Methods for Option Pricing

Analysis and implementation of continuous time models

Piotr Kurek

30.10.2023

Table of Contents

1. Introduction 3
2. Option pricing in discrete time 4
2.1. Binomial model 4
3. Option pricing in continuous time 8
3.1. Monte Carlo 8
3.1.1. Model 8
3.1.2. Implementation 10
3.1.3. Importance sampling 12
3.2. Black-Scholes 13
3.2.1. Brownian motion 13
3.2.2. Itô’s Lemma 14
3.2.3. Derivation of the Black-Scholes model 15
3.2.4. Solution to diffusion equation 17
3.2.5. Application to option pricing 19
3.2.6. Implementation 21
4. Finite Difference Methods for solving PDEs 22
4.1. Explicit method 23
4.2. Implicit method 25
4.3. Crank-Nicolson method 26
5. Results 29
5.1. Monte Carlo 29
5.2. Black-Scholes 32
5.3. Comparison 34
6. Summary 35
7. Bibliography 36
List of figures 36
List of graphs 36
List of tables 36

1. Introduction

An option is a financial contract that gives the holder a right but not an obligation to buy or sell an underlying financial instrument at a specific strike price at the expiration or before, depending on the option style. This paper will analyse different methods for option pricing. First, we start with the discrete time binomial model to grasp the basic idea of the pricing process. Then, we move to a mathematically more complex continuous time where the option price will be defined as a Geometric Brownian motion. We will use two different techniques to calculate the value of the option contract – Monte Carlo and Black-Scholes models. We then implement those two models in the R language so that we can analyse and compare the results. Finally, this paper will also give a theoretical background on the numerical methods for solving partial differential equations – Finite Difference Methods. The main goal of this project is to summarise the author's knowledge on option pricing that he gained during the MIT MicroMasters course "Mathematical Methods for Quantitative Finance" and "Financial Mathematics" course at King's College London in order to prepare for the Master's course in the field of Mathematics and Finance.

2. Option pricing in discrete time

2.1. Binomial model

To better understand the process of option pricing, let us first analyse options in the discrete time Binomial model. Consider a simple market for a single stock where the stock value at time 0 is \$100 and at time 1 is \$120 with probability p or \$90 with probability $1 - p$.

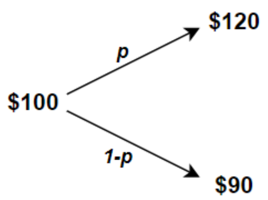


Figure 1: Simple stock model

Assume you sign a contract at time 0 for an option with a strike price of \$110 and expiration at time 1. The value of the contract depends on the option type. A vanilla European call option, a financial contract that gives the holder of the contract a right, but not an obligation, to buy an underlying asset for a value specified by the strike price at the expiration date, will have a value C_0 at time 0 and either \$10 or \$0 value at the time 1.

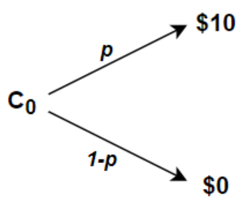


Figure 2: Call payoff model

On the other hand, the Vanilla European put option, which is a financial contract that gives the holder of the contract a right, but not an obligation, to sell an underlying asset for a value of a strike price at the expiration date, will start with value P_0 at time 0 and at time 1 will give a payoff of \$20 or \$0 as shown on *Figure 3*.

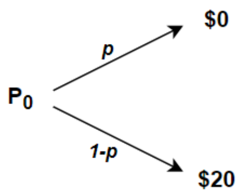


Figure 3: Put option payoff model

The payoffs of call and put options with a strike equal K are equal to:

$$call\ option: (S - K)_+ \quad [1]$$

$$put\ option: (K - S)_+ \quad [2]$$

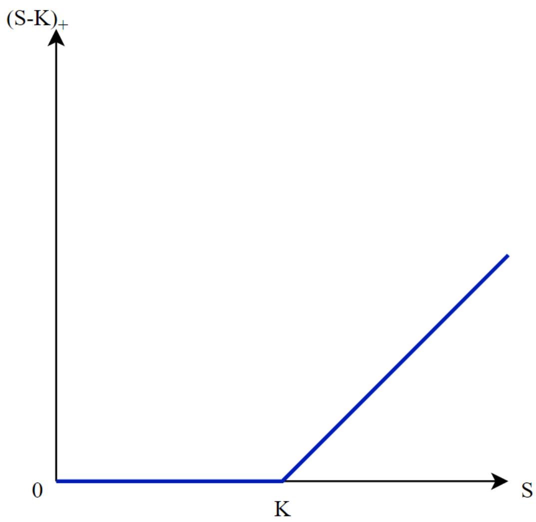


Figure 4: Call option payoff function

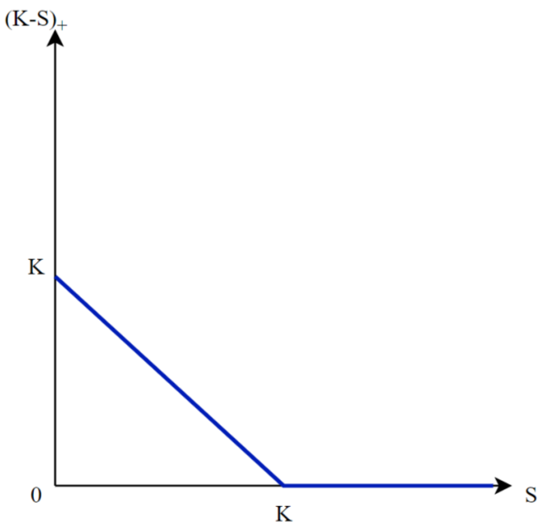


Figure 5: Put option payoff function

Now, consider a market with a single asset. At time 0, there is a known positive quantity S_0 that denotes an asset price. At time 1, the asset value will take either one of two positive values $S_1(A) = uS_0$ or $S_1(B) = dS_0$ where u and d represent up and down factors of the asset price:

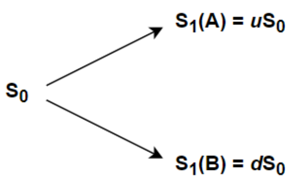


Figure 6: General one-period binomial model

In our model, we also use an interest rate r that returns $1 + r$ dollars in single time period. To avoid arbitrage, we have to assume that:

$$0 < d < 1 + r < u \qquad [3]$$

The goal of option pricing is to determine how much is the price V_0 of an option worth at time 0. To do that, we will replicate the option payoffs $V_1(A)$ and $V_1(B)$ with a portfolio consisting of stock and money markets with a no-arbitrage pricing model, with the following assumptions:

- We can buy or sell a fraction of the stock
- Interest rate is equal for lending and borrowing
- The bid-ask spread is 0
- The stock price follows a binomial tree structure, meaning stock value at any time can only take one of two possible values in the next time step

We start with a portfolio with a value C_0 at time 0 and buy Δ_0 shares of stock, changing the cash amount to $C_0 - \Delta_0 S_0$. At time 1 the value of the portfolio will be:

$$C_0 = \Delta_0 S_1 + (1 + r)(C_0 - \Delta_0 S_0) = (1 + r)C_0 + \Delta_0(S_1 - (1 + r)S_0) \qquad [4]$$

Remember, we assume that the value of the portfolio C at time 1 should be equal to the value of the derivative payoff V. Hence we have $C_1(A) = V_1(A)$ and $C_1(B) = V_1(B)$, so that the value of our initial portfolio C_0 will be equal to the replicated payoff V_0. The values of the payoffs at the time 1 are know, however, we do not know which one will be realized. Divide both sides of the [4] by $(1 + r)$ to get:

$$C_0 + \Delta_0 \left(\frac{1}{1+r} S_1(A) - S_0\right) = \frac{1}{1+r} V_1(A) \qquad [5]$$

$$C_0 + \Delta_0 \left(\frac{1}{1+r} S_1(B) - S_0\right) = \frac{1}{1+r} V_1(B) \qquad [6]$$

Multiply values of payoffs by risk-neutral probabilities p and $q = 1 - p$ to solve for two unknowns C_0 and Δ_0.

$$C_0 + \Delta_0 \left(\frac{1}{1+r} [pS_1(A) + qS_1(B)] - S_0\right) = \frac{1}{1+r} [pV_1(A) + qV_1(B)] \qquad [7]$$

We can make a delta-hedging term Δ_0 disappear by choosing the appropriate p so that:

$$S_0 = \frac{1}{1+r}[pS_1(A) + qS_1(B)] \quad [8]$$

Then:

$$C_0 = \frac{1}{1+r}[pV_1(A) + qV_1(B)] \quad [9]$$

Therefore, the option that at the time 1 has a payoff V_1, at time 0 should be priced:

$$V_0 = \frac{1}{1+r}[pV_1(A) + qV_1(B)] \quad [10]$$

Calculate p and q values:

$$S_0 = \frac{1}{1+r}[puS_0 + (1-p)dS_0] \quad [11]$$

$$p = \frac{1+r-d}{u-d}\ , \quad q = \frac{u-1-r}{u-d} \quad [12]$$

Subtract [6] from [5] to get delta-hedging formula for:

$$\Delta_0 = \frac{V_1(A)-V_1(B)}{S_1(A)-S_1(B)} \quad [13]$$

This is a process of hedging a short position in the option, which price should be equal to V_0 at time 0 under risk-natural probabilities in a Binomial model for a portfolio consisting of stock and cash.

3. Option pricing in continuous time

We now move from option pricing in the discrete time and build on the principles of the Binomial model to derive Monte Carlo and Black-Scholes models that are used to price options in continuous time.

3.1. Monte Carlo

3.1.1. Model

In this method, we will use a Monte Carlo model as a random number generator to simulate the stochastic process of the option price. We then approximate the exact results of the payoffs at expiration time T by computing the average over all simulation paths. First, for our model to be risk-neutral, we have to find a Q-measure under which the expected return of the asset at the present time t will be equal to the risk-free rate.

$$E_t^Q\left[\frac{dS_t}{S_t}\right] = rdt \qquad [14]$$

Consider an Itô process X that is a martingale if and only if its drift coefficient μ is 0:

$$dX_t = \mu dt + \sigma dB_t \qquad [15]$$

$$E_t[X_{t'}] = X_t, \qquad E_t[dX_t] = 0, \qquad for \quad t < t' \qquad [16]$$

For a discounted price process F of an asset S that is also itself an Itô process:

$$F = e^{-rt}S \quad where \quad dS = \mu S dt + \sigma S dB \qquad [17]$$

Then, using Itô lemma:

$$\frac{\partial F}{\partial S} = e^{-rt}, \quad \frac{\partial^2 F}{\partial S^2} = 0, \quad \frac{\partial F}{\partial t} = -re^{-rt}S \qquad [18]$$

$$\frac{\partial F}{F} = (\mu - r)dt + \sigma dB \qquad [19]$$

Hence $\frac{\partial F}{F}$ is a martingale if and only if $\mu = r$. Knowing this, we can find a Q-measure for an asset S:

$$\frac{dS_t}{S} = rdt + (\mu - r)dt + \sigma dB = rdt + \sigma dB^Q \quad where \quad dB^Q = \left(\frac{\mu - r}{\sigma}\right)dt + \sigma dB \qquad [20]$$

The stochastic term dB^Q is a martingale with expectation at time t equal to 0.

$$E_t^Q[dB^Q] = 0 \qquad [21]$$

Therefore, in this model, all the tradable assets S will have a discounted payoff V that is a martingale under the risk-neutral Q-measure.

$$e^{-rt}V_t = E_t^Q[e^{-rT}V_T] \qquad [22]$$

Now, we have to model a price process S of an asset that results in a payoff V_T at contract expiration time T. Remember that S is a geometric Brownian motion, which is log-normally distributed with constant drift μ and volatility σ.

$$dS_t = \mu S_t dt + \sigma S_t dB_t \qquad [23]$$

By applying the Itô lemma to [23], we obtain a change in logarithmically compounded terminal price S_T with unchanged volatility and drift shifted by $-\frac{\sigma^2}{2}$ that comes from an Itô term.

$$d(\log S_T) = \left(\mu - \frac{\sigma^2}{2}\right)dt + \sigma dB_T \qquad [24]$$

$$\log\frac{S_T}{S_0} \sim N\left(\left(u - \frac{\sigma^2}{2}\right)T,\ \sigma^2 T\right) \qquad [25]$$

$$S_T = S_0 e^{\left(\mu - \frac{\sigma^2}{2}\right)dt + \sigma(B_T - B_0)} \qquad [26]$$

The $B_T - B_0$ is a Gaussian random variable. Hence, we can replace it with the random variable z_t scaled by $\sqrt{dt}$. By replacing $\mu \to r$ to get the Q-measure, we arrive at a formula [27] for the terminal price of the asset.

$$S_T = S_0 e^{\left(r - \frac{\sigma^2}{2}\right)dt + \sigma z_t \sqrt{dt}} \qquad [27]$$

Finally, combine the equation [22] of the discounted asset payoff under measure Q with the equation [27] for the terminal asset price to get a model for the fair present values of a call and put options, respectively.

$$C_t = e^{-r(T-t)} E_t^Q[\max(S_T - K, 0)] \tag{28}$$

$$P_t = e^{-r(T-t)} E_t^Q[\max(K - S_T, 0)] \tag{29}$$

3.1.2. Implementation

Generate an ensemble of equiprobable time-dependent price paths and compute solutions as risk-neutral expectations of discounted payoffs average under the Q-measure. The asset's value at time t is a present value, in the time remaining to expiration, of the expected value of the function at expiration. The value of the function at expiration is replaced by an approximation that follows our price process S. We determine terminal values for different price paths and return discounted present average value of S_T over n simulations, that could have different realizations.

$$V_t = e^{-r(T-t)} E_t^Q[V_T] = e^{-r(T-t)} E_t^Q[V(T, S_T)] \quad where \quad S_T = S_0 e^{\left(r - \frac{\sigma^2}{2}\right)dt + \sigma z_t \sqrt{dt}} \tag{30}$$

Below is the implementation of the Monte Carlo model in R.

```r
MCPrice<-function(Np, Nt, T, r, sigma, s0, K) {


  dt = T/Nt
  Z <-  matrix(rnorm(Np * Nt, mean=0, sd=1),nrow = Nt, ncol = Np)
  dB <- Z*sqrt(dt)
  B <- matrix(numeric(Np*(Nt+1)), nrow = (Nt+1), ncol = Np)
  S <- matrix(numeric(Np*(Nt+1)), nrow = (Nt+1), ncol = Np)

  for(i in 1:Np){
    B[,i] <- c(0, cumsum(dB[,i]))
    S[,i] <- s0*exp((r - 0.5*sigma^2)*T + sigma*B[,i])
  }

  ST <- S[nrow(S),]

  # call option price
  call_payoffs <- exp(-r*T)*pmax(ST-K,0)
  call_price <- mean(call_payoffs)

  # put option price
  put_payoffs <- exp(-r*T)*pmax(K-ST,0)
  put_price <- mean(put_payoffs)

  # standard deviation
  err_call <- sd(call_payoffs)/sqrt(Np)
  err_put <- sd(put_payoffs)/sqrt(Np)

  output<-list(call_price=call_price, err_call=err_call,
               put_price=put_price, err_put=err_put)

  return(output)

}
```

Figure 7: Monte Carlo model implementation in R

The $MCPrice$ function takes as an input the following list of parameters:

- Np number of simulations
- Nt number of time steps
- T terminal time of the contract
- r risk-free rate
- $sigma$ volatility σ
- $s0$ initial price of the asset
- K strike price of the contract

First, define dt as the change in the time step of the algorithm. Then define Z to be the Np by Nt matrix populated by the Gaussian random variables, which we later multiply by the $\sqrt{dt}$ to get the stochastic variable dB. Next, iterate through all simulations and populate matrix B with a cumulative sum of Gaussian random variables and use them to create a price process by calculating the asset price for each time step in the matrix S. This gives us the price path for every simulation with a final price stored at the last row $Nt + 1$ of the matrix S, which we can store in the vector ST. The last step is to plug in the ST vector into the discounted expectation formula for the call and put options, and return the current price of the contracts along with the standard error.

3.1.3. Importance sampling

In the previous model, most paths finished out of money, below 0. Therefore, now we will optimize the algorithm to spent less time on simulations that do not contribute to the final solution. To construct a better Monte Carlo, we will use the importance sampling technique.

```r
MC_Price_ImportanceSampling<-function(Np, Nt, T, r, sigma, s0, K) {

  dt = T/Nt
  Z <-  matrix(rnorm(Np * Nt, mean=0, sd=1),nrow = Nt, ncol = Np)
  dB <- Z*sqrt(dt)
  B <- matrix(numeric(Np*(Nt+1)), nrow = (Nt+1), ncol = Np)
  S <- matrix(numeric(Np*(Nt+1)), nrow = (Nt+1), ncol = Np)

  for(i in 1:Np){
    B[,i] <- c(0, cumsum(dB[,i]))
    S[,i] <- s0*exp((r - 0.5*sigma^2)*T + sigma*B[,i])
  }

  ST <- S[nrow(S),]

  # call option price
  call_payoffs <- exp(-r*T)*pmax(ST-K,0)[ST>K]
  call_price <- mean(call_payoffs*mean(ST>K))

  # put option price
  put_payoffs <- exp(-r*T)*pmax(K-ST,0)[ST<K]
  put_price <- mean(put_payoffs*mean(ST<K))

  # standard deviation
  err_call <- sd(call_payoffs*mean(ST>K))/sqrt(Np)
  err_put <- sd(put_payoffs*mean(ST<K))/sqrt(Np)

  output<-list(call_price=call_price, err_call=err_call,
               put_price=put_price, err_put=err_put)

  return(output)
}
```

Figure 8: Monte Carlo model with importance sampling implementation code in R

The only difference between the previous Monte Carlo model and the one using importance sampling is that now, when calculating the contract payoffs, we will only consider the terminal prices that are more or less than the strike price for the call and put options, respectively. When calculating the mean payoffs we also have to multiply by the sample mean.

3.2. Black-Scholes

Option pricing in continuous time changes one of the assumptions that we had in the discrete time approach – now, the stock price follows a geometric Brownian motion that is a stochastic process allowing prices to take on any random variable. To calculate the option contract value at the present time when the contract is created, we will use a Black-Scholes model that says the option prices are log-normally distributed Gaussian random variables $X \sim N(u, \sigma^2)$ with constant drift μ, volatility σ, and probability density function:

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-u)^2}{2\sigma^2}} \quad [31]$$

A time-dependent stochastic process where $X \sim N(ut, \sigma^2 t)$ has probability density function:

$$p(x,t) = \frac{1}{\sqrt{2\pi\sigma^2 t}} e^{-\frac{(x-ut)^2}{2\sigma^2 t}} \quad [32]$$

This will be important later when solving partial differential equation by finding a solution to the diffusion equation. In the following steps, we will derive a Black-Scholes model, which formula allows us to calculate the present value of the option contract.

3.2.1. Brownian motion

The basic building block for Brownian motion is an elementary random walk with multiple steps of unit size. We can define it as a sum of many random variables z_t that have expectation 0 and variance 1.

$$B_{1,T} = \sum_{t=t_0+1}^{t_0+T} z_t \quad [33]$$

B stands for a Brownian motion, T for a path length, and 1 represents the time step. We then have:

$$E_t\left[B_{1,T}\right] = 0, \quad t \leq t_0 \quad [34]$$

$$Var_t\left(B_{1,T}\right) = T \quad [35]$$

Subdivide the time interval to be infinitesimal while preserving the distribution of terminal values by studying the scale changes for time step and step size.

$$\text{Let } \Delta t = \frac{T}{n}, \lambda = \sqrt{\Delta t} = \sqrt{\frac{T}{n}} \text{ and } \varepsilon_t = \lambda z_t \quad [36]$$

$$B_{\Delta t,T} = \sum_{t=1}^{n} \varepsilon_t = \sqrt{\Delta t} \sum_{t=1}^{n} z_t \quad [37]$$

$$E\left[B_{\Delta t,T}\right] = 0 \quad [38]$$

$$Var\left(B_{\Delta t,T}\right) = nVar(\varepsilon_t) = n\Delta t Var(z_t) = T \quad [39]$$

$$\lim_{\Delta t \to 0} B_{\Delta t,T} \sim N(0,T) \quad [40]$$

Rescaling the time step and step size simultaneously in a specific relationship resulted in the same finite terminal values – the endpoint of the Brownian motion is a Gaussian random variable with mean 0 and variance equal to the length of the time elapsed. The Brownian motion path is everywhere continuous. However, it is nowhere differentiable due to its randomness.

3.2.2. Itô's Lemma

Define an Itô process as a stochastic process of the form:

$$dX_t = adt + bdB_t \quad [41]$$

adt is a deterministic part and dB_t is a Gaussian Brownian random variable. Since X is nowhere differentiable, the usual chain rule does not hold. However, the function $F(X)$ is differentiable and we can use Taylor's theorem to expand it to identify the leading order terms in dt. We want to focus on convergence in probability and identify terms with vanishing variance as non-stochastic.

$$dF = \left(\frac{\partial F}{\partial t}dt + \frac{\partial F}{\partial X}dX + \frac{1}{2}\frac{\partial^2 F}{\partial X^2}(dX)^2 + \frac{\partial^2 F}{\partial t \partial X}dtdX + \frac{1}{2}\frac{\partial^2 F}{\partial X^2}(dt)^2 + \Theta((dt)^3, (dX)^3, \dots)\right) =$$
$$= \left(\frac{\partial F}{\partial t}dt + \frac{\partial F}{\partial X}[adt + bdB_t] + \frac{1}{2}\frac{\partial^2 F}{\partial X^2}[adt + bdB_t]^2 + \cdots\right) \quad [42]$$

We only keep terms of order dt as anything higher goes to 0 faster than dt. Variance of the higher terms vanishes to order dt so we can treat them as non-stochastic. We also know that:

$$(dX)^2 = b^2 dt \quad [43]$$

Substituting dX by $adt + bdB_t$ in [42] gives the final formula for the Itô's Lemma:

$$dF = \left(\frac{\partial F}{\partial t}dt + \frac{\partial F}{\partial X}[adt + bdB_t] + \frac{1}{2}\frac{\partial^2 F}{\partial X^2}[b^2 dt]^2\right) =$$
$$= \left[\frac{\partial F}{\partial t} + \frac{b^2}{2}\frac{\partial^2 F}{\partial X^2}\right]dt + \left[\frac{\partial F}{\partial X}\right]dX =$$
$$= \frac{\partial F}{\partial t}dt + \frac{\partial F}{\partial X}dX + \frac{b^2}{2}\frac{\partial^2 F}{\partial X^2}dt \quad [44]$$

This is a desired result for dF, which is a stochastic differential equation – the differential of a function of an Itô process is itself an Itô process. The differential has one more additional term than the chain rule that is called the Itô term, and it vanishes if function F is linear in X.

3.2.3. Derivation of the Black-Scholes model

Consider the dynamic trading strategy where we buy and hold the option position and rebalance the stock position based on the option's price and time to expiry. The stock purchases and sales flow in and out of the cash account, which earns or borrows at a risk-free rate. We also have to make a few assumptions:

- We have unlimited credit
- We can lend or borrow at a single risk-free rate r
- Stock can be traded in fractional quantities
- Full use of short sale
- The stock pays no dividends
- There are no transaction costs
- There are no trading delays or market impacts

Now consider the portfolio X with an initial value 0 that holds stocks and bonds ('cash') in quantities q and C. The value of the portfolio at the initial time is given by:

$$X_0 = q_0 S_0 + C_0 M_0 = 0 \quad [45]$$

Rebalancing the portfolio does not change the portfolio value. It simply exchanges bonds for stock of equal value at the prevailing market price. In order to rebalance the portfolio at time t:

$$X_t - X_{t-1} = S_t(q_t - q_{t-1}) + M_t(C_t - C_{t-1}) = 0 =$$
$$= S_{t-1}(q_t - q_{t-1}) + M_{t-1}(C_t - C_{t-1}) + (S_t - S_{t-1})(q_t - q_{t-1}) + (M_t - M_{t-1})(C_t - C_{t-1}) \quad [46]$$

Hence, the self-financing condition is:

$$Sdq + MdC + dSdq + dMdC = 0 \quad [47]$$

Consider a self-financing portfolio π consisting of a single option contract of value V plus dynamically rebalanced stock with bond positions.

$$Let\ V = V(t, S)\ and\ dS = (\mu S)dt + (\sigma S)dB \quad [48]$$

$$\pi = V + qS + CM \quad [49]$$

The portfolio change in value in continuous time:

$$\begin{aligned} d\pi &= dV + d(qS + CM) = \\ &= dV + (qdS + CdM) + (Sdq + MdC + dqdS + dCdM) = \\ &= dV + qdS + rCMdt = \\ &= dV + qdS + r(\pi - V - qS)dt \end{aligned} \quad [50]$$

Now, we can look for a dynamic trading strategy for $q = q(t)$ that eliminates the risk in the portfolio. The reason to expect this is possible is that there is only one stochastic driver $\left(\frac{\varphi V}{\varphi S} + q\right) dS$ in [51]. By applying the Itô's lemma to [50], we get:

$$d\pi = \left(\frac{\partial V}{\partial t} + \frac{(\sigma S)^2}{2}\frac{\partial^2 V}{\partial S^2}\right) dt + \left(\frac{\partial V}{\partial S} + q\right) dS + r(\pi - V - qS)dt \quad [51]$$

Choosing a special delta for $q(t)$ makes the coefficient of the stochastic term dS vanish, however, we cannot assume it is constant.

$$q = -\frac{\partial V}{\partial S} = -\Delta \quad [52]$$

This formula tells us how to replicate V's payoff using only stocks and bonds. With the dynamic choice of q, the portfolio is now risk-free, and hence, its growth rate is risk-free. Since the portfolio's initial value was 0, $\pi = 0$, the value must remain 0 at all times to avoid arbitrage.

$$d\pi = \left(\frac{\partial V}{\partial t} + \frac{(\sigma S)^2}{2}\frac{\partial^2 V}{\partial S^2} - rV + rS\frac{\partial V}{\partial S}\right)dt = 0 \qquad [53]$$

This leads to the Black-Scholes partial differential equation, where σ is volatility and r is a risk-free rate.

$$\frac{\partial V}{\partial t} + \frac{(\sigma S)^2}{2}\frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0 \qquad [54]$$

3.2.4. Solution to diffusion equation

The Black-Scholes equation is closely related to the diffusion equation [55] – a well-known partial differential equation that describes the macroscopic behaviour of the particles in Brownian motion sometimes referred to as a heat equation for computing the heat diffusion process in physics.

$$\frac{\partial p_0}{\partial t} = \frac{1}{2}\frac{\partial^2 p_0}{\partial z^2} \qquad [55]$$

The solution to diffusion equation is Gaussian:

$$p_0(z,t) = \frac{1}{\sqrt{2\pi t}}e^{\frac{-z^2}{2t}} \qquad [56]$$

The Gaussian solution can be used to price derivatives with any terminal payoff.

$$V(S,t) = \int p(S_T, T; S, t)V(S_T, T)dS_T \qquad [57]$$

If $p(z, t = 0) = f(z)$, then the general solution is given by:

$$p(z,t) = \int p_0(z - z', t)\, f(z')dz' = \frac{1}{\sqrt{2\pi t}}\int e^{\frac{-(z-z')^2}{2t}} f(z')dz' \qquad [58]$$

Integral in [58] is over all possible future terminal values of S and depends on terminal and current values. S and t are constant under the integral. p function will give us a bridge between terminal and present values, where $V(S_T, T)$ is a boundary condition. Let us see how we can change variables in the Black-Scholes equation to arrive at the diffusion equation. First, we express the derivative in terms of its future value:

$$V(S,t) = e^{-r(T-t)}U(S,t) \rightarrow \frac{\partial U}{\partial t} + \frac{(\sigma S)^2}{2}\frac{\partial^2 U}{\partial S^2} + rS\frac{\partial U}{\partial S} = 0 \quad [59]$$

The discount factor eliminates the $-rV$ term. Now, we substitute the backward variable $\tau = T - t$ for time evolution and replace the price with its logarithm, $\varepsilon = \log S$ where $\varepsilon = [-\infty, \infty]$.

$$\frac{\partial U}{\partial \tau} - \frac{\sigma^2}{2}\frac{\partial^2 U}{\partial \varepsilon^2} - \left(r - \frac{\sigma^2}{2}\right)\frac{\partial U}{\partial \varepsilon} = 0 \quad [60]$$

If we replace $\varepsilon + \left(r - \frac{\sigma^2}{2}\right)\tau$ with x, we get:

$$\frac{\partial U}{\partial \tau} - \frac{\sigma^2}{2}\frac{\partial^2 U}{\partial x^2} = 0 \quad [61]$$

$$\frac{\partial U}{\partial \tau} = \frac{\sigma^2}{2}\frac{\partial^2 U}{\partial x^2} \quad [62]$$

The solution to the diffusion equation above will give a solution to the Black-Scholes equation. The terminal values of Black-Scholes (payoff functions) correspond to the initial values of the diffusion equation. Here is a special solution for the Black-Scholes equation:

$$U(x,\tau) = \frac{1}{\sqrt{2\pi\sigma^2\tau}} e^{\frac{-x^2}{2(2\sigma^2\tau)}} \quad [63]$$

We can easily verify the solution:

$$\frac{\partial U}{\partial x} = \left(-\frac{x}{\sigma^2\tau}\right)U, \quad \frac{\partial^2 U}{\partial x^2} = \left(-\frac{1}{\sigma^2\tau} + \frac{x^2}{\sigma^4\tau^2}\right)U, \quad \frac{\partial U}{\partial \tau} = \left(-\frac{1}{2\tau} + \frac{x^2}{2\sigma^2\tau^2}\right)U = \frac{\sigma^2}{2}\frac{\partial^2 U}{\partial x^2} \quad [64]$$

3.2.5. Application to option pricing

Now that we know the solution for the diffusion equation works for the Black-Scholes model, we can compute the formula for option pricing. Consider the probability density function of the standard stock price path with a drift replaced by the risk-free rate so that we operate under the risk-neutral measure Q.

$$dS = rSdt + \sigma Sdb \qquad [65]$$

The probability $p(S_T, T; S, t)$ satisfies:

$$V(S,t) = e^{-r(T-t)}p(S,t) \rightarrow \frac{\partial p}{\partial t} + \frac{(\sigma S)^2}{2}\frac{\partial^2 p}{\partial S^2} + rS\frac{\partial p}{\partial S} = 0 \qquad [66]$$

Which we already know can be calculated with the general solution formula for the diffusion equation [57].

$$V(S,t) = e^{-r(T-t)}E^Q[(S_T - K)_+] = e^{-r(T-t)}\int_{-\infty}^{\infty}(S_t e^z - K)_+ f(z)dz \qquad [67]$$

Where $f(z) = \frac{1}{\sigma\sqrt{2\pi(T-t)}}e^{\frac{-(z-\left(r-\frac{1}{2}\sigma^2\right)(T-t))^2}{2\sigma^2(T-t)}}$. Let $x = \left(r - \frac{1}{2}\sigma^2\right)(T-t)$, $y^2 = \sigma^2(T-t)$ and $A = \log\frac{K}{S_t}$.

$$e^{-r(T-t)}\int_{-\infty}^{\infty}(S_t - K)_+ f(z)dz = e^{-r(T-t)}\int_A^{\infty}(S_t e^z - K)_+ \frac{1}{\sqrt{2\pi y^2}}e^{\frac{-(z-x)^2}{2y^2}}dz =$$
$$= e^{-r(T-t)}S_t\int_A^{\infty}e^z\frac{1}{\sqrt{2\pi y^2}}e^{\frac{-(z-x)^2}{2y^2}}dz - e^{-r(T-t)}K\int_A^{\infty}\frac{1}{\sqrt{2\pi y^2}}e^{\frac{-(z-x)^2}{2y^2}} \qquad [68]$$

Calculate the first integral:

$$\int_A^{\infty}e^z\frac{1}{\sqrt{2\pi y^2}}e^{\frac{-(z-x)^2}{2y^2}}dz = \frac{1}{\sqrt{2\pi y^2}}\int_A^{\infty}e^{\frac{-(z^2-2zx+x^2-2y^2z)}{2y^2}}dz \qquad [69]$$

After completing the square we have:

$$\frac{1}{\sqrt{2\pi y^2}}\int_A^\infty e^{\frac{-\left(z-(x+y^2)\right)^2}{2y^2}} e^{\frac{\left(x+y^2\right)^2-x^2}{2y^2}} dz \quad [70]$$

The second exponential simplifies to $e^{r(T-t)}$:

$$\frac{1}{\sqrt{2\pi y^2}}\int_A^\infty e^{\frac{-\left(z-(x+y^2)\right)^2}{2y^2}} e^{r(T-t)} dz \quad [71]$$

Make a substitution $z' = \frac{z-(x+y^2)}{y}$:

$$e^{r(T-t)} \int_{\frac{A-(x+y^2)}{y}}^\infty \frac{1}{\sqrt{2\pi}} e^{\frac{-z'^2}{2}} dz' = e^{r(T-t)}\left(1-\Phi\left(\frac{A-(x+y^2)}{y}\right)\right) = e^{r(T-t)}\Phi\left(\frac{-A+x+y^2}{y}\right) \quad [72]$$

Where Φ is the standard normal probability density function. Now, we can calculate the second integral. First make a substitution $z'' = \frac{z-x}{y}$:

$$\int_A^\infty \frac{1}{\sqrt{2\pi y^2}} e^{\frac{-(z-x)^2}{2y^2}} = \int_{\frac{A-x}{y}}^\infty \frac{1}{\sqrt{2\pi}} e^{\frac{-z''^2}{2}} dz'' = 1-\Phi\left(\frac{A-x}{y}\right) = \Phi\left(\frac{-A+x}{y}\right) \quad [73]$$

Therefore, the current price of the call option contract is given by:

$$C_t = S_t\Phi\left(\frac{-A+x+y^2}{y}\right) - Ke^{-r(T-t)}\Phi\left(\frac{-A+x}{y}\right) \quad [74]$$

$$C_t = S_t\Phi(d_1) - Ke^{-r(T-t)}\Phi(d_2) \quad [75]$$

Where d_1 and d_2 are:

$$d_1 = \frac{\log\frac{S_t}{K}+\left(r+\frac{1}{2}\sigma^2\right)(T-t)}{\sigma\sqrt{(T-t)}} \quad [76]$$

$$d_2 = \frac{\log\frac{S_t}{K}+\left(r-\frac{1}{2}\sigma^2\right)(T-t)}{\sigma\sqrt{(T-t)}} \quad [77]$$

The relationship between d_1 and d_2 is:

$$d_2 = d_1 - \sigma\sqrt{(T-t)} \tag{78}$$

From the put-call parity, we can also derive the formula for the vanilla European put option.

$$C_t - P_t = S_t - Ke^{-r(T-t)} \tag{79}$$

$$P_t = Ke^{-r(T-t)}\Phi(-d_2) - S_t\Phi(-d_1) \tag{80}$$

3.2.6. Implementation

The implementation of the Black-Scholes pricing mechanism for the call and put option contracts is relatively straightforward.

```r
BSPrice<-function(K, T, s0, r, sigma){

  d1 <- (1/(sigma*sqrt(T)))*(log(s0/K)+(r+sigma^2/2)*T)
  d2 <- d1-sigma*sqrt(T)
  call_price <- pnorm(d1)*s0-pnorm(d2)*K*exp(-r*T)
  put_price <- pnorm(-d2)*K*exp(-r*T)-pnorm(-d1)*s0

  output<-list(call_price=call_price, put_price=put_price)

  return(output)
}
```

Figure 9: Black-Scholes model implementation in R

The $BSPrice$ function takes as an input the following list of parameters:

- T terminal time of the contract
- r risk-free rate
- $sigma$ volatility σ
- $s0$ initial price of the asset
- K strike price of the contract

4. Finite Difference Methods for solving PDEs

In this section, we will derive and implement the three finite difference methods (FDMs) that can be used to price options by approximating the continuous time differential equation by the finite difference equations. The Black-Scholes equation can usually be solved with the analytic solution, as seen in the previous section. At the same time, the finite difference methods allow us to calculate approximate solutions for $V(S,t)$ over the small discrete time intervals. Depending on what difference equations we use to approximate the differential equation, we can get three distinct FDMs – explicit method, implicit method, and Crank-Nicolson method. The main concept revolves around discretizing a differential equation, which is a process of transferring the continuous function into its discrete counterpart. There are three ways we can approximate the Black-Scholes equation.

$$\frac{\partial V}{\partial t} + \frac{(\sigma S)^2}{2}\frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} = rV \qquad [81]$$

The well-known discrete derivative form for any function f with a difference step h is given by:

$$f'(x) = \lim_{h\to 0}\frac{f(x+h)-f(x)}{h} \qquad [81]$$

Now, consider the Taylor's series expansion for $f(x+h)$:

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \frac{h^3}{3!}f'''(x) + \cdots \qquad [82]$$

Subtract $f(x)$ and divide [82] by h to obtain the forward approximation.

$$\frac{f(x+h)-f(x)}{h} = f'(x) + \frac{h^2}{2}f''(x) + \frac{h^3}{3!}f'''(x) + \cdots = \frac{f(x+h)-f(x)}{h} + O(h) \qquad [83]$$

$$f'(x) = \frac{f(x+h)-f(x)}{h} + O(h) \qquad [84]$$

The backward approximation is obtained by Taylor's series expansion of $f(x-h)$.

$$f(x-h) = f(x) - hf'(x) + \frac{h^2}{2}f''(x) - \frac{h^3}{3!}f'''(x) + \cdots \qquad [85]$$

$$f'(x) = \frac{f(x)-f(x-h)}{h} + O(h) \qquad [86]$$

We get the central approximation by subtracting [85] from [82] and dividing by $2h$.

$$f'(x) = \frac{f(x+h)-f(x-h)}{2h} = f'(x) + \frac{h^2}{3!}f'''(x) + \frac{h^4}{5!}f'''''(x) = f'(x) + O(h^2) \qquad [87]$$

This time, we ended up with an error of order h^2. Hence, this approximation should converge to the true value faster than forward and backward approximations. To fully discretize the Black-Scholes equation, we also need a standard approximation for a second-order derivative. This can be done by adding the [82] and [85], and dividing by h^2.

$$f''(x) = \frac{f(x+h)-2f(x)+f(x-h)}{h^2} + O(h^2) \qquad [88]$$

The second step after deriving the derivatives approximations is to specify a grid for underlying asset prices and its boundaries. The discrete grid is typically generated by dividing the time axis $[0, T]$ into N equal time step of size dt and price axis into M equal steps of size dS. This gives us a grid consisting of points (S, t) with $N + 1$ time points and $M + 1$ price points. The solution space is bounded by the option's maturity time T and terminal condition for the call option's payoff:

$$f_{i,N} = (idS - K)_+ \qquad [89]$$

And for the put option:

$$f_{i,N} = (K - idS)_+ \qquad [90]$$

4.1. Explicit method

The process of discretization of Black-Scholes PDE is as follows:

- Use backward approximation for $\frac{\partial V}{\partial t}$ term.

$$\frac{\partial V}{\partial t} = \frac{V_{i,j} - V_{i-1,j}}{dt} \qquad [91]$$

- Use central approximation for $\frac{\partial V}{\partial S}$ term.

$$\frac{\partial V}{\partial S} = \frac{V_{i,j+1} - V_{i,j-1}}{2dS} \tag{92}$$

- Use standard approximation for second-order derivative $\frac{\partial^2 V}{\partial S^2}$.

$$\frac{\partial^2 V}{\partial S^2} = \frac{V_{i,j+1} - 2V_{i,j} + V_{i,j-1}}{dS^2} \tag{93}$$

Substitute approximations back into the [81] to obtain an explicit finite difference equation.

$$\frac{V_{i,j} - V_{i-1,j}}{dt} + \frac{(\sigma j dS)^2}{2} \frac{V_{i,j+1} - 2V_{i,j} + V_{i,j-1}}{dS^2} + rjdS \frac{V_{i,j+1} - V_{i,j-1}}{2dS} = rV_{i,j} \tag{94}$$

$$V_{i-1,j} = a_j V_{i,j-1} + b_j V_{i,j} + c_j V_{i,j+1} \tag{95}$$

Where a_j, b_j and c_j are explicit finite difference parameters.

$$a_j = \frac{1}{2} dt(\sigma^2 j^2 - rj) \tag{96}$$

$$b_j = 1 - dt(\sigma^2 j^2 + r) \tag{97}$$

$$c_j = \frac{1}{2} dt(\sigma^2 j^2 + rj) \tag{98}$$

Looking at [95], we can deduce that given option terminal prices at expiry time $i = N$, we can calculate all the previous points inside the grid by going backward in time, until we arrive at the option price at the current time. For computations, we can rewrite [95] in the form of a matrix notation.

$$V_{i-1} = AV_i + K_i \tag{99}$$

$$\begin{bmatrix} V_{i-1,1} \\ V_{i-1,2} \\ \vdots \\ \vdots \\ V_{i-1,M-1} \end{bmatrix} = \begin{bmatrix} b_1 & c_1 & 0 & \dots & 0 & 0 \\ a_2 & b_2 & c_2 & \dots & 0 & 0 \\ 0 & a_3 & b_3 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & a_{M-1} & b_{M-1} \end{bmatrix} \begin{bmatrix} V_{i,1} \\ V_{i,2} \\ \vdots \\ \vdots \\ V_{i,M-1} \end{bmatrix} + \begin{bmatrix} a_1 V_{i,0} \\ 0 \\ \vdots \\ 0 \\ c_{M-1} V_{i,M} \end{bmatrix} \tag{100}$$

The equation [99] is only stable when $\|A\|_\infty \leq 1$ and for some combinations of values a_j, b_j and c_j this is not the case, meaning the infinity norm of A is greater than 1, making an explicit method unstable.

4.2. Implicit method

The only difference between implicit and explicit methods is that we use a forward approximation for $\frac{\partial V}{\partial t}$ term.

$$\frac{\partial V}{\partial t} = \frac{V_{i+1,j} - V_{i,j}}{dt} \tag{101}$$

The rest of the terms are identical to the ones in the explicit method. Substituting back into the original PDE:

$$\frac{V_{i+1,j} - V_{i,j}}{dt} + \frac{(\sigma j dS)^2}{2}\frac{V_{i,j+1} - 2V_{i,j} + V_{i,j-1}}{dS^2} + rjdS\frac{V_{i,j+1} - V_{i,j-1}}{2dS} = rV_{i,j} \tag{102}$$

$$a_j V_{i,j-1} + b_j V_{i,j} + c_j V_{i,j+1} = V_{i+1,j} \tag{103}$$

Where:

$$a_j = \frac{1}{2}dt(rj - \sigma^2 j^2) \tag{104}$$

$$b_j = 1 + dt(\sigma^2 j^2 + r) \tag{105}$$

$$c_j = \frac{1}{2}dt(-rj - \sigma^2 j^2) \tag{106}$$

This time, when going backward in time, the calculations are not trivial as from a single value at time T we have to calculate three values for the previous time step $T - dt$. We again use the system of linear equations for calculations:

$$BV_i = V_{i+1} - K_i \tag{107}$$

$$\begin{bmatrix} b_1 & c_1 & 0 & \dots & 0 & 0 \\ a_2 & b_2 & c_2 & \dots & 0 & 0 \\ 0 & a_3 & b_3 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & a_{M-1} & b_{M-1} \end{bmatrix} \begin{bmatrix} V_{i,1} \\ V_{i,2} \\ \vdots \\ \vdots \\ V_{i,M-1} \end{bmatrix} = \begin{bmatrix} V_{i+1,1} \\ V_{i+1,2} \\ \vdots \\ \vdots \\ V_{i+1,M-1} \end{bmatrix} - \begin{bmatrix} a_1 V_{i,0} \\ 0 \\ \vdots \\ 0 \\ c_{M-1} V_{i,M} \end{bmatrix} \qquad [108]$$

The algorithm is stable if $\|B^{-1}\|_\infty \leq 1$ and this time it can be proven that it is the case for all combinations of values a_j, b_j and c_j – hence, the implicit method is always stable. However, this method requires calculating the inverse matrix B^{-1}, which can be computationally expensive.

4.3. Crank-Nicolson method

This method combines the explicit and implicit methods and represents their average. It was introduced to improve the accuracy up to order $O(dt^2)$. Consider *Figure 10* representing the grid notes:

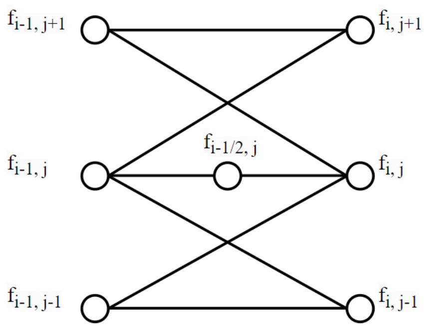


Figure 10: Crank-Nicolson method visualization

The explicit method prices the node $f_{i-1,j}$ using three nodes on the right-hand side, while the implicit method prices the left-hand side nodes using the value of the $f_{i,j}$ node. The average of those two methods would be to price all the nodes on the left-hand side using all the nodes on the right-hand side. To discretize the Black-Scholes PDE, consider the central node $f_{i-1/2,j}$ and its approximations:

- Use central approximation for $\frac{\partial V}{\partial t}$ term.

$$\frac{\partial V_{i-1/2,j}}{\partial t} = \frac{V_{i,j} - V_{i-1,j}}{dt} \qquad [109]$$

- Use central approximation for $\frac{\partial V}{\partial S}$ term.

$$\frac{\partial V_{i-1/2,j}}{\partial S} = \frac{1}{2}\left[\frac{V_{i-1,j+1}-V_{i-1,j-1}}{2dS} + \frac{V_{i,j+1}-V_{i,j-1}}{2dS}\right] \quad [110]$$

- Use standard approximation for second-order derivative $\frac{\partial^2 V}{\partial S^2}$.

$$\frac{\partial^2 V_{i-1/2,j}}{\partial S^2} = \frac{1}{2}\left[\frac{V_{i-1,j+1}-2V_{i-1,j}+V_{i-1,j-1}}{dS^2} + \frac{V_{i,j+1}-2V_{i,j}+V_{i,j-1}}{dS^2}\right] \quad [111]$$

Substituting into the original PDE:

$$\frac{V_{i,j}-V_{i-1,j}}{dt} + \frac{rjdS}{2}\frac{V_{i-1,j+1}-V_{i-1,j-1}}{2dS} + \frac{rjdS}{2}\frac{V_{i-1,j+1}-V_{i-1,j-1}}{2dS} + \frac{(\sigma jdS)^2}{4}\frac{V_{i-1,j+1}-2V_{i-1,j}+V_{i-1,j-1}}{dS^2} + \frac{(\sigma jdS)^2}{4}\frac{V_{i,j+1}-2V_{i,j}+V_{i,j-1}}{dS^2} = \frac{r}{2}V_{i-1,j} + \frac{r}{2}V_{i,j} \quad [112]$$

Which reduces to:

$$-a_j V_{i-1,j-1} + (1-b_j)V_{i-1,j} - c_j V_{i-1,j+1} = a_j V_{i,j-1} + (1+b_j)V_{i,j} + c_j V_{i,j+1} \quad [113]$$

Where:

$$a_j = \frac{1}{4}dt(\sigma^2 j^2 - rj) \quad [114]$$

$$b_j = -\frac{dt}{2}(\sigma^2 j^2 + r) \quad [115]$$

$$c_j = \frac{1}{4}dt(\sigma^2 j^2 + rj) \quad [116]$$

We have $N-1$ linear equations with $N-1$ unknowns. We can get a unique value for each node by solving the system of linear equations simultaneously.

$$CV_{i-1} = DV_i + K_{i-1} + K_i \quad [117]$$

$$\begin{bmatrix} 1-b_1 & -c_1 & 0 & \dots & 0 & 0 \\ -a_2 & 1-b_2 & -c_2 & \dots & 0 & 0 \\ 0 & -a_3 & 1-b_3 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & -a_{M-1} & 1-b_{M-1} \end{bmatrix} \begin{bmatrix} V_{i-1,1} \\ V_{i-1,2} \\ \vdots \\ \vdots \\ V_{i-1,M-1} \end{bmatrix} =$$

$$= \begin{bmatrix} 1+b_1 & c_1 & 0 & \dots & 0 & 0 \\ a_2 & 1-b_2 & c_2 & \dots & 0 & 0 \\ 0 & a_3 & 1+b_3 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & a_{M-1} & 1+b_{M-1} \end{bmatrix} \begin{bmatrix} V_{i,1} \\ V_{i,2} \\ \vdots \\ \vdots \\ V_{i,M-1} \end{bmatrix} + \begin{bmatrix} a_1 V_{i-1,0} \\ 0 \\ \vdots \\ 0 \\ c_{M-1} V_{i-1,M} \end{bmatrix} + \begin{bmatrix} a_1 V_{i,0} \\ 0 \\ \vdots \\ 0 \\ c_{M-1} V_{i,M} \end{bmatrix} \quad [118]$$

The algorithm is stable if $\|C^{-1}D\|_\infty \leq 1$, and as with the implicit method, this is always the case, meaning the Crank-Nicolson method is always stable. The method converges at the rates $O(dt^2)$ and $O(dS^2)$ which are faster than in the previous methods. We again have to calculate an inverse matrix C^{-1} that is computationally inefficient, nonetheless, for tri-diagonal matrices there are some algorithms that can speed up the inversion calculation.

5. Results

5.1. Monte Carlo

For the contract with the initial value of the underlying asset and strike price both equal to $100, where the number of simulations is 10,000, expiration time is in exactly 1 year, the number of steps until the expiration equals 252 trading days with a volatility of 0.3 and a risk-free rate set to 0.1, the initial values of the put and call options are $16.69 and $7.21, with a standard error of 0.24 and 0.11, respectively. For the same input values, the model that uses the importance sampling technique gives similar values, however, they are less volatile as their standard error is now lower as presented in *Figure 13*.

```
sigma <- 0.3
mu <- 0.05
Nt <- 252
Np <- 1e4
s0 <- 100
K <- 100
r <- 0.1
T <- 1

results<-MCPrice(Np, Nt, T, r, sigma, s0, K)
```

Figure 11: Monte Carlo calculation

```
$call_price
[1] 16.66881

$err_call
[1] 0.238326

$put_price
[1] 7.210073

$err_put
[1] 0.1118791
```

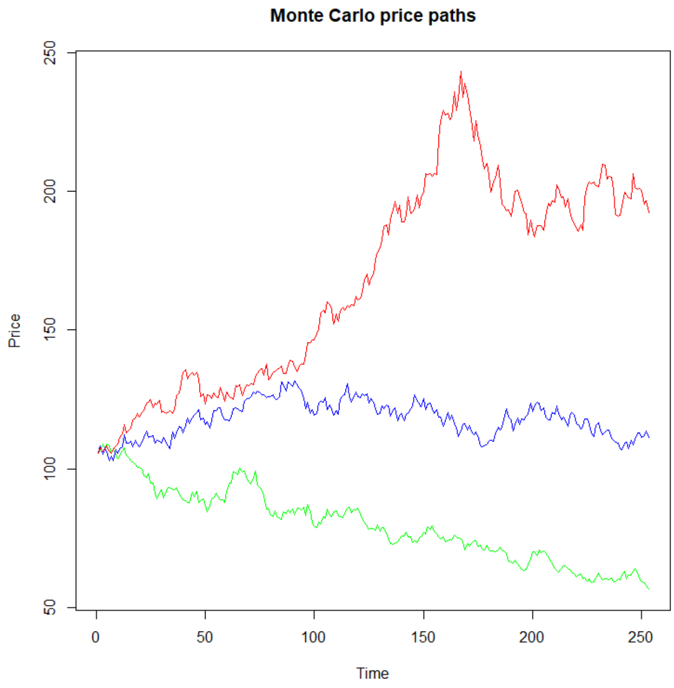
Figure 12: Monte Carlo results

```
$call_price
[1] 16.86197

$err_call
[1] 0.1481406

$put_price
[1] 7.111502

$err_put
[1] 0.04855791
```

Figure 13: Monte Carlo with importance sampling results

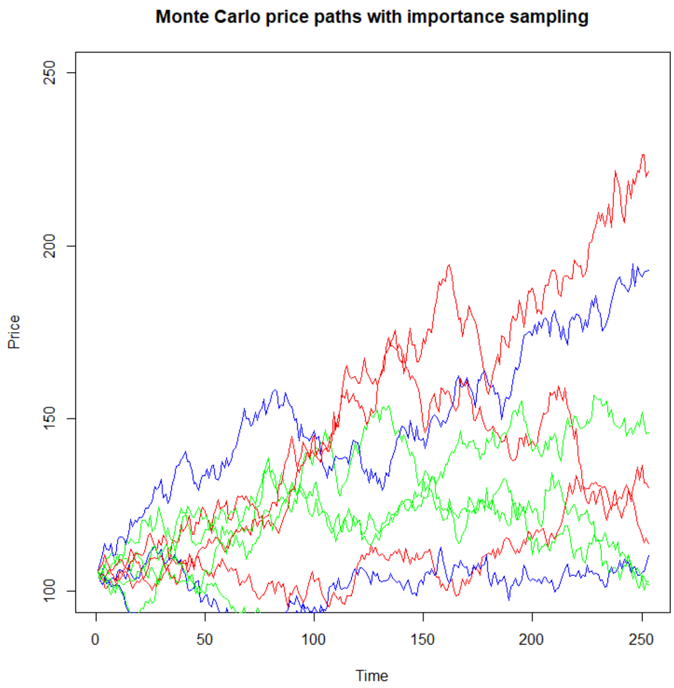
Dividing the time to payoff by the number of trading days in the simulation means that the Monte Carlo model operates in discrete time steps. However, the accuracy of results is not sensitive to the duration of steps Δt, because the sum of binomial random converges quickly to a Gaussian distribution as expected under the Central Limit Theorem.

Graph 1 depicts three different possible price paths. The red function is the extreme case where the terminal value finished well above the strike value, making a call option in-the-money. The green function represents another possible extreme where the terminal value is realized below the strike price, making, in this case, the put option contract profitable.



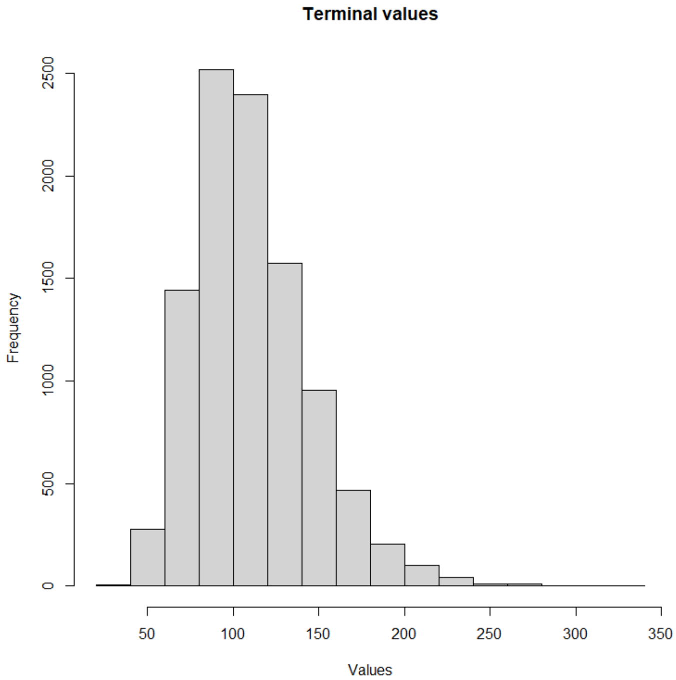
Graph 1: Monte Carlo price paths

Graph 2, on the other hand, represents the call option price paths that we are only concerned with when performing the importance sampling. All the paths that finish below the strike price of $100 are cut off as they do not contribute to the payoff calculation.



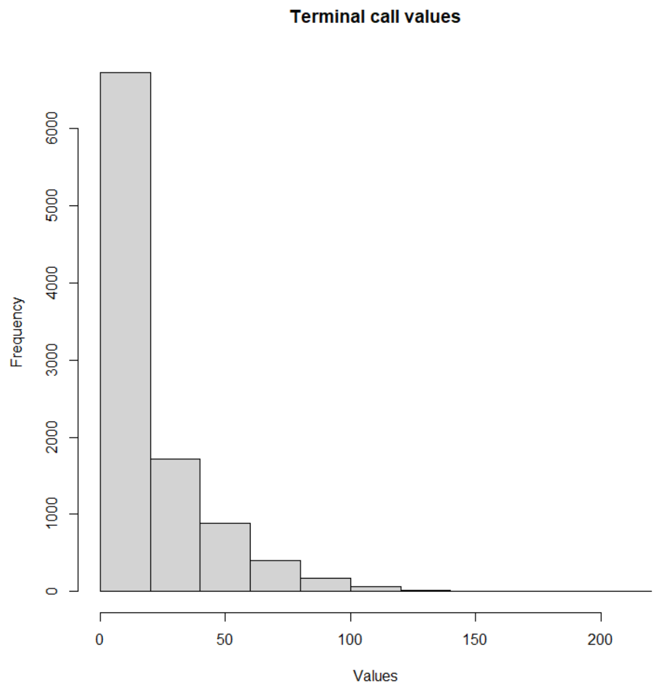
Graph 2: Monte Carlo with importance sampling price paths

Graph 3 shows that the price paths are log-normally distributed as expected. The peak on the histogram around the value of the strike price of $100 proves that if the random walk starts at strike K, most prices will end up closer to the origin value – this is dependent on the drift value. The blue function on the *Graph 1* is an example of such a path.



Graph 3: Monte Carlo terminal values histogram

Moreover, we can further investigate this behaviour of the random walk by looking at *Graph 4* that shows the discounted expectation of the call payoffs before taking their average. On the histogram, there is a spike at value 0, meaning many paths finish out-of-money, and they are then decreasing in quantities for greater prices. This is expected for a Gaussian distribution as it is less likely that the stock will significantly exceed the strike price by finishing at some extreme value at the tail of the distribution.



Graph 4: Monte Carlo terminal call values histogram

The precision of the Monte Carlo model can be reflected by standard error that decreases as $\frac{1}{\sqrt{N_p}}$, hence the accuracy of results can be significantly improved by increasing the number of paths N_p, as we will see in the results comparison section.

5.2. Black-Scholes

The Black-Scholes model results in unique, true values for the initial price of the call and put option contracts. The results for the contract with an initial value of \$100 of the underlying asset, strike at the \$100, volatility value set to 0.3, interest rate at 0.1 and time to expiry equal to 1 year are as following:

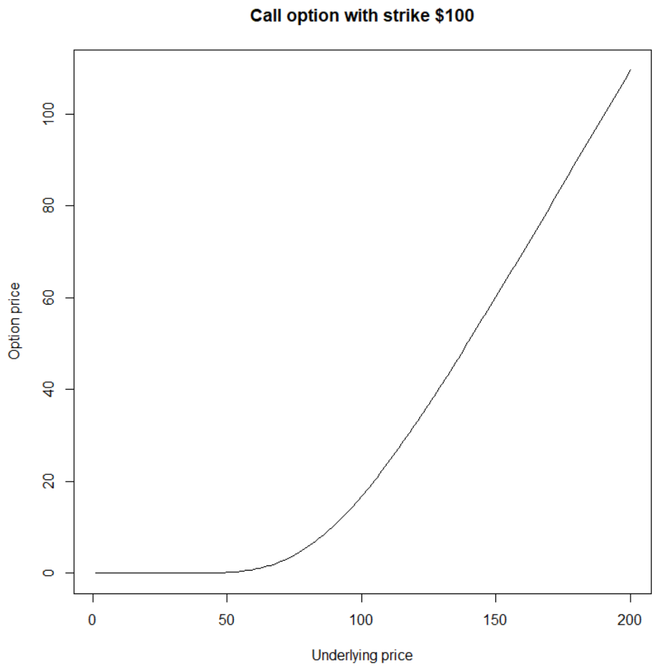
```
BSPrice(100, 1, 100, 0.1, 0.3)
```

Figure 14: Black-Scholes calculation

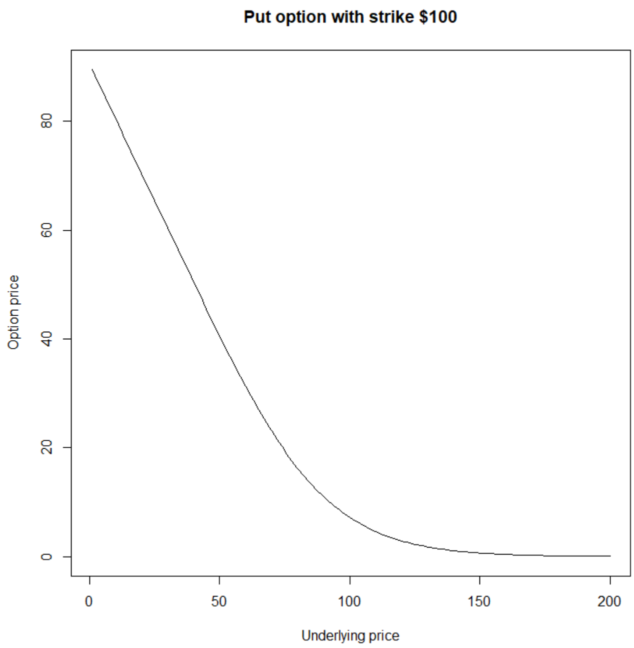
```
$call_price
[1] 16.73413

$put_price
[1] 7.217875
```

Figure 15: Black-Scholes results

The initial price of the call option contract is \$16.73, and the price for the put option is \$7.22. We can look at how the value of the underlying affects the price of the option contracts by keeping the strike price constant at \$100.



Graph 5: Black-Scholes call option price function



Graph 6: Black-Scholes put option price function

Those two graphs represent the change in the option contract value when we manipulate the underlying stock price ($\$0 - \$200$). They look exactly like the option payoff graphs from the beginning, and this makes sense as for the call option, if we keep the strike constant and increase the value of the underlying stock, we expect to pay more for the option as with the stock value further from strike price the option is less likely to finish below it and we are more likely to receive a positive payoff. The opposite happens for the put option contract. As the value of the underlying is less and less below the strike price, we expect a higher payoff, which is reflected by a higher price we have to pay to purchase the contract.

5.3. Comparison

In the Monte Carlo model, the extreme cases, such as an option that is deep out-of-money, will require more paths to achieve a good level of precision since most of the paths will contribute to a terminal value of zero. We can rerun the Monte Carlo model with a different number of simulations and analyse how the model converges to true values given by the Black-Scholes model as the number of simulations grows.

Number of simulations	1000	10000	100000	1000000
Call option price	17.69697	17.085740	16.773420	16.743510
Put option price	6.95561	7.134678	7.274046	7.213352

Table 1: Convergence of Monte Carlo values

As expected, with the increased number of simulations, the model converges to true values. Nonetheless, we have to keep in mind that the Monte Carlo is only a simulation based on the expectation, and it will result in different values each time we run the code. After precisely 1 million simulated paths, the model gives us a price for a call option of \$16.74 and a put option of \$7.21 which is almost the same as what we got from the Black-Scholes formula.

6. Summary

This personal project was my first attempt at research in the field of financial mathematics. It allowed me to better understand the concepts I have learned over the past few months by diving deeper into the mathematical details of the model derivation. Thanks to Dr. Ryan Donnelly, my lecturer in the Financial Mathematics module, I have expanded the scope of this paper and looked into more complicated numerical methods for solving partial differential equations. I am now more confident with my skills in solving stochastic and partial differential equations. This project also proved to be a perfect opportunity to learn a new programming language R, by attempting to implement basic Black Scholes and Monte Carlo models. In the future, I plan to write another similar project on portfolio optimization where I will work on my linear algebra skills by implementing the minimum variance portfolio method. Moreover, I hope to extend this project by implementing all three finite difference methods I learned about and analyse the obtained results to better understand their stability and convergence rates.

7. Bibliography

List of figures

Figure 1: Simple stock model 4
Figure 2: Call payoff model 4
Figure 3: Put option payoff model 5
Figure 4: Call option payoff function 5
Figure 5: Put option payoff function 5
Figure 6: General one-period binomial model 5
Figure 7: Monte Carlo model implementation in R 11
Figure 8: Monte Carlo model with importance sampling implementation code in R 12
Figure 9: Black-Scholes model implementation in R 21
Figure 10: Crank-Nicolson method visualization 26
Figure 11: Monte Carlo calculation 29
Figure 12: Monte Carlo results 29
Figure 13: Monte Carlo with importance sampling results 29
Figure 14: Black-Scholes calculation 32
Figure 15: Black-Scholes results 32

List of graphs

Graph 1: Monte Carlo price paths 30
Graph 2: Monte Carlo with importance sampling price paths 30
Graph 3: Monte Carlo terminal values histogram 31
Graph 4: Monte Carlo terminal call values histogram 32
Graph 5: Black-Scholes call option price function 33
Graph 6: Black-Scholes put option price function 33

List of tables

Table 1: Convergence of Monte Carlo values 34

List of references

- Shreve, Steven E. 2005. Stochastic Calculus for Finance 1, the Binomial Asset Pricing Model. New York: Springer.
- Shreve, Steven E. 2010. Stochastic Calculus for Finance. II, Continuous-Time Models. New York: Springer.
- “MATLAB and Simulink Consulting Services.” n.d. www.goddardconsulting.ca. Accessed October 30, 2023. http://www.goddardconsulting.ca/index.html.