M. Langdown 1761704

Numerical Methods of Pricing Financial Options

1 Introduction

1.1 Why are financial options important?

Financial Options are derivatives of real world assets and are used mainly either try and create profit through placing bets on the prospects of an asset or, they can be used to reduce risk by betting against the event you want to happen so that you are insured against the opposite. For example, if a business moves towards gold mining it may be appropriate to take out an option which favours the depreciation of the value of gold, that way, if gold drops in value, their business will suffer but they will also make a small profit and reduce their overall losses since the cost of the insuring option will be low. Options trading underpinned massive economic growth shortly after the derivation of the Black-Scholes model [Stewart (2012)]. Options trading has grown to be even larger than asset trading, in fact, $596 trillion is the estimated worth of the over the counter derivatives market compared to $167 trillion estimated total worldwide asset growth [Leibenluft (2008)].
The Black-Scholes equation was very popular when it was first derived but unfortunately it was largely abused and this meant that it could often lead to economic slump and also near collapse of the banking system [Stewart (2012)]. Over time, various abuses of the Black-Scholes equation arose when people ignored the fact that it could not provide accurate answers when market conditions were unsuitable, for example when big news stories hit and made the stock market very volatile.

1.2 Why do we need to price options?

It is very important to price financial options because options are contracts decided between a writer and buyer who must agree on the price of an option before purchasing. The methods of pricing options give writers and buyers a good idea of how much the option should be worth. Furthermore, options are unique in that they can be traded during the contract prior to expiry and as such it is useful to be able to price options during multiple points in their lifespan so that this type of trading can be supported.

1.3 What are the aims of this paper?

This paper aims to apply, expand upon and compare some of the pricing methods taught in Mathematical Finance with alternative methods. The application and comparisons will come from the python files in which I have directly programmed the algorithms later in the paper in python so that answers can be produced from the methods for comparison. Also I will make comments on future methods and why they are better than older numerical solutions detailed in this paper.

2 Financial Derivative Definitions and Theorems

2.1 What are financial options?

Financial Derivatives are referred to as derivatives because their value is derived from a real life asset's value. **Financial Options** are derivatives based on an underlying asset commonly called S. Financial options are predictions on what will happen to the value of S over a time period up to an expiry or maturity often denoted T. At the expiry, the holder of the option is given the option to buy/sell the asset or accept the loss, and such the writer must go along with what the holder decides. A **European Vanilla Option** is an option that can only be exercised at the expiry. There are two types of options to consider:

1. **European Vanilla Call Options** give the holder the right to purchase an asset at a price referred to as the strike price, denoted E, at expiry which is agreed upon writing the option. If the value of the underlying asset, S, is greater than E at expiry, then the holder of the option makes a profit.

2. **European Vanilla Put Options** give the holder the right to sell an asset at a price referred to as the strike price, at expiry which is agreed upon writing the option. If the value of the underlying asset, S, is less than E at expiry, then the holder of the option makes a profit.

These two options have holders profits that are given by the following two equations:

$$\text{European Vanilla Call Payoff} = \max(\{S - E, 0\}), \tag{1}$$

$$\text{European Vanilla Put Payoff} = \max(\{E - S, 0\}), \tag{2}$$

where S is the value of the underlying asset on which the option is based, E is the strike price. Sometimes options can be **At-the-Money** (**ATM**) which is where the underlying asset S is at the strike price E. Both a call and a put option with similar properties can be ATM simultaneously. An option which is ATM has no intrinsic value but can be sold as an option due to prospects of S going up or down in future. Below shows the graph of how the payoff of these options is affected by the value of their underlying asset.

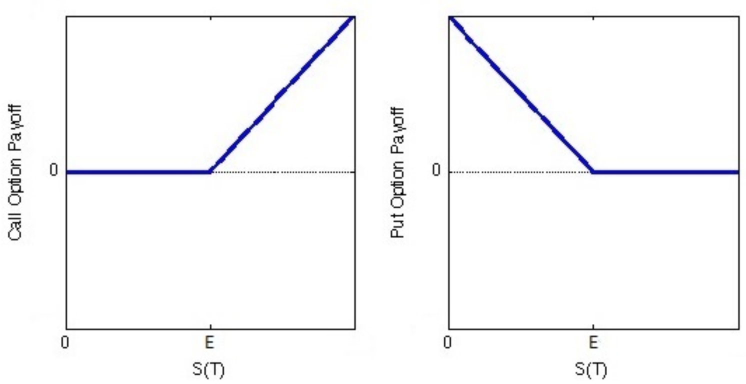

Figure 1: Payoff Function Diagrams for European Vanilla Options adapted from [Knight (2014)]

2.2 Terms and Theorems On Financial Options

Listed below are the common terms and useful theorems that will be used and referenced throughout this paper.

Volatility refers to the risk or uncertainty related to the fluctuations of asset or stock pricing. Larger volatility implies greater fluctuation of stock prices and means that the stock prices can move up and down larger amounts whilst following the general trend. The volatility of an underlying asset can affect any financial options dependant upon that asset, higher volatility of asset price implies a higher price for options since their payoffs are truncated at zero [Bodie & Merton (1998)], meaning the large drops in value will only reduce payoff to zero and the large rises in value will increase the payoff linearly.

Long selling is referring to selling assets that you actually own. **Short selling** on the other hand is referring to selling assets at current market value that you do not own and then are obliged to purchase at an either specified or unspecified time in the future.

Arbitrage can be described as the opportunity to make instantaneous risk-free profit. Arbitrage is very sought after and there are highly paid **arbitrageurs** who try to profit from price differences in different assets in an attempt to find arbitrage opportunities. However, due to this large scale and constant search for arbitrage the arbitrage opportunities get used up very quickly.

The **No Arbitrage Principle** allows assumption of no arbitrage opportunities due to the fact that supply and demand cause these opportunities to get used up very quickly. This can be stated mathematically as: If at time T an asset A has price $A(T)$ below and above respectively by $k, K \in \mathbb{R}$. Then at a later time, t we have $ke^{r(t-T)} \leqslant A(t) \leqslant Ke^{r(t-T)}$.

The **Law of One Price** states that the price of a financial asset will be the same regardless of the location when exchange rates are taken into account. This stems from the fact that differences in price of the asset would create arbitrage opportunities which cannot be true due to the No Arbitrage Principle.

Dividends are payments made by companies to stock or asset holders on a particular date known as the **ex-dividend**. The dividend will be paid to share holders only on the ex-dividend date on not to any previous share holders that have since sold their shares. Companies will offer share holders the option to pay their dividend in and purchase more stocks with it. The **dividend yield**, D_y, can be described as the ratio of a companies annual dividend compared to the stock price, that is, $D_y = \frac{\text{Annual Dividend}}{\text{Stock Price}}$.

Risk-free Rate is the interest an investor would expect to receive on an investment which carries no risk. A true risk-free rate does not exist because investments always carry a level of risk. Governmental treasuries are usually used for risk-free rate as there is almost no chance that a government will default on its financial obligations. For example, in the U.S it is reasonable to use the three-month U.S Treasury Bill as a risk-free rate for U.S based investors.

3 Basic Two-State Model

Let us consider a very basic option model where we will compare stock price and call option payoff on common underlying asset S. We assume the option has strike price E and has expiry T. Furthermore, we make the assumption that S will take only one of two values at expiry with equal probability 0.5 which differ by some d from the starting asset value. Then we have:

$$\mathbb{E}(\text{Stock Payoff}) = 0.5(S+d) + 0.5(S-d) = S,$$

$$\mathbb{E}(\text{Call Option Payoff}) = 0.5(d) + 0.5(0) = \frac{d}{2},$$

where $d \in \mathbb{R}$, is determined by the volatility of the stock prices. In particular we notice that for low volatility we will have a smaller d which means we will have a lower expected call option payoff but it will not affect expected stock payoff. But when we have high volatility we will have a larger d which means we will have a larger expected call option payoff but once again will not affect expected stock payoff. We also notice that for stock payoff the potential profits or losses are less than an option over the same asset, this affect is described as **gearing** (which is way traders often speculate and make bets, options are often traded more than assets for this reason [Leibenluft (2008)]). We now wish to start to expand the model, but first we need to be able to describe the value of an option by comparing it with its synthetic version since pricing the raw option is more difficult.

4 Synthetic Option Pricing

4.1 What are synthetic options?

A **Synthetic Call Option** is an option where an investor buys and holds shares in a stock or asset and then purchases an ATM put option on the same stock or asset as insurance.
A **Synthetic Put Option** is an option where an investor who holds stock or assets which are short will purchase an ATM call option as insurance.

4.2 Synthetic call pricing

To price a call option we start by creating a synthetic call option. By the Law of One Price, we have that the price of the call must be equal to the cost of creating the synthetic call option. To create a synthetic call option we let h describe the **hedge ratio**, the ratio of the stock or asset you purchase, we simultaneously borrow an amount of money denoted b. Then the formula for the cost, C_c of the synthetic call option is,

$$C_c = hS - b, \tag{3}$$

where S is the value of the underlying stock or asset [Bodie & Merton (1998)].

4.3 Synthetic put pricing

Similarly, to price a put option we start by creating a synthetic put option. By the Law of One Price, we have that the price of the call must be equal to the cost of creating the synthetic put option. To create a synthetic put option we let h describe the hedge ratio, but this time we short-sell that fraction of the share and we lend an amount of money b. Then the formula for the cost, C_p of the synthetic put option is,

$$C_p = -hS + b, \tag{4}$$

where S is the value of the underlying stock or asset [Bodie & Merton (1998)]. These synthetic options allow us to find the value of the original option by direct equality, we can start to see what values these synthetic options can take at different times by constructing a binomial tree, detailed in the next section.

5 Binomial Option Pricing

5.1 Upgrading the model

The below derivation is a deviation from Dr Leppinen's binomial method and gives a different model. For simplicity we assume that the stock or asset pays no dividends. We review the assumption that the stock or asset will take one of two possible prices at its expiry and we instead subdivide the time period T into n subdivisions each of length $\frac{T}{n}$, and assume that the stock can go up or down by some amount d. Then the stock or assets final price is bounded above by $S + nd$ and below by S. Now we can consider our options at each of the $n-1$ points, as the end point has no options. This makes the following decision tree,
Figure 2 is a binomial tree diagram adapted from [Bodie & Merton (1998)] which shows the possible values of the stock at each of the $n-1$ nodes and allows us to see all possible paths that the value of the stock can take. The bottom half of the binomial tree is not there because as soon as you start to make a loss, you sell the stock and pay the debt. This improved model detailed above is known as the **Binomial Option-Pricing Model**, which we will futher generalise below.

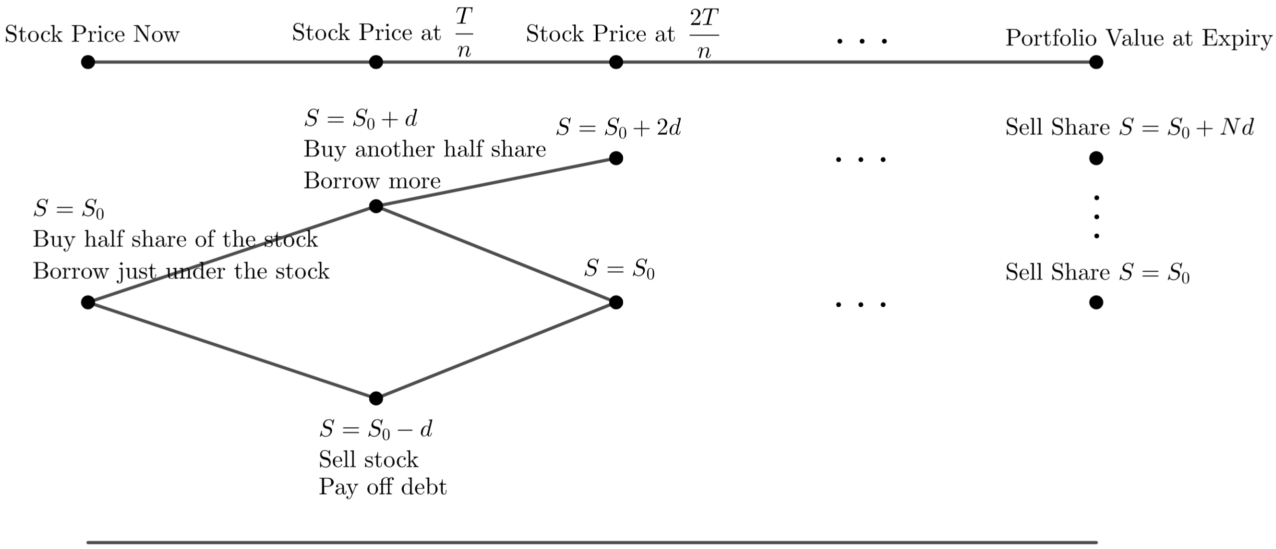

Figure 2: Binomial Option Pricing Decision Tree Diagram

5.2 Further generalisation of the binomial model

Once again we review an assumption of the model, we review the assumption that the stock or asset price will go up or down in value by the same value over a time interval and instead adopt the idea that it will go up or down in accordance with u and d where u can be described as one plus the rate of increase of the stock if the stock goes up an d can be described similarly as one plus the rate of decrease of the stock if the stock goes down, and hence, $0 < d < 1$ and $u > 1$. We also release the assumption that the stock pays no dividends and instead assume that the stock pays dividend, $\delta \geqslant 0$, annually at a continuous rate. We continue to derive the binomial model equations over one period. To this end, first we notice that with δ the end of period share price from point n to point $n+1$ in the binomial tree is, from the synthetic option equations (3) and (4):

$$S_{n+1} = \left\{ \begin{array}{ll} uS_n exp(\delta h), & \text{when stock value increases.} \\ dS_n exp(\delta h), & \text{when stock value decreases.} \end{array} \right\}$$

assuming that the dividends are reinvested into the stock. We define the cost of a European option, C and notice C takes values C_u if stock value goes up and C_d if stock value goes down as described below. From our synthetic option we also recall that $C = hS - b$ at time $t = 0$, where h is the share of the stock owned and b is the borrowing amount. We notice that after each time period T/n we get the linear equations, [Ma (2015)].

$$\left\{ \begin{array}{ll} C_u = huSexp\left(\frac{\delta T}{n}\right) + bexp\left(\frac{rT}{n}\right), & \text{when stock value increases.} \\ \\ C_d = hdSexp\left(\frac{\delta T}{n}\right) + bexp\left(\frac{rT}{n}\right), & \text{when stock value decreases.} \end{array} \right\}$$

where r is the risk-free rate. We know the values of S, δ and r which leaves two unknowns, h and b in two equations, so we can solve for the unknowns, giving us

$$h = exp\left(-\frac{\delta T}{n}\right)\left(\frac{C_u - C_d}{S(u-d)}\right) \quad \text{and} \quad b = exp\left(\frac{-rT}{n}\right)\left(\frac{uC_d - dC_u}{u-d}\right),$$

We can now substitute these into the synthetic option portfolio equations (3) and (4) to get,

$$C = hS + b = exp\left(-\frac{rT}{n}\right)\left(C_u\frac{exp\left(\frac{(r-\delta)T}{n}\right) - d}{u-d} + C_d\frac{u - exp\left(\frac{(r-\delta)T}{n}\right)}{u-d}\right) \tag{5}$$

5.3 Calculating u and d

We can use arbitrage in the binomial tree to calculate values for u and d that fit our assumptions on our up and down factors as follows. Firstly, u and d must follow the relationship:

$$d < exp\left(\frac{(r-\delta)T}{n}\right) < u, \quad \text{and then multiplying } S \text{ yields} \implies dS < Sexp\left(\frac{(r-\delta)T}{n}\right) < uS.$$

From the No Arbitrage Principle we have that the above inequality cannot be violated by the values of u and d. We can set u and d by increasing or decreasing the volatility adjustment to the risk-free return factor to satisfy the equation above. This gives us:

$$u = exp\left(\frac{(r-\delta)T}{n} + \sigma\sqrt{\frac{T}{n}}\right), \quad \text{and} \quad d = exp\left(\frac{(r-\delta)T}{n} - \sigma\sqrt{\frac{T}{n}}\right).$$

5.4 Using the risk-neutral probability

We can calculate C in the binomial model over one period easily by deducing the **Risk-Neutral Probability** formulae. We notice that the in (5) the coefficients of C_u and C_d do not rely on the value of the underlying asset, S and therefore are constant on all values of S. We also notice that the coefficients sum to 1. By the No Arbitrage Principle, u and d must satisfy

$$d < exp\left(\frac{(r-\delta)T}{n}\right) < u \implies \frac{exp\left(\frac{(r-\delta)T}{n}\right) - d}{u-d} > 0, \quad \frac{u - exp\left(\frac{(r-\delta)T}{n}\right)}{u-d} > 0.$$

This means that we can interpret the above as probabilities, so we define

$$p^* = \frac{exp\left(\frac{(r-\delta)T}{n}\right) - d}{u-d}, \quad \text{and} \quad 1-p^* = \frac{u - exp\left(\frac{(r-\delta)T}{n}\right)}{u-d}.$$

Then equation (5) becomes [Ma (2015)],

$$C = exp\left(\frac{-rT}{n}\right)(p^*C_u + (1-p^*)C_d). \tag{6}$$

5.5 American option pricing

American options are similar to European options but unlike European options they can be exercised at any time and the holder of the option can make that profit at any point before the expiry. This means that we need to adapt the binomial method to allow for early exercising of the option which means the cost of the option at previous points is affected as follows:

$$C = \left\{ \begin{array}{ll} \max(\{exp\left(\frac{-rT}{n}\right)[p^*C_u + (1-p^*)C_d], \quad S-E\}), & \text{if option is a call option.} \\ \max(\{exp\left(\frac{-rT}{n}\right)[p^*C_u + (1-p^*)C_d], \quad E-S\}), & \text{if option is a put option.} \end{array} \right\}$$

6 Binomial Model Application

6.1 Iterative algorithm for European and American options

We can now apply the above model iteratively working back from the final node (N-th node) of the binomial tree to calculate the initial cost of the option. We start by defining the set on the set of n-th nodes and its function:

$$S_n = \{Su^{N-n}d^n : n \in \{0,1,...,N\}\}, \text{ and } f(S_n, i) \text{ is defined as the } i\text{-th member of } S_n.$$

We let C_n be the set of values of the cost of the option at each of the n-th points along the binomial tree and calculate the cost of the option at expiry using equations (1) and (2):

$$C_N = \left\{ \begin{cases} S_i - E, & \text{if option is a call option,} \\ E - S_i, & \text{if option is a put option.} \end{cases} : i \in \{0, 1, ..., n\} \right\}$$

We then define the useful function, allowing us to iterate back to the initial cost of the option:

$$f(C_n, i) \text{ is defined as the } i\text{-th member of } C_n.$$

Then we can define $C_{i,n}$ in order to calculate C_{n-1} as follows:

$$C_{i,n} = \left\{ \begin{array}{lr} exp\left(\frac{-rT}{N}\right)(p^* f(C_n, i) + (1-p^*) f(C_n, i+1)), & \text{if a European option.} \\ \max(\{exp\left(\frac{-rT}{n}\right)[p^* f(C_n, i) + (1-p^*) f(C_n, i+1)], S - E\}), & \text{if an American call option.} \\ \max(\{exp\left(\frac{-rT}{n}\right)[p^* f(C_n, i) + (1-p^*) f(C_n, i+1)], E - S\}), & \text{if an American put option.} \end{array} \right\}$$

From the above we have that,

$$C_{n-1} = \{C_{i,n} : i \in \{0, 1, ..., n-1\}\}.$$

We continue this process iteratively until we have calculated C_0 which is the cost of the option at time $t = 0$ and therefore the cost to buy the option. Note that it is important for American options to be run with large N so that it can value the American option at as many points as possible due to its ability to be exercised early.

6.2 Pricing Other Options

There are some other types of including options called **binaries**, which take one of two values at expiry. Although binaries are temporarily banned in the UK due to the effect that unregulated trading can have across the whole market [Leibenluft (2008)], and their trading has diminished, they are still traded in other parts of the world and are still important assets to consider. There are other common options which are comprised of calls and puts. Some of the more common binaries and options are as follows, described simply by their payoff functions, pf:

1. **Cash-Or-Nothing (Call)**: $pf(S) = BH(S - E)$ where B is the cash amount decided upon writing the option, and H is the heaviside step function. This means that $pf(S) = B$ if $S > E$ and $pf(S) = 0$ otherwise.

2. **Bullish-Vertical-Spread**: $pf(S) = \max(\{S - E_1, 0\}) - \max(\{S - E_2, 0\})$ where E_1, E_2 are two strike prices on the same asset.

3. **Supershare**: $pf(S) = \frac{1}{d}(H(S - E) - H(S - E - d))$ where $d \in \mathbb{R}^+$ is a parameter set on writing the option.

4. **Asset-Or-Nothing (Call)**: $pf(S) = SH(S - E)$.

5. **Straddle**: $pf(S) = \max(\{S - E, 0\}) + \max(\{E - S, 0\})$.

They are very simple to price for European style options as we can just change the algorithm as follows:

$$C_N = \left\{ \begin{array}{lr} BH(S_i - E), & \text{if Cash-or-Nothing Call,} \\ BH(E - S_i), & \text{if Cash-or-Nothing Put,} \\ \max(\{S_i - E_1, 0\}) - \max(\{S_i - E_2, 0\}), & \text{if Bullish-Vertical-Spread,} \\ \frac{1}{d}(H(S_i - E) - H(S - E - d)), & \text{if option Supershare,} \\ SH(S_i - E), & \text{if Asset-or-Nothing Call,} \\ SH(E - S_i), & \text{if Asset-or-Nothing Put,} \\ \max(\{S_i - E, 0\} + \max(\{E - S, 0\})), & \text{if Straddle.} \end{array} : i \in \{0, 1, ..., n\} \right\}$$

Then, after calculating the possible values for the cost of the option at the final node we repeat the same steps as the algorithm and continue back to the cost at the first node.
The above algorithms are programmed in python and stored in my Github repo [Langdown (2020)].

7 The Black-Scholes Option Pricing Model

7.1 The Black-Scholes equation and its analytical solutions

The **Black-Scholes** option pricing model can be stated as the equation below, from [Wilmott et al. (1995)]:

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

The **Black-Scholes Equation** (BSE) is a partial differential equation in $V(S,t)$, the value of a European option. Where σ is the volatility of the stock or asset, r is the risk-free rate, $\delta \geqslant 0$ is the dividend payment paid at a continuous rate and S is the value of the stock or asset at the start of the option and E is strike price of the option. The equation can be solved analytically when r, σ and δ are constants as a function of time. Appropriate boundary and initial conditions can be used to solve the equation for $V(S,t)$, for European call and put options. We are still under the assumptions of the No Arbitrage Principle. It yields, [Wilmott et al. (1995)]:

$$V(S,t) = \left\{ \begin{array}{ll} C(S,t) = Sexp\left(-\delta(T-t)\right)N(d_1) - Eexp\left(-r(T-t)\right)N(d_2), & \text{if call option,} \\ P(S,t) = Eexp\left(-r(T-t)\right)N(-d_2) - Sexp\left(-\delta(T-t)\right)N(-d_1), & \text{if put option,} \end{array} \right\}$$

where N is the cumulative distribution function for a standard normal distribution and T is the expiry date of the option, and, d_1, d_2 are given by

$$d_1 = \frac{ln(\frac{S}{E}) + (r - \delta + \frac{\sigma^2}{2})(T-t)}{\sigma\sqrt{T-t}} \quad \text{and,} \quad d_2 = d_1 - \sigma\sqrt{T-t}.$$

We can derive solutions for the values of some binary options similarly,

$$V(S,t) = \left\{ \begin{array}{ll} C(S,t) = Bexp\left(-r(T-t)\right)N(d_2), & \text{if option is a Cash-or-Nothing call,} \\ C(S,t) = Sexp\left(-\delta(T-t)\right)N(d_1), & \text{if option is an Asset-or-Nothing call,} \\ P(S,t) = Bexp\left(-r(T-t)\right)N(-d_2), & \text{if option is a Cash-or-Nothing put,} \\ P(S,t) = Sexp\left(-\delta(T-t)\right)N(-d_1), & \text{if option is an Asset-or-Nothing put.} \end{array} \right\}$$

7.2 The Black-Scholes Model and Other Option Types

The Black-Scholes Model is designed for European style options and is able to find analytical solutions for those option types. However, it is not designed to find analytical solutions for American style options due to the uncertainty that they can be exercised at any time. But, it is generally considered unwise to exercise American style options before their expiry, in particular when the underlying asset has low volatility and is unlikely to drastically change. For these reasons it is considered reasonable to price American options using the BSE when the interest rates and dividends are low. High volatility assets when paired with high risk-free interest rates and high dividends should not be priced using the BSE.

8 Finite Differences Method

8.1 The heat equation

In the previous section it was shown that the BSE has integral solutions for European Vanilla options. However, American options' integral solutions do not exist. However, we can show the BSE has its own equivalent heat equation and use finite difference methods to acquire solutions for American options. The heat equation (in arbitrary variables u, τ, x) is given by:

$$\frac{\partial^2 u}{\partial x^2} = \frac{\partial u}{\partial \tau}.$$

It can be shown through the substitution of variables, $S = Ee^x$ and $t = \frac{T-2\tau}{\sigma^2}$, that the BSE has its own heat equation [Wilmott et al. (1995)] which we can then solve with finite difference methods.

8.2 Using the Black-Scholes heat equation

We discretise the Black-Scholes heat equation similarly to the way that Dr Leppinen has in his notes. We use the method of finite differences to calculate $u(x, \tau)$ subject to initial and boundary conditions. Hence, we create a numerical grid by discretising the domain of the BSE. The BSE has domain $0 \leqslant S \leqslant \infty$ and $0 \leqslant t \leqslant T$, from the fact that $S = Ee^x$ and $t = \frac{T-2\tau}{\sigma^2}$ we have that the new variables have domain $-\infty \leqslant x \leqslant \infty$ and $0 \leqslant \tau \leqslant \frac{\sigma^2 T}{2}$. In the interest of using finite differences we truncate the x domain using a large positive and a large negative number, x^+ and x^-. Now we finalise the discretisation by choosing a grid size, dx and a two relevant numbers, a positive and a negative, N^+ and N^- respectively so that the x domain becomes $\{x_i = idx : N^- \leqslant i \leqslant N^+\}$. To discretise the τ domain we repeat the process and get $\{\tau_j = jd\tau : 0 \leqslant j \leqslant \frac{\sigma^2 T}{2d\tau}\}$. To summarise, our discretised approximation of $u(x, \tau)$ is $u_{i,j} = u(x_i, \tau_j)$.
There are two important conditions that we can use to navigate the numerical grid. The first is an initial condition:

$$u(i, 0) = \max\left(\left\{exp\left(\frac{(k+1)idx}{2}\right) - exp\left(\frac{(k-1)idx}{2}\right), 0\right\}\right).$$

The other is the boundary conditions at $x = x^-$ or $x = x^+$:

$$u(x^-, \tau_j) = \left\{ \begin{array}{ll} 0, & \text{if option is a European call option,} \\ exp\left(\frac{(k+1)x}{2} + \frac{(k+1)^2 jd\tau}{4}\right), & \text{if option is a European put option,} \end{array} \right\}, \quad \text{and}$$

$$u(x^+, \tau_j) = \left\{ \begin{array}{ll} exp\left(\frac{(k+1)x}{2} + \frac{(k+1)^2 jd\tau}{4}\right), & \text{if option is a European call option,} \\ 0, & \text{if option is a European put option.} \end{array} \right\}$$

The numerical grid can be solved explicitly and implicitly, below are brief descriptions of the explicit and implicit methods.

8.3 American Options' Initial and Boundary Conditions

We let $g(x, \tau)$ be the payoff function for an American option, it is similar to the European payoff function but scaled in τ. $g(x, \tau)$ takes either of two values depending on the type of option:

$$g(x, \tau) = \left\{ \begin{array}{ll} exp\left(\frac{(k+1)^2\tau}{4}\right) max\left(exp\left(\frac{(k+1)x}{2}\right) - exp\left(\frac{(k-1)x}{2}\right), 0\right), & \text{if American call option,} \\ exp\left(\frac{(k+1)^2\tau}{4}\right) max\left(exp\left(\frac{(k-1)x}{2}\right) - exp\left(\frac{(k+1)x}{2}\right), 0\right), & \text{if American put option.} \end{array} \right\}$$

For the new scaled payoff function, the scaled boundary conditions at x^- and x^+ are given by: $u \to g$ as $x \to \pm\infty$, under the assumption that u, $\frac{\partial u}{\partial x}$ are both continuous when $\tau > 0$. We notice that the initial conditions remain the same, since:

$$g(x,0) = \max\left(exp\left(\frac{(k \pm 1)x}{2}\right) - exp\left(\frac{(k \mp 1)x}{2}\right), 0\right) = u(x,0)$$

Since the option is American we require that $u(x,\tau) \geqslant g(x,\tau)\ \forall x, \tau$. We ensure that this is true by performing a maximisation after each step:

$$u(i,j) = \max\{u(i,j), g(i,j)\}\ \forall i,j.$$

8.4 Applying finite difference methods to the grid

- Using explicit finite difference methods gives us the following partial derivatives:

$$\left.\frac{\partial u}{\partial \tau}\right|_{x_i,\tau_j} = \frac{u_{i,j+1} - u_{i,j}}{d\tau} + O(d\tau) \quad \text{and} \quad \left.\frac{\partial^2 u}{\partial x^2}\right|_{x_i,\tau_{j+1}} = \frac{u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1}}{dx^2} + O(dx^2)$$

  We substitute these into the heat equation to find an iterative relation [Duffy (2006)]:

$$u_{i,j+1} = \frac{\partial \tau}{\partial x^2}\left(u_{i-1,j} + u_{i+1,j}\right) + \left(1 - 2\frac{\partial \tau}{\partial x^2}\right) u_{i,j}. \tag{7}$$

  It is important to note that the numerical method above is numerically stable only for values satisfying the time-stepping constraint, $d\tau < \frac{dx^2}{2}$ Duffy (2006).

- Using implicit finite difference methods gives us the following partial derivatives:

$$\left.\frac{\partial u}{\partial \tau}\right|_{x_i,\tau_{j+1}} = \frac{u_{i,j+1} - u_{i,j}}{d\tau} + O(d\tau), \quad \text{and} \quad \left.\frac{\partial^2 u}{\partial x^2}\right|_{x_i,\tau_j} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{dx^2} + O(dx^2).$$

  Once again substituting these into the heat equation we get a difference iterative relation [Ran (2010)]:

$$u_{i,j} = \left(1 + 2\frac{d\tau}{dx^2}\right) u_{i,j+1} - \frac{d\tau}{dx^2}(u_{i-1,j+1} + u_{i+1,j+1}). \tag{8}$$

  Unlike the explicit method, the implicit method has no time-stepping restriction, but it has twice as many computations compared to the explicit method. Below is a diagram of a geometrical interpretation of how the explicit and implicit methods can be used to calculate new points on the grid.

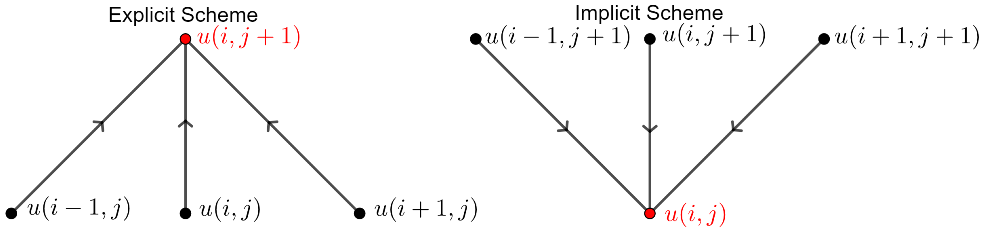

Figure 3: Diagrams of the Finite Difference Scheme's Geometrical Interpretations

- Crank-Nicolson Method can be written in terms of explicit and implicit methods: Crank-Nicolson is defined as the average of the explicit and implicit methods and therefore, in the heat equation it can described iteratively using the implicit and explicit iterative equations, (7) and (8):

$$u_{i,j+1} = \frac{1}{2}\left(\alpha(u_{i-1,j} + u_{i+1,j}) + (1-2\alpha)u_{i,j}\right) + \frac{1}{2}\left(-\alpha(u_{i-1,j+1} + u_{i+1,j+1}) - u_{i,j}\right)$$

$$\implies u_{i,j+1} = \frac{\alpha}{2}(u_{i-1,j} + u_{i+1,j} - u_{i-1,j+1} - ui + 1, j + 1 + 2u_{i,j})$$

  where $\alpha = \frac{d\tau}{dx^2}$. This can be solved using matrix manipulation or found by doing both the explicit and the implicit methods and calculating their average at each iteration. Despite being computationally expensive and also its need to satisfy the time-stepping constraint of the explicit method, Crank-Nicolson converges much faster and can produce much more accurate values at lower step numbers [Consulting (2012)].

9 Finite Difference Methods Application

9.1 Explicit iterative algorithm

We apply the above methods, creating a numerical grid, applying boundary conditions and using explicit iterative methods to calculate the value of a European vanilla option. We start by defining a set S, so, we let S_j be the set of values of $u(x_i, \tau_j)$ for a fixed j. To simplify the algorithm we define some variables according to a step size decided at run time, n:

$$dx = \frac{ln\left(\frac{S}{E}\right)}{n}, \quad d\tau = \left\{\min a, \text{ s.t } \frac{\sigma^2 T}{2na} < \frac{dx^2}{2}\right\}, \quad x^- = ln\left(\frac{S}{E}\right) - 100dx, \quad \text{and } x^+ = ln\left(\frac{S}{E}\right) + 100dx.$$

We do this so that the grid is symmetrical and so that the grid is positioned central to the solution and accurate around the values of the solution whilst satisfying the time stepping-constraint in the least computationally intensive way by choosing a minimum $d\tau$. Subsequently we define S_j which will be used iteratively to find the line of values of which the solution lies on:

$$S_j = \{u(x_i, \tau_j) : i \in \{0, ..., 200\}\}, \text{ with initial conditions} \quad S_0 = \{u(x_i, 0) : i \in \{0, ..., 200\}\}.$$

Now we define the iterative set, S_{j+1}:

$$S_{j+1} = \{u(x^-, \tau_{j+1}), \frac{\partial \tau}{\partial x^2}(u_{i-1,j} + u_{i+1,j}) + \left(1 - 2\frac{\partial \tau}{\partial x^2}\right) u_{i,j} \ \forall i \in \{1, ..., 199\}, u(x^+, \tau_{j+1})\},$$

where the first and final members of the set are given by the boundary conditions and those lying between are given by the explicit method. We continue to find S_{j+1} until we arrive at S_J, where $J = \frac{\tau_{max}}{d\tau}$. Then, since our numerical grid is symmetrical in x and i we can take the middle-most value of the set to be the value of $u\left(ln\left(\frac{S}{E}\right), \tau_{max}\right)$, which can then be used to find V.

9.2 Implicit iterative algorithm

We apply the above method, using a numerical grid with boundary and initial conditions but this time we employ linear algebra methods to solve a system. Once again, to simplify the algorithm we define variables according to step size n:

$$dx = \frac{ln\left(\frac{S}{E}\right)}{n}, \quad d\tau = \frac{\sigma^2 T}{2n}, \quad x^- = ln\left(\frac{S}{E}\right) - 100dx, \quad x^+ = ln\left(\frac{S}{E}\right) + 100dx.$$

We change the value of $d\tau$ since we no longer need to abide by time-stepping constraints. The other variables are described similarly to the first problem for the same reason, symmetry and centralisation of the numerical grid about the solution. We start by defining a vector, $\mathbf{u}$, [Ran (2010)]:

$$\mathbf{u}_j = [u(x^-, \tau_j), u(x^- + dx, \tau_j), ..., u(0, \tau_j), ..., u(x^+ - dx, \tau_j), (u(x^+, \tau_j)].$$

Then we can see from our explicit scheme that we have $\mathbf{Au} = \mathbf{b}$, where $\mathbf{A}$ is given by the implicit iterative rule:

$$\mathbf{A} = \begin{pmatrix} 1+2\alpha & 1 & 0 & 0 & 0 & \cdots & 0 \\ -\alpha & 1+2\alpha & -\alpha & 0 & 0 & \cdots & 0 \\ 0 & -\alpha & 1+2\alpha & \alpha & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & -\alpha & 1+2\alpha & \alpha & 0 \\ 0 & \cdots & 0 & 0 & -\alpha & 1+2\alpha & -\alpha \\ 0 & \cdots & 0 & 0 & 0 & 1 & 1+2\alpha \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} u(x^-, \tau_{j+1}) \\ u(x^- + dx, \tau_j) + \alpha u(x^-, \tau_{j+1}) \\ u(x^- + 2dx, \tau_j) \\ \vdots \\ u(x^+ - 2dx, \tau_j) \\ u(x^+ - dx, \tau_j) + \alpha u(x^+, \tau_{j+1}) \\ u(x^+, \tau_{j+1}) \end{pmatrix}$$

Now, we use tridiagonal matrix solving methods to find a solution to $\mathbf{u}_{j+1}$ in an iterative way until we reach $\mathbf{u}_J$, where $J = j_{max}$. Then, the value of the middle-most value in the set is $u\left(ln\left(\frac{S}{E}\right), \tau_{max}\right)$, which can then be used to find V.

9.3 Crank-Nicolson iterative algorithm

Once again we apply the above methods. However, due to its value being the average of the explicit and implicit method [Consulting (2012)], we are able to calculate both and use them to calculate the Crank-Nicolson solution. We start by calculating boundary and initial conditions for the explicit and implicit methods, then we compute $Eu(i, j+1)$ iteratively as in the explicit method section above and $Iu(i, j+1)$ iteratively as in the implicit method section above. Now we calculate

$$u(i, j+1) = 0.5(Eu(i, j+1) + Iu(i, j+1)),$$

the average of the two values and then use the new $u(i, j+1)$ to calculate $u(i, j+2)$, continuing in this way until we reach $\mathbf{u}_J$, where $J = j_{max}$. Then, the value of the middle-most value in the set is $u\left(ln\left(\frac{S}{E}\right), \tau_{max}\right)$, which can then be used to find V.
The above finite difference algorithms have been programmed in python and are kept in my GitHub repo, [Langdown (2020)].

10 Model Comparisons

For the model analysis we consider time complexity, error convergence and analyse apostiari, which involves analysing the algorithms after running them on a particular system. Space complexity is omitted because the space used is negligible in modern computer systems.

10.1 Time complexity analysis

For the following time complexity analysis I have taken a heuristic approach and plotted $log_2(n)$ versus run time in seconds and compare performance of each method. For the graph plots below I have plotted the binomial method in black, the explicit method in blue, the implicit in red and the C-N method in green. The time readings are taken from running each algorithm on a standard option, all of which have parameters: $S = 40$ (call) and $S = 35$ (put), $E = 35$ (call) and $E = 40$ (put), $r = 0.05$, $\sigma = 0.3$, $T = 3$ and $\delta = 0$.

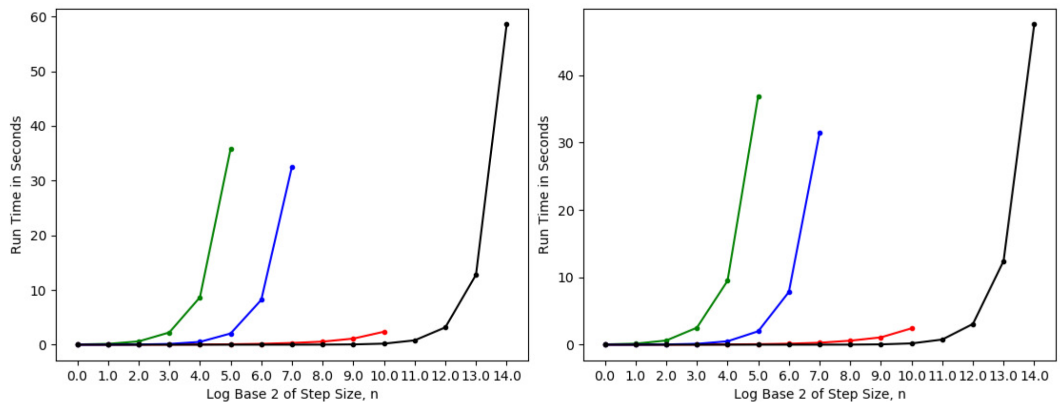

Figure 4: Time graphs of all methods on a European vanilla call (left) and put (right) option.

From figure 4 above, we can see that the binomial method is the fastest method for European options as it only starts to show exponential run-time increase at step size 2^{10}, whereas the finite difference methods start to grow exponentially at lower step sizes, in particular the C-N method starts growing almost instantly. We can also see that the put option pricing takes less time than call option pricing for the binomial method but remains the same for other methods.

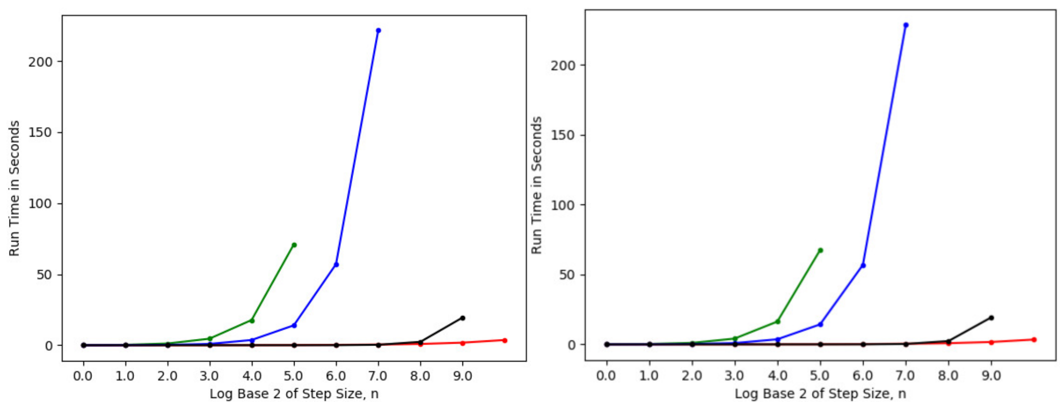

Figure 5: Time graphs of all methods on an American call (left) and put (right) option.

From figure 5 above, we can see that the implicit method is fastest for American option pricing as it only starts to show growth at 2^{10} which is closely followed by the binomial method, the explicit method is much slower in this case and shows much faster growth and then the C-N method is the slowest due to the fact it calculates both explicit and implicit. In this case we can see that for American options the run time is the same for calls and their equivalent puts.

10.2 Error analysis

We recall from section 7.1 that the BSE gives us analytical solutions to European option pricing. We can use those analytical solutions to see if a method converges to the analytical solution when we vary the step size. We do this by plotting $log_2(1/Error)$ against $log_2 n$ where error is simply the distance between the analytic and algorithmic solution. For the graph plots below I have plotted the binomial method in black, the explicit method in blue, the implicit in red and the C-N method in green. Also, I have included three purple lines showing errors of 0.01, 0.001 and 0.0001.
We can see from figure 6 below, that the binomial method has strong linear convergence to the solution for both options. We can also see that for the European call option the explicit

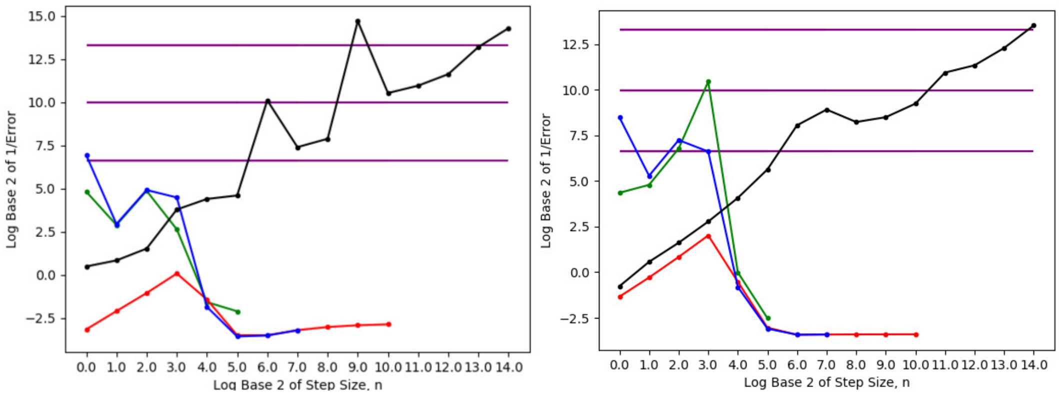

Figure 6: Error graphs of all methods on a European call (left) and put (right) option.

and C-N methods are quite accurate for both methods but especially accurate for European puts. We can see that the implicit method is not very accurate for either options. It is important to note that the observed behaviour where the error starts to deteriorate is where truncation error starts to take over, this is where the values being worked with by the computer as so small that they are considered to be zero causing strange behaviour. Fortunately the binomial method does not work with numbers that get progressively smaller, whereas the finite difference methods split the grid smaller and smaller as we increase step size, so it suffers truncation error at larger step sizes, a way to counter this is to increase the size of the grid so that differences get bigger.

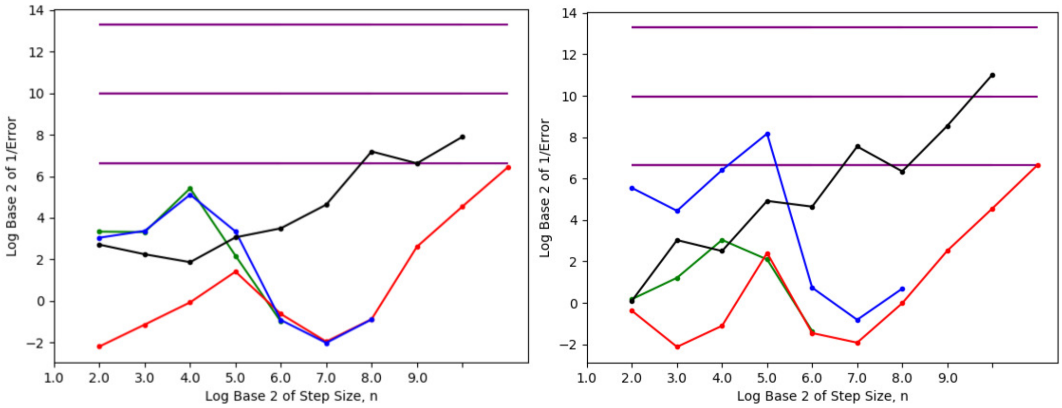

Figure 7: Error graphs of all methods on an American call (left) and put (right) option.

Above, in figure 7 we take error to be the distance between each step size solution to test the convergence to a stable solution as American options do not have analytical solutions. We can see that the binomial method converges linearly to a solution and that the explicit and C-N methods converge to a solution in their regions unaffected by truncation error. However, the explicit method does not converge very steadily in its region unaffected by truncation error. Also we can see that after truncation error the implicit method converges linear to a solution but this is unimportant because previous analysis suggests that this is the wrong solution.

In conclusion, the explicit, C-N and binomial models hold reasonable accuracy before suffering from truncation errors but the implicit model seems to be particularly inaccurate. In terms of the run times the binomial and implicit models are much faster and do not suffer from exponential behaviour until well after the other models' exponential behaviours begin. For American options the binomial method shows strong convergence to a stable solution similarly to its convergence to an analytical solution in European options. This analysis shows

that the binomial method is the strongest method because it can be run at very large step sizes before taking exponentially large amounts of time, does not suffer from truncation error and linearly converges to an analytical solution or a stable solution for American options.
A brief comment on new and developing methods: The models used in this paper are limited, no matter how accurate their numerical solutions converge due to the fact they are derived from the BSE or follow no arbitrage and as such require those unrealistic assumptions. In volatile markets where price is effected more by news stories than statistical methods we must adopt the new strategies. One of the newest models being used in volatile markets is sentiment analysis which involves scouring through social media and analysing text to find a general sentiment towards certain assets in the market to see how they will move in future, good sentiments mean optimism about the assets value. Another method is XVA techniques which take into account credit risk, and capital costs that are from holding capital and deciding how these factors that change with the market can effect the cost of derivatives.

References

Bodie, Z. & Merton, R. C. (1998), *Finance Preliminary Edition*, Prentice Hall.

Consulting, G. (2012), 'Option pricing using the crank-nicolson finite difference method'. Last accessed: 6 February 2020.
URL: *https://www.goddardconsulting.ca/option-pricing-finite-diff-crank-nicolson.html*

Duffy, D. J. (2006), *Finite Difference Methods in Financial Engineering: A Partial Differential Equation Approach*, John Wiley & Sons Ltd.

Knight, J. T. (2014), A Prospect Theory-Based Real Option Analogy for Evaluating Flexible Systems and Architectures in Naval Ship Design, PhD thesis, University of Michigan.

Langdown, M. (2020), 'Github repo nmpfo'.
URL: *https://github.com/Mackenzie-pl/NMPFO*

Leibenluft, J. (2008), 'How can the derivatives market be worth more than the world's total financial assets?'. Last accessed: 16 February 2020.
URL: *https://slate.com/news-and-politics/2008/10/how-can-the-derivatives-market-be-worth-more-than-the-world-s-total-financial-assets.html*

Ma, D. (2015), 'The binomial option pricing model - part 1 — financial mathematics'. Last accessed: 15 December 2019.
URL: *https://financialmathmodels.wordpress.com/2015/06/04/the-binomial-option-pricing-model-part-1/*

Ran, A. (2010), 'The implicit finite-difference method in matrix terms'. Last accessed: 18 February 2020.
URL: *https://www.cs.vu.nl/~ran/LUmethod.pdf*

Stewart, I. (2012), 'The mathematical equation that caused the banks to crash'. Last accessed: 18 February 2020.
URL:
https://www.theguardian.com/science/2012/feb/12/black-scholes-equation-credit-crunch

Wilmott, P., Howison, S. & Dewynne, J. (1995), *The Mathematics of Financial Derivatives: A Student Introduction*, Cambridge University Press.