance of the estimates, making it a more efficient choice for this type of simulation.

4.2.3 Multilevel Monte Carlo for European Call

Multilevel Monte Carlo (MLMC) methods in numerical analysis are algorithms for computing expectations that arise in stochastic simulations. Just as Monte Carlo methods, they rely on repeated random sampling, but these samples are taken on different levels of accuracy.*(Giles, M.B. (2008)*

The final estimated option price is the average across all levels:

$$C_{\text{MLMC}} = \frac{1}{L} \sum_{l=0}^{L-1} P_l$$

MLCM is used to improve efficiency of MC estimators by dividing computation across multiple levels increasing resolution.

$$P_l = e^{-rT} \cdot \frac{1}{M} \sum_{i=1}^{M} \text{Payoff}(S_T)$$

Using the same parameter values we can compute the graph below:

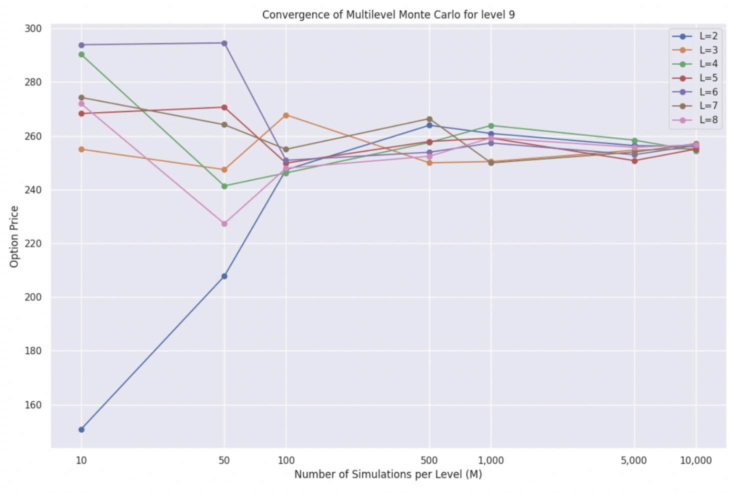


Figure 7 Multilevel MC for level 9

As the number of simulations per level increases, the option price estimates become a more stable value, reducing variance and increasing accuracy. The lower levels are more volatile at lower numbers of simulations, while higher levels converge more quickly and consistently. This demonstrates that higher levels of MLMC provide more accurate and stable estimates, especially as the number of simulations increases.

4.2.4 Quasi Monte Carlo for European Call

The QMC method uses the Sobol sequence, to generate the random variables. This results in more uniform sampling compared to standard Monte Carlo methods. This helps achieve faster convergence.

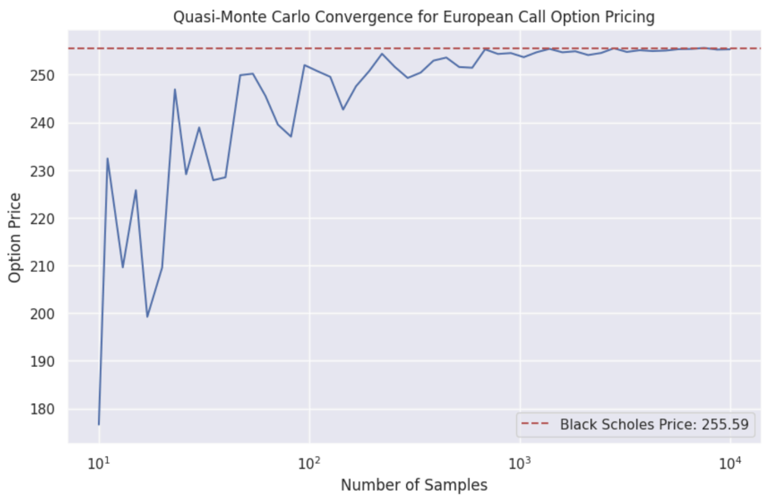


Figure 8 Quasi MC graph

As the sample size increases, the estimated option price oscillates initially but gradually converges toward the theoretical Black-Scholes. The fluctuations are significant at lower sample sizes, showing increased variability when fewer points are used, but as the number of samples grows, the QMC method stabilizes and approaches the actual price.

4.2.5 Overall calculation for the Monte Carlo methods

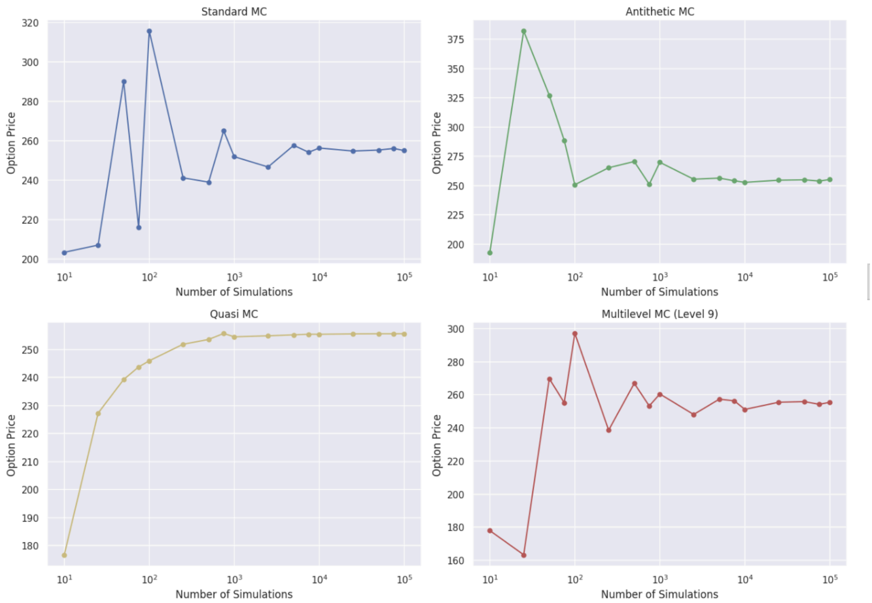


Figure 9 Comparison of estimation performance for MC methods

Metric	Standard Monte Carlo	Antithetic Variates	MLMC	QMC Sobol
Black-Scholes Price	255.59		255.5882	255.5882
Option Price	240.6885	253.0721	255.7662	254.4327
Absolute Error	14.8997	2.5161	0.1780	1.1554
Relative Error	0.0583	0.0098	0.0007	0.0045
Computation Time (s)	0.0001	0.0012	0.0784	0.0014
Standard Error	9.8075	4.5552	4.1447	1.7884
Efficiency	1.0610e+02	3.9822e+01	7.4247e-01	2.2541e+02
Precision	0.1020	0.2195	0.2413	0.5592

Figure 10 Comparison of Evaluation Metrics

From the result on the graph,we can say that the standard MC method shows slower convergence with higher variability, while antithetic variates and QMC provide much smoother and faster convergence towards the Black-Scholes price. MLMC shows moderate fluctuation at low simulation levels but stabilizes at higher levels.

From figure 10, it can be concluded that antithetic variates and QMC exhibit the lowest absolute errors and standard errors, showing higher precision and efficiency in comparison to the other methods. MLMC also performs well, achieving the lowest relative error but it has higher computation time. Overall, QMC appears to be the most efficient and precise method, closely followed by the antithetic variates method.

4.3 Random path

4.3.1 Random path with 1000 simulations

The random path is generated by simulating the effects of the drift (growth) and volatility (random fluctuations) of the asset price over time. This is essential for option pricing, particularly for pricing path-dependent options (e.g., barrier options), where the option's payoff depends on the path taken by the asset price.

Example:Random path for European call for 1000 simulations

We first generate a random path with the wiener process by summing normal random variables representing daily time steps. The asset prices for each time point is then calculated using GBM formula, then plot and show 20 random simulated paths of the asset price over time and also a horizontal line for the strike price.

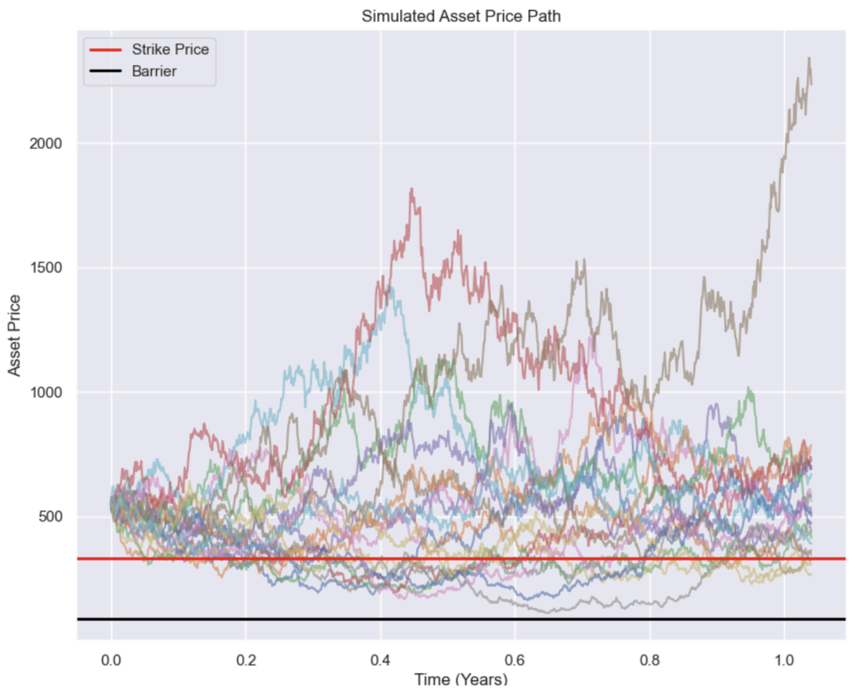


Figure 11 Simulated asset price path

While many paths initially hover around the strike price, only a few paths rise significantly above it, indicating potential favorable outcomes for a call option. Some paths remain below or around the barrier level, which may impact options like barrier options, where a breach can terminate the option.

This highlights the stochastic nature of asset prices and the uncertainty in predicting their movements. This model is useful for option pricing, where understanding the possible range of outcomes can help in determining fair option prices.

5. Finite Difference Method

5.1 Finite Difference Method explanation and example

Definition:

The Finite Difference Method (FDM) is a numerical technique used to solve partial differential equations (PDEs). In this case, this method is commonly employed to price complex financial instruments where analytical solutions might be difficult or impossible to derive.

The domain is differentiated into a grid and simulation is done for temporal and spatial variations. Simulations can be done in forward, backward or central differential form.

Numerical Example: Estimating pi value with FDM

$$y''(t) + y(t) = 0$$
$$y(0) = 1 \quad and \quad y'(0) = 0$$

Find t when y(t)= –1, Using a symmetric difference we get

$$\frac{d^2y(t)}{dt^2} \approx \frac{y(t+\Delta t) - 2y(t) + y(t-\Delta t)}{\Delta t^2}$$

$$\therefore\ y(t+\Delta t) - 2y(t) + y(t-\Delta t) + \Delta t^2 y(t) = 0$$

$$y(t+\Delta t) = 2y(t) - y(t-\Delta t) - \Delta t^2 y(t)$$

Can be rewritten as:

$$y_{n+1} = 2y_n - y_{n-1} - \Delta t^2 y_n$$

We can use forward difference for y1:

$$\frac{dy(t)}{dt} \approx \frac{y(t+\Delta t) - y(t)}{\Delta t}$$

$0 \approx \frac{y(\Delta t) - y(0)}{\Delta t}$, so $y(\Delta t) = 1$

This gives us y1, but we can also use a Taylor expansion to get y1:

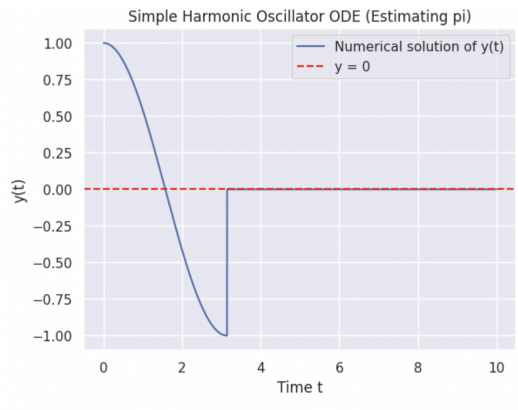
$$y(\Delta t) \approx y(0) + y'(0)\Delta t + \frac{1}{2}y''(0)\Delta t^2$$

$$= y(0) + y'(0)\Delta t + \frac{1}{2}(-y(0))\Delta t^2$$

$$y(\Delta t) \approx y(0) - \frac{1}{2}y(0)\Delta t^2$$

So finally we can write as:

$$y_1 = y_0 - \frac{1}{2}\Delta t^2 y_0$$



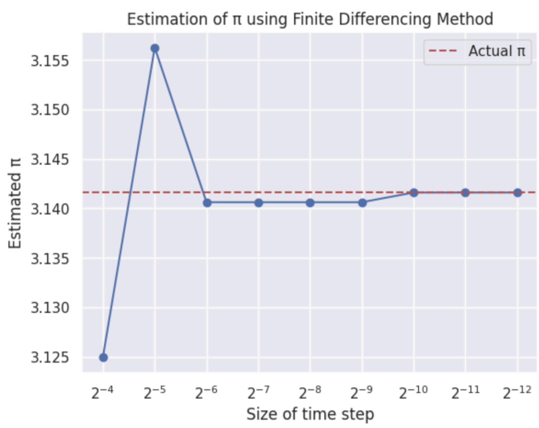


Figure 12 and 13 Simple harmonic oscillator ODE and estimation as simulation increase

The estimate is very accurate, showing the instability that can occur for certain step sizes and how the method converges to the correct value as the time step becomes sufficiently small.

Properties of FDM:

Metric	Monte Carlo
Efficiency	High for Low dimension, Low for High dimension
Convergence	Fast for small time steps, but depends on grid size and stability
Precision	Can get high but sensitive to step size
Scalability	Limited scalability
Applicability	For pricing vanilla options, especially European Options
Computational Cost	Moderate to high, depends on implementation

Figure 14 Properties of Finite Difference Method

5.2 Finite Difference Method for European Call PDE

5.2.1 Explicit Euler Method for European Call PDE

The Explicit Euler Method is a straightforward approach for numerical integration to solve ordinary differential equations (ODEs) by stepping forward in increments of time using the derivative's current value. We want to Simulate the price evolution of a European call option as time progresses towards maturity, using the Explicit Euler method.

Parameters:

- **Stock price (S)** is constant at $165.
- **Strike price (K)** of the option is $100.
- **Time to maturity (T)** is 1 year.
- **Risk-free rate (r)** is (5%) per annum.
- **Volatility (σ)** of the stock is 20% per annum.
- **Time step (Δt)** for our simulation is 0.01 years (about 3.65 days).

$$C(S,t) = S \cdot N(d_1) - K \cdot e^{-r(T-t)} \cdot N(d_2)$$

Explicit Euler Method for Solving the Heat Equation (PDE):

The main loop implements the Explicit Euler method for solving the Black-Scholes Partial Differential Equation (PDE). The Black-Scholes PDE for option pricing is:

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0$$

For each time step:

- **First derivative** of option price with respect to stock price (\frac{\partial V}{\partial S}) is computed as:

$$\frac{\partial V}{\partial S} \approx \frac{V[j+1] - V[j-1]}{2 \cdot dS}$$

- **Second derivative** (\frac{\partial^2 V}{\partial S^2}) is computed as:

$$\frac{\partial^2 V}{\partial S^2} \approx \frac{V[j+1] - 2V[j] + V[j-1]}{dS^2}$$

- The new option value is updated using the explicit Euler formula:

$$V[j] = V[j] + dt \cdot \left(0.5\sigma^2 S_j^2 \frac{\partial^2 V}{\partial S^2} + rS_j \frac{\partial V}{\partial S} - rV[j]\right)$$

We can also use the Heat Equation method, where we transform V and the other variables to give us a function $u(t, x)$ that solves:

$$u_t - u_{xx} = 0, t > 0$$
$$u(0, x) = max[e^{\frac{q-1}{2}x} - e^{\frac{q+1}{2}x}, 0]$$

After computing these formulas, we have plots shown at Figure 15 and 16. It can be concluded that the explicit scheme shows reasonable estimates for moderate to high stock prices. It follows the Black Scholes formula well. Figure 16 shows a spike in lower prices stocks showing that there might be instability for lower stock values but at the end still estimates the actual price pretty well.

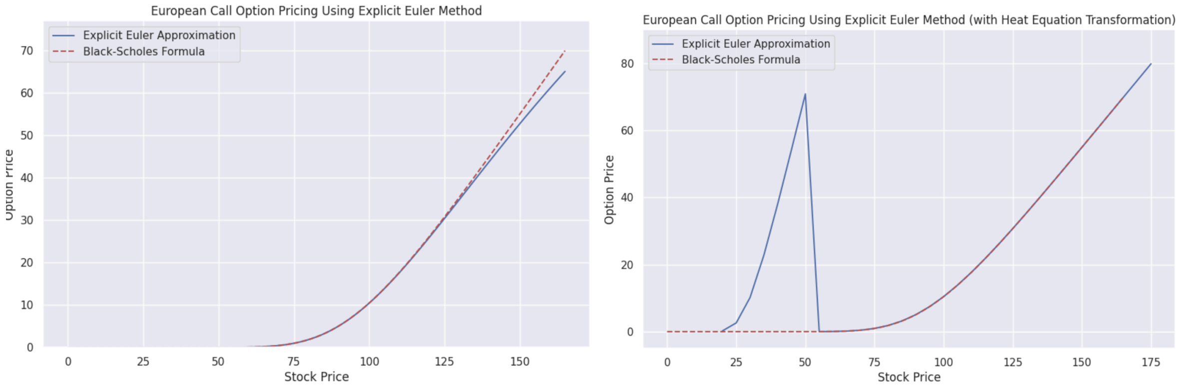


Figure 15 & 16 European call option pricing estimate with Explicit method & Modified with heat transformation

5.2.2 Implicit Euler Method for European Call PDE

The Implicit Euler method is a backward time-stepping method, meaning it uses the function values at the next time step to compute derivatives. This approach can provide greater stability, especially for stiff equations, allowing for larger time steps without numerical instability.

Given the same parameter, we can use the formula to compute option price estimation.

Given an ODE:

$$\frac{dy}{dt} = f(t, y)$$

The Implicit Euler update formula is:

$$y_{n+1} = y_n + \Delta t \cdot f(t_{n+1}, y_{n+1})$$

This requires solving an equation for y_{n+1} because f depends on y_{n+1}, typically necessitating iterative methods or solvers, especially when f is nonlinear.

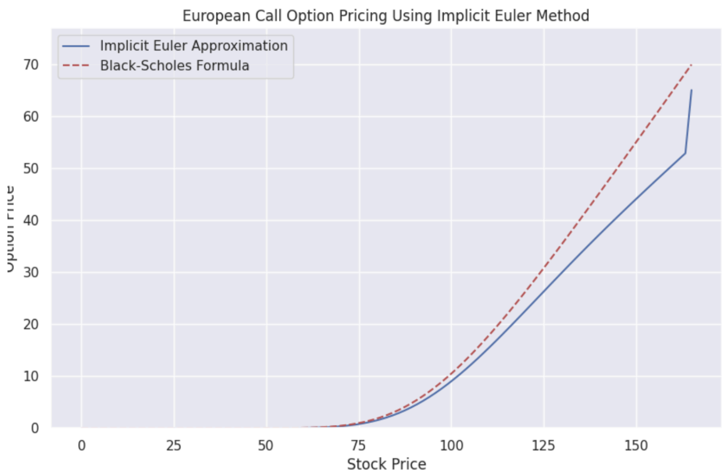


Figure 17 European call option pricing estimate with Implicit method

The graph shows close alignment of implicit estimation with the black scholes price, but there is a noticeable deviation for higher stock prices and it overestimates the price. This shows that while this method provides a stable and accurate approximation, it becomes less accurate as stock prices increase. This can be due to the implicit behavior, where they can be more stable but less accurate.,

5.2.3 Crank Nicolson Method for European Call PDE

The Crank-Nicolson method is a finite difference method used to numerically solve partial differential equations (PDEs), particularly parabolic PDEs, like the Black-Scholes equation used in option pricing. It is a second-order accurate method in both space and time, and it combines both implicit and explicit schemes to achieve a balance between stability and accuracy.

Consider these assumptions:

- **S_max** = 200 (Maximum stock price)
- **K** = 100 (Strike price)
- **T** = 1.0 (Time to maturity in years)
- **r** = 0.05 (Risk-free interest rate)
- **sigma** = 0.2 (Volatility of the underlying stock)
- **M** = 100 (Number of spatial grid points for stock prices)
- **N** = 1000 (Number of time steps)
- **dt** = T / N (Time step size)
- **dS** = S_max / M (Stock price step size)

And consider the Crank Nicolson Formula:

For a PDE of the form:

$$\frac{\partial V}{\partial t} = \mathcal{L}(V)$$

where L(V) represents spatial derivatives, the Crank-Nicolson discretization is:

$$\frac{V_i^{n+1} - V_i^{n}}{\Delta t} = \frac{1}{2}\left(\mathcal{L}(V_i^{n+1}) + \mathcal{L}(V_i^{n})\right)$$

Here, Vi^n represents the value of the option at spatial point i and time step n, and L(V) includes first and second derivatives with respect to stock price S.

Finite Difference Discretization:

For the Black-Scholes PDE:

$$\frac{\partial V}{\partial t} = \frac{1}{2}\sigma^2 S_i^2 \frac{\partial^2 V}{\partial S^2} + rS_i \frac{\partial V}{\partial S} - rV$$

The Crank-Nicolson scheme for each time step can be written as:

$$\frac{V_i^{n+1} - V_i^{n}}{\Delta t} = \frac{1}{2}\left(\frac{1}{2}\sigma^2 S_i^2 \frac{V_{i-1}^{n+1} - 2V_i^{n+1} + V_{i+1}^{n+1}}{\Delta S^2} + rS_i \frac{V_{i+1}^{n+1} - V_{i-1}^{n+1}}{2\Delta S} - rV_i^{n+1}\right)$$
$$+\frac{1}{2}\left(\frac{1}{2}\sigma^2 S_i^2 \frac{V_{i-1}^{n} - 2V_i^{n} + V_{i+1}^{n}}{\Delta S^2} + rS_i \frac{V_{i+1}^{n} - V_{i-1}^{n}}{2\Delta S} - rV_i^{n}\right)$$

This results in a tridiagonal system of equations that can be solved efficiently.

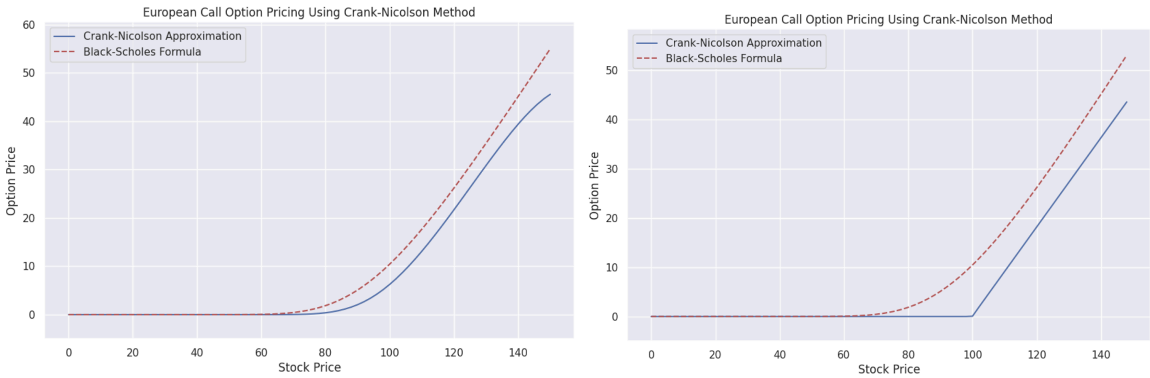


Figure 18 & 19 European call option pricing estimate with Crank-Nicolson

The graph produced shows that for lower stock prices, Crank Nicolson shows a fairly accurate approximation of the black scholes price especially for stock prices below strike price. In this case the Crank Nicolson slightly underestimates the option price at higher stock prices, this might be due to numerical limitations of the method.

5.3 Overall Calculation for Finite Difference Methods
5.3.1 Finite Difference Methods Evaluation

From all the computation of FDM methods we can then evaluate some metrics to compare how they predict European Call PDE.

	Explicit Euler	Implicit Euler	Crank-Nicolson
Average Black-Scholes Price	11.424598	11.424598	54.970140
Average Option Price	10.893482	3.708601	50.658350
Average Absolute Error	0.531191	7.715998	4.311790
Average Relative Error	0.033713	0.315921	0.078439
Computation Time (s)	0.600650	0.057308	0.035759
Standard Error	0.112033	1.358462	1.988291
Efficiency	1.326445	0.094556	14.064943
Precision	0.033713	0.315921	1.884096

Figure 20 Metrics Measurement for FDM methods

Explicit Euler provides a faster computation time but comes with a trade-off in accuracy, especially for larger stock prices. Implicit Euler, while offering slightly better stability, shows a higher absolute error and longer computation time compared to Explicit Euler. These metrics indicate that while Explicit and Implicit Euler have their uses, they become less reliable as the complexity and size of the problem grow.

Though it might not seem like it, the Crank-Nicolson method stands out as the most balanced between accuracy and stability among the three. Its average option price is closer to the Black-Scholes reference value, with small errors in both absolute and relative terms. This method demonstrates its ability to handle larger stock prices without significant deviation from the true option value, which makes it more reliable across different price levels. While Crank-Nicolson has a slightly higher standard error than the Explicit Euler method, its overall performance especially in maintaining stability for higher stock prices proves it to be the best among the three methods. Its balance of computational speed and accuracy reinforces its reputation as a preferred method for option pricing.

6. Comparison of Monte Carlo and Finite Difference method

6.1 Advantages and Dis-advantages

Monte Carlo

These methods are highly flexible and can handle multi dimensional problems well.They perform particularly well when dealing with high-dimensional settings, and their simulations can easily be parallelized, which speeds up computations significantly. This flexibility allows Monte Carlo to tackle complex option pricing problems across multiple variables. However, the convergence rate is relatively slow compared to other methods, and it can be inefficient for simpler options. Moreover, Monte Carlo can be less accurate for short-term options, where rapid convergence is crucial.

Finite Difference

Finite Difference Methods (FDM), on the other hand, are highly accurate for pricing European call and put options, especially those governed by the Black-Scholes Partial Differential Equation (PDE). FDM converges faster than Monte Carlo for one-dimensional problems, such as single-asset European options, and is most effective in low-dimensional problems with a well-structured computational grid. However, FDM becomes difficult to implement and computationally expensive in high-dimensional settings. It also requires clear boundary conditions, which can be challenging for certain complex options. Additionally, FDM is less suited for path-dependent options, where Monte Carlo might be more appropriate.

6.2 Applicability in different cases

Monte Carlo methods are Ideal for *path-dependent* options where the payoff depends on the historical prices of the underlying asset. They also work well for pricing basket options or other *options involving multiple underlying assets. Complex exotic options* with non-standard payoffs (e.g., rainbow options or options with stochastic volatility)

Finite Difference is best suited for *vanilla European call and put* options where the payoff structure is simple and can be modeled using the Black-Scholes PDE. It's also efficient for *one-dimensional problems* like single-asset options. Furthermore, it works well for *short-term options*, as the finite difference grid can handle fine time steps without the convergence issues faced by Monte Carlo method.

7. Conclusion and Summary

We have explored two key numerical methods for derivative pricing: Monte Carlo and the FInite Difference Method. For Monte Carlo techniques: Standard, Antithetic, Multilevel, and Quasi to price European call options. We also examined random path generation for asset prices. For the FDM, we focused on European Call PDE using the explicit Euler, implicit Euler, and Crank-Nicolson methods.

By comparing these methods, we highlighted that Monte Carlo is flexible and effective for high-dimensional problems but computationally expensive with slow convergence and FDM is more accurate and stable for lower-dimensional problems with well-defined boundaries, but becomes less efficient in high dimensions. For European call options, Quasi-Monte Carlo works best among the Monte Carlo methods, and the Crank-Nicolson method is the most effective among FDM approaches.

One limitation of this project is the high computing cost in order to get more accurate and specific computations. Future research could explore adaptive meshing techniques for FDM to enhance its accuracy in such scenarios. Additionally, combining Monte Carlo with FDM in hybrid approaches could help leverage the strengths of both methods for more efficient and precise derivative pricing.

8. References

Giles, M. B. (2008). Multilevel Monte Carlo path simulation. *Operations Research*, 56(3), 607–617. https://doi.org/10.1287/opre.1070.0496

Platen, P., & Rendek, K. (2011). Approximating the first passage density of Brownian motion via a random walk. *Economics & Finance*, 7(3), 13-26. Retrieved from https://ideas.repec.org/a/rze/efinan/v7y2011i3p13-26.html