Monte Carlo and Finite Difference Methods in Pricing Financial Derivatives

Group Number: 15 SID: 530473808

September 19, 2025

Contents

1 Introduction

Pricing financial derivatives accurately is essential in modern finance, particularly for managing risk and developing trading strategies. Analytical solutions like the Black–Scholes formula work well for basic contracts, but more complex or path-dependent derivatives require numerical methods. This report focuses on two of the most widely used computational techniques in quantitative finance: Monte Carlo simulation and Finite Difference methods. By analyzing both methods across different option types including European, American, and barrier we evaluate their accuracy, efficiency, and suitability for real-world applications.

2 Background

2.1 A Brief Background on Financial Derivatives

Derivatives play a critical role in today's financial markets, enabling investors to hedge risk, speculate on market movements, and enhance portfolio performance. Products like options, futures, and swaps are used across asset classes including equities, currencies, commodities, and interest rates to manage exposure and tailor risk profiles. Banks and corporations use derivatives to lock in prices, hedge against volatility, or manage cash flows, while hedge funds and traders use them to leverage positions and execute complex arbitrage strategies. The growth of structured products and volatility-linked instruments has further increased the demand for accurate and flexible pricing methods, especially as markets become more interconnected and complex.

2.2 A Brief Background to Monte Carlo and Finite Difference Methods

Monte Carlo and Finite Difference methods are core numerical tools for pricing financial derivatives. Monte Carlo estimates prices by simulating many risk-neutral asset paths and averaging discounted payoffs (Broadie & Glasserman, 1996). The method is highly flexibly, especially for path-dependent or high-dimensional options however it converges slowly without variance reduction techniques like antithetic variables, importance sampling, or control variates (Paskov & Traub, 1995). Finite Difference methods, on the other hand, discretize the Black–Scholes PDE and solve it backward in time (Tilley, 1993). Methods like Explicit, Implicit, and Crank–Nicolson offer different trade-offs between speed and stability, with Crank–Nicolson achieving second-order accuracy (Joy et al., 1996). While Monte Carlo excels in complex scenarios, Finite Difference methods are often more efficient for simple, low-dimensional problems (Boyle, 1976).

3 Monte Carlo Methods

3.1 Basic Theory and Terminal Prices

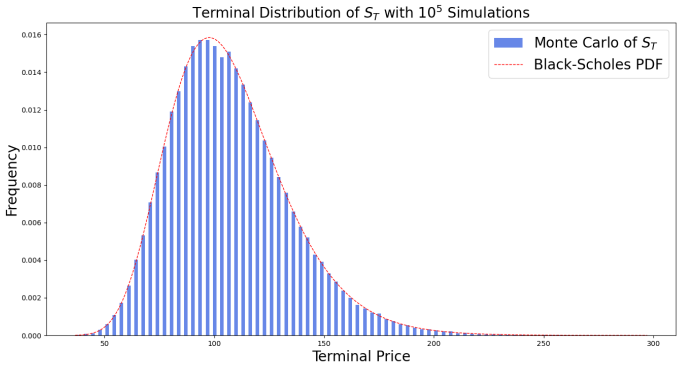

Figure 1: Terminal Price distribution of asset with initial price 100, volatility 0.25, risk-free rate 0.07 and time to maturity 1. Note the right skew of the Black-Scholes PDF as a result of the risk-neutral model.

In short, the Monte Carlo method works by repeatedly simulating the terminal price of the underlying asset. Under a risk-neutral model, an option's fair value is the average of its discounted payoff over all future outcomes

(Carriere, 1996). Therefore, by simulating a large number of terminal prices an underlying can take on, we can calculate the average discounted payoff and thus price the European option (Tilley, 1993).

Now that we know what Monte Carlo is, we can delve into proving its correctness. The method relies on one simple principle, the law of large numbers. We will discuss how this apply later. For now, recall from the Black-Scholes model:

$$S_T = S_0 e^{(r-\frac{1}{2}\sigma^2)T+\sigma\sqrt{T}X} \tag{1}$$

Under the risk-neutral model, all assets are priced such that their expected return is the risk-free rate (Duffie & Glynn, 1995). Therefore, the fair value of any derivative is the expected discounted payoff, since otherwise the investor could earn more by simply investing in a risk-free asset (Boyle, 1976). Therefore, the value of any asset is:

$$V = e^{-rT} E[f(S_T)] \tag{2}$$

Using the Black-Scholes price function for S_T, the price for a European call is:

$$\text{Price} = e^{-rT} E[(S_0 e^{(r-\frac{1}{2}\sigma^2)T+\sigma\sqrt{T}X} - K)^+] \tag{3}$$

The above equation can be derived by simply substituting the Black-Scholes price function for the underlying into the payoff of a European call $\Psi = (S_T - K)^+$, where S_T is the underlying price at expiry and K is the strike price.

How can we find the $E[f(S_T)]$ in Equation (3)? Well through the Monte Carlo method! We can simulate sample X from a normal distribution a large number of times. Using law of large numbers:

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n} X_i = \mu \tag{4}$$

Therefore, the more we sample X, the more our average approaches the true expected value! We will see later that Monte Carlo methods can also be used to generate full paths, allowing easy pricing of price dependent options like Asian, Barrier, Digital etc without solving the SDE (Duffie & Glynn, 1995).

3.1.1 Efficiency, Convergence and Precision

When generating a large number of paths for an asset's price, the Monte Carlo method can be computationally expensive (Boyle et al., 1997). We can show this using the Central Limit Theorem:

$$\frac{\bar{X}_n - \mu}{\frac{\sigma}{\sqrt{n}}} \xrightarrow{d} \mathcal{N}(0,1) \qquad \therefore \bar{X}_n - \mu \approx \frac{\sigma}{\sqrt{n}} Z, \quad Z \sim \mathcal{N}(0,1) \tag{5}$$

We can also write in terms of standard error $\text{SE} = \frac{\sigma}{\sqrt{n}}$. Therefore, we can see that the error shrinks in proportion to $\frac{1}{\sqrt{n}}$ or in the order of $\mathcal{O}(\frac{1}{\sqrt{n}})$. This is our convergence rate, which is not fast at all! Thus without any improvements, the basic Monte Carlo method is quite inprecise and inefficient.

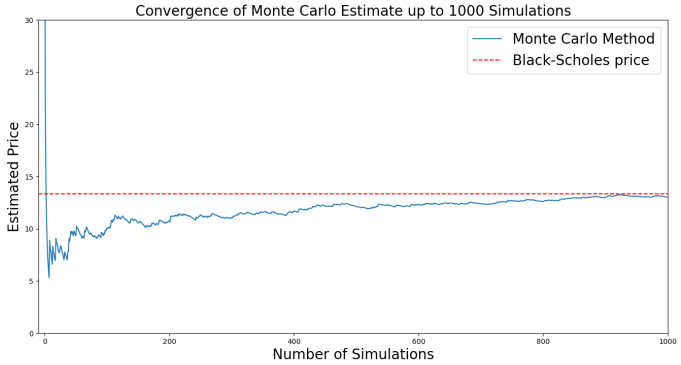

Figure 2: Convergence diagram of Monte Carlo method with no variance reduction for up to 1000 simulations.

3.2 Monte Carlo Variance Reduction Methods

From before, we estimate the error to be in the order of $\mathcal{O}(\frac{1}{\sqrt{n}})$. Given we estimate $E[f(x)] \approx \frac{1}{N}\sum f(x_i)$e have:

$$\text{Error} = \sqrt{\frac{\text{Var}[f(x)]}{n}} \tag{6}$$

Therefore, Equation (6) suggests that to reduce error, we can either increase the number of simulations or reduce $\text{Var}[f(x)]$ (Boyle, 1976). As covered before, increasing number of simulations can be computationally expensive for little improvement. Then how do we reduce $\text{Var}[f(x)]$? Well, if we are able to find a similar function $g(x)$ that has the property of having the same mean $E[g(x)] = E[f(x)]$ and with smaller variance given the same number of simulations $\text{Var}[g(x)] \ll \text{Var}[f(x)]$, we can simply substitute it in! Since we are getting closer estimate for our derivative price in the same number of simulations, in theory this method has higher efficiency, precision and faster convergence compared to our basic Monte Carlo method. Techniques that use this principle are called Variance Reduction methods (Duffie & Glynn, 1995).

3.2.1 Variance Reduction using Antithetical Variables

Since we are sampling from a normal distribution for our X, we can exploit its symmetry by pairing each sample $X \sim \mathcal{N}(0,1)$ with its antithetic counterpart $-X$. This works especially well when pricing simple options like European put/calls as they are monotonically increasing/decreasing, meaning that $f(X)$ and $f(-X)$ move in opposite direction and variance reduction is maximized when we take the average $\frac{f(X)+f(-X)}{2}$. In general, we can say:

$$\text{Var}\left(\frac{f(X)+f(-X)}{2}\right) < \text{Var}(f(X)) \qquad E\left[\frac{f(X)+f(-X)}{2}\right] = E\left[f(X)\right] \tag{7}$$

Hence, we have found a viable $g(x)$! We will look how we apply this principle in pricing a variety of options later.

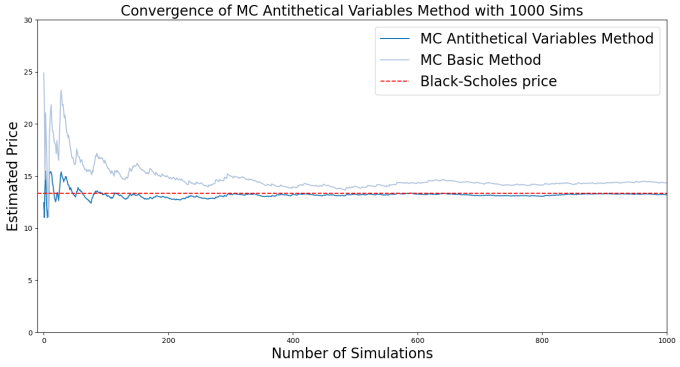

Figure 3: Convergence diagram comparing Monte Carlo method with variance reduction method using antithetical variables against basic Monte Carlo method.

3.2.2 Variance Reduction using Importance Sampling

If we simulate n paths and only a small number end up in-the-money (ITM), each of these points has higher leverage, creating high variance in our price estimate. We can reduce the variance especially for deep out-of-the-money situations by shifting our sampling distribution so that a larger proportion (typically around 50%) of terminal prices end up with a positive payoff. This technique is called Importance Sampling. Under BS pricing model, a European Call has positive payoff when:

$$S_0 e^{\left(r-\frac{1}{2}\sigma^2\right)T+\sigma\sqrt{T}X^*} - K = 0 \qquad \therefore X^* = \frac{\ln\frac{K}{S_0} - \left(r-\frac{1}{2}\sigma^2\right)T}{\sigma\sqrt{T}} \tag{8}$$

Thus to make $\mathbb{P}(\text{ITM}) \approx 50\%$, we sample from a shifted normal distribution $Y \sim \mathcal{N}(\beta, 1)$ where $\beta = X^*$. It should be clear that if we just pick a new distribution, then this violates the core principle of variance reduction in

that $E[g(x)] \neq E[f(x)]$. Therefore, how do we make sure the means are the same? Consider our $E[f(x)]$ expressed as an integrand:

$$E[f(x)] = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x) e^{-\frac{x^2}{2}} dx = \frac{1}{\sqrt{2\pi}} \int_{\infty}^{\infty} f(y-\beta) e^{-\frac{(y-\beta)^2}{2}} dy = E\left[f(x-\beta) e^{\beta x - \frac{1}{2}\beta^2}\right] = E[g(x)] \tag{9}$$

Finally, we obtain $g(x) = f(x-\beta)e^{\beta x - \frac{1}{2}\beta^2}$ that satisfies our principles of variance reduction! Again, we will look into applying this method in pricing a variety of options later.

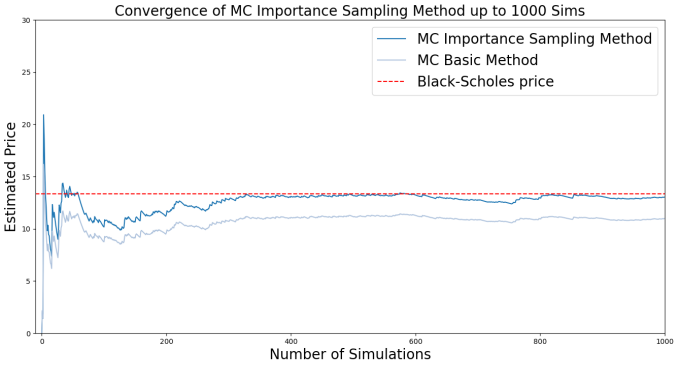

Figure 4: Importance Sampling method of variance reduction and the basic Monte Carlo method. Note the faster convergence.

3.2.3 Variance Reduction using Control Variates

The control variates method takes advantage of the fact that we are dealing with financial derivatives, and thus it is usually easy to find a second asset that you know the exact mean of and is highly correlated with your derivative's payoff (Boyle, 1976). In the case of the European call, that is simply the discounted price of the underlying! We simply subtract the functions to minimize the noise and add back the known mean. Therefore:

$$g(x) = (\underbrace{f(x) - \beta h(x)}_{\text{subtracting noise}}) + \underbrace{\beta E[(h(x)]}_{\text{adding mean}} \tag{10}$$

Through differentiating $\text{Var}[g(x)]$, we can pick $\beta_{\min}$ that minimizes variance.

$$\beta_{\min} = \frac{\text{Cov}[f(x), h(x)]}{\text{Var}[h(x)]} \tag{11}$$

Two things to note are that for this method to work we will need $\mu \ll N$ and since $\text{Var}_{min}[f(x)] = (1-\rho^2)\text{Var}[f(x)]$ where ρ is the correlation between $f(x)$ and $h(x)$, it is important to choose $h(x)$ to maximize ρ. Like before, we will see this principle in action when pricing a variety of options later.

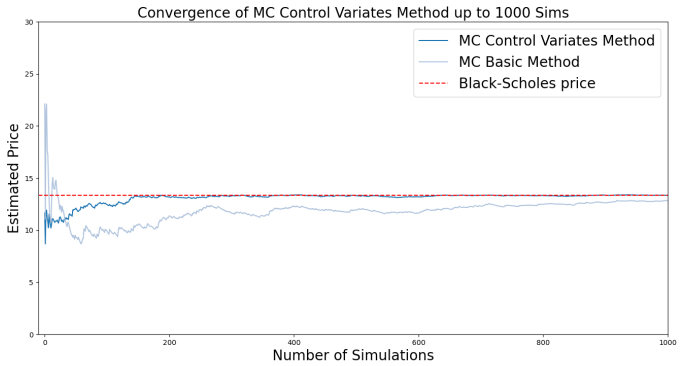

Figure 5: Monte Carlo using control variates method of variance reduction plotted with basic Monte Carlo method. Convergence depends on correlation of $h(x)$.

3.3 Random Paths

So far, we have only tried pricing path-independent derivatives such as the humble European option. However, how do we apply the Monte Carlo method to generate full paths and thus price path-dependent derivatives? Likewise, the Monte Carlo method can be used to repeatedly simulate the Stochastic Differential Equation (SDE) of the underlying asset. In doing so, we can simulate the potential path price of the asset could follow over a number of time steps under a risk-neutral model.

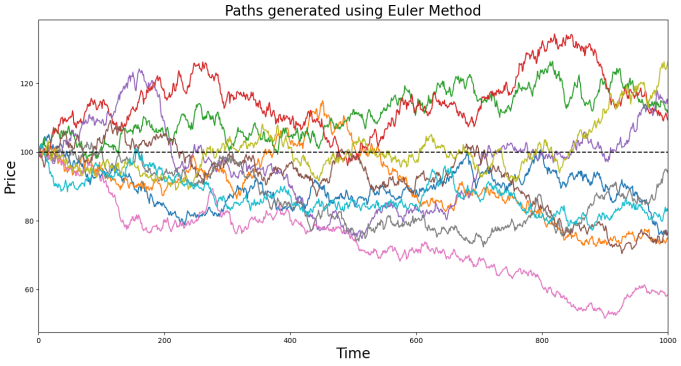

Figure 6: 10 potential paths of an asset's price given initial price 100, volatility 0.25, risk-free rate 0.07 and time to expiry 1. Pay no attention to 'Euler Method', we will go into that in the following section.

Under risk neutral dynamics, the Black-Scholes Model can be written as Ito SDE:

$$dS_t = rS_t dt + \sigma S_t dW_t \tag{12}$$

Which describes the paths an asset can take. Several numerical schemes exist to simulate the Black-Scholes SDE, with the Euler-Maruyama method and Milstein's method being among the most famous.

3.3.1 Euler-Maruyama Method

Suppose we split the timeline of an asset $[0, T]$ into m equal time steps of size $\Delta t = \frac{T}{m}$. Since its a Brownian motion, the increments of the Wiener process are normally distributed among other properties.

$$W_{t+\Delta t} - W_t \sim \mathcal{N}(0, (\sqrt{t_2 - t_1})^2), \qquad \Delta W_n = \xi_n \sqrt{\Delta t} \qquad \xi_n \sim \mathcal{N}(0, 1) \tag{13}$$

Therefore we can discretize our original SDE:

$$S_{t+1} = S_t + dS_t = S_t + rS_t \Delta t + \sigma S_t \xi \sqrt{\Delta t} \qquad \xi \sim \mathcal{N}(0, 1) \tag{14}$$

Likewise with the terminal prices, the MC error of Euler method is in the order of $\mathcal{O}(\frac{1}{\sqrt{n}})$. Since we are discretizing the paths, we also have its associated error in the order of $\mathcal{O}(\frac{1}{\sqrt{m}})$. Hence the total error when simulating full paths with Euler method is $\mathcal{O}(\frac{1}{\sqrt{n}}) + \mathcal{O}(\frac{1}{\sqrt{m}})$. Error can be reduced via modifying our variance reduction principles used for terminal prices or by either increasing n or m, however this may be computationally expensive. The source code for this paper uses this method by sampling $\xi_0, \xi_2, ..., \xi_m$ from $\mathcal{N}(0,1)$ then iterating from $S_{t_0}, S_{t_1}, ..., S_{t_{m-1}}$ with variance reduction.

3.3.2 Milstein's Method

Perhaps one of the most glaring issues with Euler's method is that it assumes the diffusion part of the Ito SDE to be constant (Boyle, 1976). However, in live financial markets, this is rarely the case. Milstein's method solves this by keeping one more term (the third term) in the Ito-Taylor expansion of the SDE over one time step.

$$\underbrace{Y_{t+1} = Y_t + b(Y_t, t)\Delta t + \sigma(Y_t, t)\Delta W}_{\text{Euler Method}} + \underbrace{\frac{1}{2}\sigma(Y_t, t)\partial_y \sigma(Y_t, t)(\Delta W^2 - \Delta t)}_{\text{Ito-Taylor Expansion 3rd Term}} \tag{15}$$

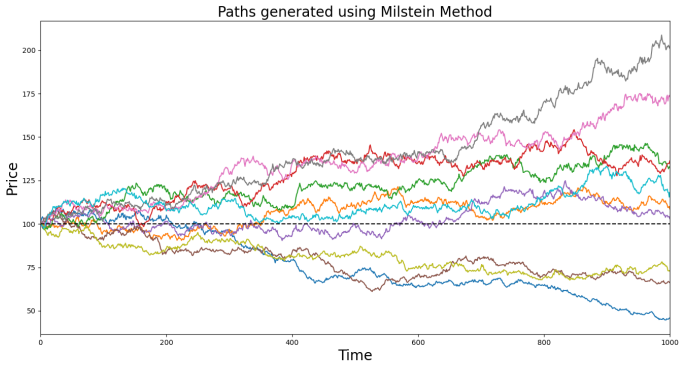

Figure 7: 10 paths generated with Milstein method. The asset has initial price 100, volitity 0.25, risk-free rate 0.07 and time to expiry 1.

By retaining the third term, Milstein's method is able to reduce error associated with discretization to the order of $\mathcal{O}(\frac{1}{m})$. Therefore, the error of Milstein's method is $\mathcal{O}(\frac{1}{\sqrt{n}}) + \mathcal{O}(\frac{1}{m})$.

3.4 Quasi-Monte Carlo with Sobol

Clearly, one issue with the Monte Carlo method is that the error due to random sampling is $\mathcal{O}(\frac{1}{\sqrt{n}})$ (Joy et al., 1996). Therefore, to optimize sampling, deterministic low discrepancy points are generated for each dimension. In this way, we fill the interval much more evenly than random sampling (Joy et al., 1996). Whilst other methods exist, we choose the Sobol sequence to generate points $U \in [0,1]$ because of its interpretability compared to other sequences and its suitability for higher dimension derivatives.

Once we generate the Sobol sequence, we transform it to become normally distributed via inverse cumulative distributive function:

$$Z = \Phi^{-1}(U) \tag{16}$$

Since we are not randomly sampling Z, therefore the quasi-Monte Carlo method is deterministic and the results are repeatable. Moreover the convergence is smooth as most noise due to random sampling is removed. Unlike previous improvements, although the method generally leads to higher accuracy, technically it is not a variance reduction technique as it doesn't have variance (since it's deterministic).

3.5 Approximating Greeks

This section introduces the idea of the Greeks, which are metrics that describe a derivative. In particular, Δ measures how much the derivative's price changes when the underlying price is changed. Γ measures how much the

Δ changes as the underlying price is changed.

$$\Delta = \frac{\partial V}{\partial S} = e^{-q(T-t)}\Phi(d_1) \qquad \Gamma = \frac{\partial^2 V}{\partial S^2} = \frac{e^{-q(T-t)\Phi(d_1)}}{S\sigma\sqrt{T-t}} \tag{17}$$

We can calculate these using the same basic principles of Monte Carlo. We start by simulating the terminal distribution and pricing our option. We change the stock price by a small known amount ξ. Then using central difference approximation from Equation (24), we have:

$$\Delta \approx \frac{V(S_0+\xi) - V(S_0-\xi)}{2\xi} \qquad \Gamma \approx \frac{V(S_0-\xi) - 2V(S_0) + V(S_0+\xi)}{\xi^2} \tag{18}$$

3.6 Heston Stochastic Volatility Model

While this paper assumes constant volatility, real markets rarely behave this way. The Heston Stochastic Volatility (HSV) model treats variance as a mean-reverting stochastic process. Volatility reverts to a long-run mean θ at a rate governed by κ, while ξ controls the randomness of this variance. Unlike the asset price, which follows geometric Brownian motion, HSV models volatility as dynamic and state-dependent. Finally, v_0 is the initial variance at $t = 0$ and ρ is the correlation between variance and price of the underlying.

$$dS_t = rS_t dt + \sqrt{v_t}S_t dW_t^S, \qquad dv_t = \kappa(\theta - v_t)dt + \xi\sqrt{v_t}dW_t^v \quad \xi = \mathcal{N}(0,1) \tag{19}$$

We will later see that the stochastic model of volatility represents live markets much better.

3.7 Implied Volatility using Newton Raphson Method

Suppose you see a derivative priced at V in the market, with an underlying asset priced at S and a strike price of K. You also know the current risk-free rate r and time to expiry T. However, you want to determine the implied volatility σ of the derivative before investing. Since the market price should match the Black–Scholes price at the correct volatility, we solve $V_{\text{BS}}(\sigma) = V_{\text{mkt}}$. To find the unknown σ, we can use the Newton–Raphson method! Which iteratively updates the volatility using:

$$\sigma_{n+1} = \sigma_n - \frac{V_{\text{BS}}(\sigma_n) - V_{\text{mkt}}}{v(\sigma_n)} \tag{20}$$

Where v is the sensitivity of the option price to changes in volatility, that is, the slope of the pricing function at σ_n. We start with an initial guess σ_0, then use the formula above to compute a better estimate σ_1, and continue this process iteratively. Each step brings us closer to the implied volatility that makes the model price equal to the observed market price. The method continues until the difference is within a chosen tolerance.

4 Finite Difference Methods

4.1 Explicit Scheme

Let's move away from the Monte Carlo method and look at another method to price a derivative. Under Black-Scholes, the option price $V(t, S)$ solves the PDE:

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} = rV - rS\frac{\partial V}{\partial S} \tag{21}$$

Finite difference methods steps the solution backwards through time in small discrete intervals from a known terminal condition at T to 0 (the initial value of the option) (Brennan & Schwartz, 1978). We can approximate the derivative $\frac{dy}{dt}$ as:

$$\text{Forward Difference: } \frac{dy}{dt} \approx \frac{y(t+\Delta t) - y(t)}{\Delta t} \tag{22}$$

$$\text{Central Difference: } \frac{dy}{dt} \approx \frac{y(t+\Delta t) - y(t-\Delta t)}{2\Delta t} \tag{23}$$

$$\text{Backward Difference: } \frac{dy}{dt} \approx \frac{y(t) - y(t-\Delta t)}{\Delta t} \tag{24}$$

The explicit scheme is a type of finite difference method that uses the forward difference approximation in Equation (22) (Brennan & Schwartz, 1978). For now, let's look back to the Black-Scholes PDE described in Equation (21). By letting:

$$[c]x \coloneqq \ln \frac{S}{K}, \qquad t \coloneqq \frac{\sigma^2}{2}(T-\tau), \qquad q \coloneqq \frac{2r}{\sigma^2}, \qquad u(t,x) \coloneqq \frac{1}{K}V\left(T - \frac{2t}{\sigma^2}, Ke^x\right) e^{\frac{1}{2}(q-1)x + \frac{1}{4}(q+1)^2 t} \tag{25}$$

$$\therefore u_t = u_{xx}, \qquad u(0,x) = \left(e^{\frac{q+1}{2}x} - e^{\frac{q-1}{2}x}\right)^+, \qquad t > 0, x \in \mathbb{R} \tag{26}$$

We can simplify Equation (26) by using central difference approximation on u_{xx} and forward difference approximation on u_t.

$$u_n^{m+1} = \alpha u_{n+1}^m + (1-2\alpha)u_n^m + \alpha u_{n-1}^m, \quad \alpha = \frac{\Delta\tau}{(\Delta x)^2} \tag{27}$$

Where m is the time index and n is our space. To make it clearer, imagine our asset path drawn on a grid with the total number of time indices as the columns and the total number of space indices as the rows. Therefore u_n^m denotes the u at time index m and space index n (Wu & Kwok, 1997).

4.1.1 Stability and Efficiency

Keen-eyed readers may note that in Equation (27) u_n^{m+1} can be thought of as a random variable where α denotes its probability of moving to either time step $n+1$ or $n-1$ and $1-2\alpha$ is the probability of it remaining as is. Thus it makes sense to assume that $1-2\alpha \geq 0$ or $0 < \alpha \leq \frac{1}{2}$. In fact this is one of the core conditions for the stability of the results obtained from the explicit method. We can see that the results are quite disastrous when this stability condition isn't adhered.

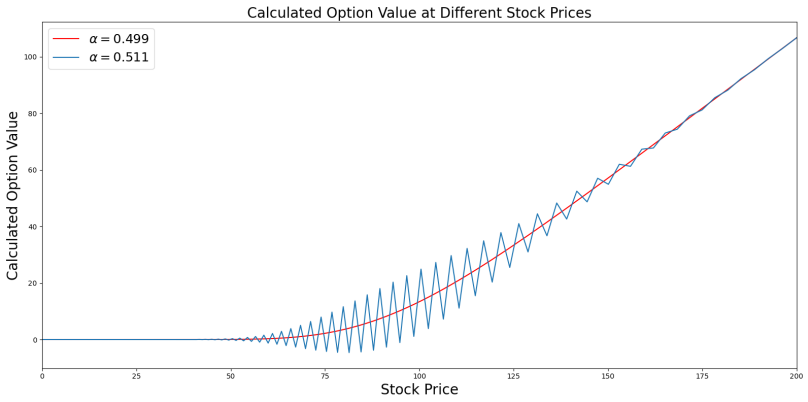

Figure 8: Showing the calculated option value at different stock prices S. Note that when $\alpha > \frac{1}{2}$, the calculated value blows up at around the strike. $\alpha = 0.511$ was chosen for better interpretability because of scale.

If we get the Taylor expansion of forward, backward and central difference approximations, we can see that the error associated with forward and backward differences is of first order $\mathcal{O}(\Delta t)$ whilst the error of central difference approximation is of second order $\mathcal{O}(\Delta x^2)$. Therefore the total error associated with our finite difference method is $\mathcal{O}(\Delta t) + \mathcal{O}(\Delta x^2)$

Each explicit step updates all N space nodes, so the cost per step is $\mathcal{O}(N)$ and the total cost of M steps is $\mathcal{O}(MN)$. Since to get u_n^{m+1} we only rely on previous known values, we don't need to solve linear equation systems (unlike other methods we will see later), so at first glance, this may seem quite efficient! However, the stability condition $\alpha = \frac{\Delta t}{\Delta x^2} \leq \frac{1}{2}$ forces $\Delta t \propto \Delta x^2$, hence $M \propto \frac{1}{\Delta x^2}$ and $N \propto \frac{1}{\Delta x}$, hence total operations scales like $\mathcal{O}(N^3)$ in 1D. Therefore when reducing error for the explicit scheme, even though the cost per step is cheap, the total runtime grows quickly.

4.2 Other Finite Difference Methods

4.2.1 Implicit Scheme

Perhaps one of the biggest drawbacks of the explicit scheme is the stability restriction, so what can we do about it? Well, instead of using the forward difference approximation, we now try using backward difference. This is the general idea behind the implicit scheme. It happens that the implicit scheme is always stable for all α. We will explore why in the following subsection. Similar to explicit scheme we start at the heat-form of the Black-Scholes PDE, but we use backward approximation for u_t:

$$u_t = u_{xx}, \qquad \therefore -\alpha u_{n-1}^{m+1} + (1+2\alpha)u_n^{m+1} - \alpha u_{n+1}^{m+1} = u_n^m, \qquad t > 0, x \in \mathbb{R} \tag{28}$$

Since u^{m+1} is unknown, we cannot get u^m directly and must solve for it implicitly. We can express our linear equation system as the following vector equation:

$$\underbrace{\begin{bmatrix} u_1^{m-1} \\ u_2^{m-1} \\ \vdots \\ u_{n-2}^{m-1} \\ u_{n-1}^{m-1} \end{bmatrix}}_{\overrightarrow{u_{m-1}}} = \underbrace{\begin{bmatrix} 1+2\alpha & -\alpha & 0 & \cdots & 0 & 0 & 0 \\ -\alpha & 1+2\alpha & -\alpha & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -\alpha & 1+2\alpha & -\alpha \\ 0 & 0 & 0 & \cdots & 0 & -\alpha & 1+2\alpha \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} u_1^m \\ u_2^m \\ \vdots \\ u_{n-2}^m \\ u_{n-1}^m \end{bmatrix}}_{\overrightarrow{u_m}} + \underbrace{\begin{bmatrix} -\alpha u_0^m \\ 0 \\ \vdots \\ 0 \\ -\alpha u_n^m \end{bmatrix}}_{\overrightarrow{b_m}} \tag{29}$$

$$\therefore \overrightarrow{u_m} = \mathbf{A}^{-1}(\overrightarrow{u_{m-1}} + \overrightarrow{b_m}) \tag{30}$$

Efficiency can be improved by adopting a tridiagonal Gaussian elimination algorithm. We first transform the matrix $\mathbf{A}$ into an upper triangular matrix by a forward sweep to express each row with just 2 variables. We can now solve using backwards substitution.

In this way, we can compute in $\mathcal{O}(n)$ time for each timestep, hence total work is $\mathcal{O}(MN)$. Like the explicit scheme, the implicit scheme has first order error associated with Δt and second order error associated with Δx.

4.2.2 Crank-Nicolson Method

The Crank-Nicolson (CN) method averages the stencil forms of the implicit and explicit schemes, essentially using the trapezoidal rule between time levels m and $m+1$ (Giles & Carter, 2006). We simply add our explicit and implicit scheme equations and divide by 2.

$$u_n^{m+1} - u_n^m = \frac{\alpha}{2}u_{n+1}^m + (1-\alpha)u_n^m + \frac{\alpha}{2}u_{n-1}^m \tag{31}$$

We can express this simply as the following vector equation:

$$\underbrace{\begin{bmatrix} 1+\alpha & -\frac{\alpha}{2} & \cdots \\ -\frac{\alpha}{2} & 1+\alpha & \cdots \\ \vdots & \vdots & \ddots \\ 0 & 0 & \cdots \\ 0 & 0 & \cdots \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} u_1^{m+1} \\ u_2^{m+1} \\ \vdots \\ u_{n-2}^{m+1} \\ u_{n-1}^{m+1} \end{bmatrix}}_{\overrightarrow{u_{m+1}}} - \underbrace{\begin{bmatrix} \frac{\alpha}{2}u_0^{m+1} \\ 0 \\ \vdots \\ 0 \\ \frac{\alpha}{2}u_n^{m+1} \end{bmatrix}}_{\overrightarrow{b_{m+1}}} = \underbrace{\begin{bmatrix} 1+\alpha & \frac{\alpha}{2} & \cdots \\ \frac{\alpha}{2} & 1+\alpha & \cdots \\ \vdots & \vdots & \ddots \\ 0 & 0 & \cdots \\ 0 & 0 & \cdots \end{bmatrix}}_{\mathbf{B}} \underbrace{\begin{bmatrix} u_1^m \\ u_2^m \\ \vdots \\ u_{n-2}^m \\ u_{n-1}^m \end{bmatrix}}_{\overrightarrow{u_m}} - \underbrace{\begin{bmatrix} \frac{\alpha}{2}u_0^m \\ 0 \\ \vdots \\ 0 \\ \frac{\alpha}{2}u_n^m \end{bmatrix}}_{\overrightarrow{b_m}} \tag{32}$$

$$\therefore \overrightarrow{u_{m+1}} = \mathbf{A}^{-1}(\mathbf{B} \times \overrightarrow{u_m} + \overrightarrow{b_{m+1}} + \overrightarrow{b_m}) \tag{33}$$

Since each step takes $\mathcal{O}(N)$ operations, therefore the total operations per timestep still follows $\mathcal{O}(N)$. Therefore total work across all time steps is $\mathcal{O}(MN)$.

Similar to the implicit scheme, the CN method is also unconditionally stable. We can show this by expressing our internal node vector equation where $\mathbf{G}$ is a symmetric tri-diagonal matrix with 2 on the diagonal and -1 on the sub- and super-diagonals. By diagonalizing $\mathbf{G}$ we can show that:

$$g_k = \frac{1 - \frac{\alpha}{2}\mu_k}{1 + \frac{\alpha}{2}\mu_k}, \qquad \mu_k = 4\sin^2\left(\frac{k\pi}{2N}\right), \quad k = 1, \ldots, N-1 \tag{34}$$

Since $\mu_k \geq 0$, hence $-1 \leq g_k \leq 1$. Hence the spectral radius $\rho(\mathbf{A}^{-1}\mathbf{B}) = \max|g_k| \leq 1$. Since error e is given by $e_{m+1} = \mathbf{A}^{-1}\mathbf{B}e_m$, by definition all error modes decay and hence the method is unconditionally stable.

Similar to explicit and implicit scheme, the CN method achieves second order accuracy in space of $\mathcal{O}(\Delta x^2)$. However, since time discretization is done using trapezoidal rule, the error in time is actually $\mathcal{O}(\Delta t^2)$. Therefore the total error is $\mathcal{O}(\Delta t^2) + \mathcal{O}(\Delta x^2)$. This is a remarkable improvement over our previous explicit and implicit schemes. Even though the order of work per step is the same for all methods, the CN method may be more computationally expensive per step due to the more complex RHS of the equation. However, it can achieve higher accuracy with larger Δt, and therefore can be computationally cheaper when higher accuracies are required.

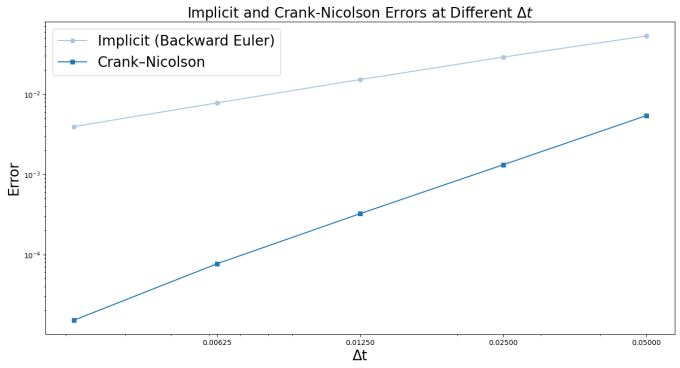

Figure 9: RMS error for implicit and Crank-Nicolson methods at different Δt. Y axis has been scaled for interpretability.

5 Comparision between Methods

Whilst Monte Carlo methods aim to simulate a large number of random price paths of the underlying in order to estimate the discounted average payoff to price a derivative, finite difference methods price options by directly discretizing and solving the Black-Scholes PDE to find the valuation of a derivative (Cont & Voltchkova, 2005).

In general, finite difference methods suffer from the curse of dimensionality (Fadugba1 & Nwozo, 2014), therefore the method is able to price at high accuracy and fast convergence for options with 1 or 2 dimensions. However the method struggles when pricing higher dimension options (such as multiple asset derivatives like exotic basket options) and path dependent options (such as asian and barrier options) (Fadugba1 and Nwozo, 2014).

$$d = 1, N = 100, \qquad d = 2, N = 100^2 \qquad d = 3, N = 100^3 \ldots$$

As the number of dimensions d increases, we need exponentially more time steps N to maintain the same grid. Since finite difference methods have faster convergence, when pricing low dimension options such as simple European options, CN methods are usually preferred. However, when pricing multiple asset options, a Monte Carlo approach may be more accurate.

Another consideration is that, when pricing early exercise options (such as the American option), Monte Carlo accuracy decreases as the optimal exercise strategy must be approximated (Fu et al., 1999). Hence for low dimensional American options, finite difference methods are preferred.

6 Numerical Examples

This section is dedicated to discussing the results ***only****. Explanations, comparisons and graphs of results can be found the method's respective sections.*

6.1 European Call

6.1.1 Using MC Terminal Distribution

S_0	K	BS Exact	Basic MC	Anti. MC	I.Sampling MC	CV MC	QMC
70	130	0.1131	0.05913	0.1451	0.1199	0.07281	0.1051
75	125	0.3636	0.2497	0.3698	0.3836	0.2991	0.3665
100	100	13.36	12.11	12.97	13.26	13.08	13.36
125	75	55.15	53.45	54.96	53.45	55.19	55.17
150	50	103.38	101.27	103.14	101.27	103.38	103.38

Basic MC	0.5574
Anti. MC	0.2998
Impor. MC	0.2219
CV MC	0.2087
Quasi. MC	NA

Table 1: Left: European Call prices under different MC methods. Right: Standard errors for $S_0 = 100$, $K = 100$.

As expected, without any variance reduction method, the basic MC has the highest error rate. Notice that variance reduction using control variates has the lowest standard error. Since we are using simulated data, and taking $h(x)$ as the discounted price of the underlying, the correlation of $h(x)$ with $f(x)$ is extremely high. Therefore, variance reduction is maximised.

Let's now look at variance reduction through importance sampling. We can see that as $S_0 < K$, our price becomes much more accurate. However, at $S_0 > K$, there is no noticeable difference compared to basic MC method. Since when $S_0 < K$, more observations are OTM, importance sampling is able to shift distribution to concentrate on ITM situations. However, when most observations are already ITM, there is not much to be improved by shifting the distribution. Finally, note that error is not meaningful for QMC as it is deterministic sampling and hence doesn't have variability.

6.1.2 Using MC Random Paths

S_0	K	BS Exact	Euler	Milstein
75	125	0.3636	0.3626	0.3822
100	100	13.36	13.59	13.35
125	75	55.15	55.33	55.48
150	50	103.38	104.03	103.31

Table 2: Calculated Prices for European Call with initial price S_0, Strike K, risk-free rate 0.07, volatility 0.25, and time to expiry of 1. Each method has been run for 10000 paths and 1000 time steps.

In theory, Milstein's method offers higher path accuracy. However, we can see that there is very little difference between both methods. Why is this? Since we are looking at only the terminal price distribution, and Milstein's method improves primarily path discretion accuracy, hence the improvements are quite marginal (Fournié et al., 1999).

6.1.3 Calculating Δ and Γ

When we increase S_0 and the option becomes deep ITM, Δ increases as the call behaves more similarly to the underlying. Γ decreases as it peaks ATM and Δ change slows down as it approaches 1. When we increase σ, even though the upside for calls is higher, the probability of ITM is the same, hence Δ is the same. However, Γ decreases since the higher volatility means the stock is less sensitive to small moves. When we increase r, since option more likely to be ITM, Δ increases. Unless r increase is substantial, Γ changes very little since the peak will still be around the ATM point. When T increases, since there is more time to be ITM, Δ increases. However, as the Δ curve becomes flatter, Γ decreases.

S_0	K	σ	r	T	Δ	Γ
100	100	0.25	0.07	1	0.6572	0.01478
125	100	0.25	0.07	1	0.9028	0.005495
100	100	0.50	0.07	1	0.6518	0.007409
100	100	0.25	0.10	1	0.7002	0.01387
100	100	0.25	0.07	2	0.7168	0.009614

Table 3: Δ and Γ estimates using Monte Carlo method with antithetical variable variance reduction over 10000 simulations. Small change $\xi = 0.1$

6.1.4 Heston Stochastic Volatility Model

κ	θ	v_0	ξ	ρ	with HSVM
3	0.0625	0.0625	0.25	0	13.41
4	0.0625	0.0625	0.25	0	13.34
3	0.10	0.0625	0.25	0	14.82
3	0.0625	0.10	0.25	0	14.09
3	0.0625	0.0625	0.50	0	13.14
3	0.0625	0.0625	0.25	0.25	13.21

Table 4: European Call price under HSV model for $S_0 = 100$ and $K = 100$.

Increasing θ raises the long-run average variance, leading to higher expected volatility over the life of the option. Since call options benefit from both upside and increased volatility, the price of the option increases. The same reasoning applies to increasing the initial variance v_0, which boosts short-term volatility. While other parameters like κ, ρ, and ξ shape the dynamics of variance, their impact on option price is typically smaller compared to θ and v_0, as variance affects both in-the-money and out-of-the-money options.

6.1.5 Calculating Implied Volatility

V	S_0	K	σ
10	100	100	0.1565
10	100	125	0.3971
10	100	75	NA
20	100	100	0.4289

Table 5: Volatility estimates to $\epsilon = 0.00000001$ using Newton Raphson Method.

As K increases, the option becomes more OTM. To maintain same V, we need higher σ (since σ increases call price). When we decrease K, the options becomes worth more. Therefore, the same V cannot exist in an arbitrage-free market. Therefore no σ exists. When we increase V, when it's ATM, σ must increase in order to cause this increased value.

6.2 Barrier Options

6.2.1 Using MC Random Paths

S_0	K	B	Euler	Milstein
100	90	110	19.41	18.71
100	95	105	15.17	17.34
100	105	95	5.596	5.226
100	110	90	1.843	1.896

(a) Down-and-In

S_0	K	B	Euler	Milstein
100	90	110	0.4177	0.4575
100	95	105	0.01623	0.02723
100	105	95	0	0
100	110	90	0	0

(b) Up-and-Out

S_0	K	B	Euler	Milstein
100	90	110	18.77	17.62
100	95	105	16.55	16.38
100	105	95	11.48	10.74
100	110	90	8.863	8.413

(c) Up-and-In

Table 6: Barrier option prices using Euler and Milstein schemes for different types: Down-and-In, Up-and-Out, and Up-and-In. Here S_0 is the initial stock price, K is the strike, and B is the barrier level.

Down and in barrier options are a type of exotic option that can only be exercised if at any point before expiry the option drops below the boundary price B. It should be obvious then that if $B \geq S_0$, the option behaves as a standard European call since the boundary condition has been met (Boyle & Tian, 1998). However, if for $B \leq S_0$, the option must hit the barrier before returning above the strike K to make a profit, making the option's value much cheaper. As we increase the spread between K and B, the option becomes cheaper (Zvan et al., 2000).

Up and Out barrier options can only be exercised if at no point before expiry does the price of the underlying exceed the boundary price. Therefore, for values when $S_0 \geq B$, the asset is never exercisable and has value 0. However, for $S_0 < B$, the option decreases in price the closer B is, since more it is more likely for option to exceed the boundary.

Up and In barrier options can only be exercised if at any point before expiry the underlying exceeds the boundary price. Therefore, for greater B, the option is cheaper. Note that since we are also making K smaller, the difference is not as stark here as the high B is balanced by the fact that a low K means we profit more for paths that are exercisable. Moreover, when $B < S_0$, the option behaves as a regular European call since it is always exercisable.

6.2.2 Using Crank-Nicolson Scheme

S_0	K	B	CN Approx.
100	105	95	3.196
100	90	95	15.71
100	90	85	16.08
100	90	80	16.08

Table 7: Down and Out option prices using Crank-Nicolson scheme.

The most important difference when implementing the CN scheme to barrier options is that we must take care to set our Dirichlet boundaries correctly. For the down and out option, we take space indices only from B upwards. Our left boundary should be 0 as the option is dead once hitting the boundary condition and it should be priced as a standard European call on the right boundary as the probability of dropping below the boundary approaches 0. We can see that decreasing our B increases the price of our option as the underlying has a smaller chance to cross the barrier. However, if too far below our strike K, it becomes more negligible and the payoff approaches that of a European call.

6.3 American Options

We explore the American Put as the American call in our model is never optimally exercised early.

S_0	K	MC Approx.
70	130	0.04941
75	125	0.08938
100	100	7.597
125	75	45.40
150	50	97.23

Table 8: American Puts with volatility 0.25, riskfree rate 0.07, time to expiry 1. We price using Monte Carlo with Least Squares regression to estimate the early exercise boundary.

The main difference with American options is that we need an early exercise curve. How do we adapt this to the Monte Carlo method before? What we can do is that at every point, we estimate the continuation value or the expected discounted future cash flow if we do not exercise at this point in time (Longstaff & Schwartz, 2001). We can calculate this with a polynomial regression with the underlying S. We then compare the intrinsic value of the derivative (exercising now) and the continuation value (exercising later). If intrinsic value is greater, we exercise now, and we do not exercise if it is smaller (Longstaff & Schwartz, 2001).

6.4 Yield Curve Swap Solver

A swap is when two parties agree to swap a floating interest rate for a fixed interest rate either for profit or to avoid risk. The fixed interest rate is called the swap rate. Therefore $PV_{\text{fixed}} = PV_{\text{float}}$. We can solve for swap rate using Newton Raphson, working the same way as when we derived implied volatility.

$$D_k^{n+1} = D_k^n - \frac{f_k(D_k^n)}{S_k \alpha_k + 1} \tag{35}$$

Maturity	Swap Rate	Discount	Zero Rate
1	3.64%	0.9648	3.64%
2	3.58%	0.9305	3.64%
3	3.62%	0.8956	3.62%
4	3.68%	0.8597	3.66%
5	3.76%	0.8229	3.77%
6	3.85%	0.7848	3.89%
7	3.94%	0.7456	3.94%

Table 9: AUD Par Swap Rates, Bootstrapped Discount Factors, and Implied Zero Rates

As maturity increases, swap rate and zero rate increases because of time value of money. Moreover, discounted money decreases for the same reason as zero rates are increasing.

7 Conclusion

In this report, we explored and implemented two core numerical approaches for pricing financial derivatives: Monte Carlo simulation and finite difference methods. Through a range of examples including European, American, and barrier options, we demonstrated how each method performs under different conditions, with Monte Carlo offering flexibility for complex, path-dependent structures, and finite difference methods providing faster convergence in low-dimensional settings. Variance reduction techniques—such as antithetic variables, importance sampling, and control variates—significantly improved the efficiency of Monte Carlo simulations. Overall, the methods developed offer practical, accurate, and adaptable tools for modern derivative pricing and risk management.

References

Boyle, P. P. (1976). Options: A monte carlo approach. *Journal of Financial Economics*, *4*(1), 324–348.

Boyle, P. P., Broadie, M., & Glasserman, P. (1997). Monte carlo methods for security pricing. *Journal of Economic Dynamics and Control*, *21*(8-9), 1267–1321.

Boyle, P. P., & Tian, Y. S. (1998). An explicit finite difference approach to the pricing of barrier options. *Applied Mathematical Finance*, *5*(1), 17–52.

Brennan, M. J., & Schwartz, E. S. (1978). Finite difference methods and jump processes arising in the pricing of contingent claims: A synthesis. *Journal of Financial and Quantitative Analysis*, *13*(3), 461–474.

Broadie, M., & Glasserman, P. (1996). Estimating security price derivatives using simulation. *Management Science*, *42*(2), 269–285.

Carriere, J. (1996). Valuation of the early-exercise price for options using simulations and nonparametric regression. *Insurance: Mathematics and Economics*, *19*(1), 19–30.

Cont, R., & Voltchkova, E. (2005). A finite difference scheme for option pricing in jump diffusion and exponential lévy models. *SIAM Journal on Numerical Analysis*, *43*(4), 1596–1626.

Duffie, D., & Glynn, P. (1995). Efficient monte carlo simulation of security prices. *Annals of Applied Probability*, *5*(4), 897–905.

Fadugba1, S. E., & Nwozo, C. R. (2014). On the comparative study of some numerical methods for vanilla option valuation. *Communications in Applied Sciences*, *2*(1), 65–84.

Fournié, E., Lasry, J.-M., Lebuchoux, J., Lions, P.-L., & Touzi, N. (1999). Applications of malliavin calculus to monte carlo methods in finance. *Finance and Stochastics*, *3*(4), 391–412.

Fu, M. C., Laprise, S. B., Madan, D. B., Su, Y., & Wu, R. (1999). Pricing american options: A comparison of monte carlo simulation approaches.

Giles, M. B., & Carter, R. (2006). Convergence analysis of crank–nicolson and rannacher time-marching. *Journal of Computational Finance*, *9*(4), 89–112.

Joy, C., Boyle, P. P., & Tan, K. S. (1996). Quasi-monte carlo methods in numerical finance. *Management Science*, *42*(6), 926–938.

Longstaff, F. A., & Schwartz, E. S. (2001). Valuing american options by simulation: A simple least-squares approach. *Review of Financial Studies*, *14*(1), 113–147.

Paskov, S., & Traub, J. F. (1995). Faster valuation of financial derivatives. *Journal of Portfolio Management*, *22*(1), 113–120.

Tilley, J. A. (1993). Valuing american options in a path simulation model. *Transactions of the Society of Actuaries*, *45*, 83–104.

Wu, L., & Kwok, Y.-K. (1997). A front-fixing finite difference method for the valuation of american options. *Journal of Financial Engineering*, *6*(1), 83–100.

Zvan, R., Vetzal, K. R., & Forsyth, P. A. (2000). Pde methods for pricing barrier options. *Journal of Economic Dynamics and Control*, *24*(11-12), 1563–1590.