



Technische Universiteit Delft
Faculteit Elektrotechniek, Wiskunde en Informatica
Delft Institute of Applied Mathematics

Pricing Barrier Options in Discrete Time

Verslag ten behoeve van het
Delft Institute for Applied Mathematics
als onderdeel ter verkrijging

van de graad van

BACHELOR OF SCIENCE
in
TECHNISCHE WISKUNDE

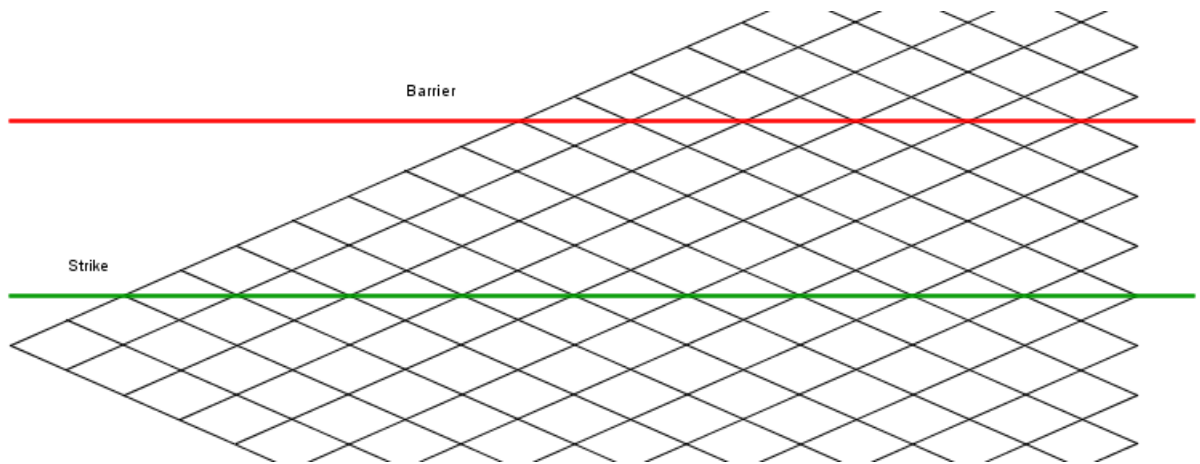
door

TINA SOL

Delft, Nederland
August 2009

Pricing Barrier Options in Discrete Time

BACHELOR THESIS



M.K. Sol (1217933)

Supervisor: Dr. J.A.M. van der Weide

August 21, 2009

Contents

Introduction	1
1 Options	4
1.1 Introduction	4
1.2 Standard options	4
1.3 Barrier options	7
2 Binomial No-Arbitrage Pricing Model	10
2.1 Introduction	10
2.2 One-period binomial model	10
2.3 Multi-period binomial model	14
2.4 Binomial trees	17
2.5 Formula for the option price	19
3 Counting problems	21
3.1 Introduction	21
3.2 Counting problems and price formulae	21
3.3 Reflection principle	22
3.4 Counting paths of barrier options	23
3.5 Counting paths of Parisian options	24
4 Regular barrier option prices	28
4.1 Method I: counting paths	28
4.2 Method II: recursively backwards	29
4.3 Matlab implementation	30
4.4 Results	33
5 Parisian option prices	35
5.1 Counting paths	35
5.2 Matlab implementation	36
5.3 A relation with regular barrier prices?	36
5.4 Results	37
6 Conclusion	40
A Matlab code	41
Bibliography	46

Introduction

During the past year, financial markets have been a topic of discussion more than ever before. A great variety of products is traded on these markets of which stocks and bonds are probably most commonly known. Options belong to a class of derivative products: products whose value is derived from another asset, called the underlying. There are many types of non-standard options for which the label ‘exotic options’ is used. One of these types is called ‘barrier-options’ which - briefly stated - means that an option can become worthless when the price of the underlying hits a barrier.

Many mathematicians have worked on pricing options. In continuous time this has led to the famous formula of Black & Scholes. Finding prices however can also be done in discrete time. This bachelor thesis deals with pricing options and specifically barrier options in discrete time. A special form of barrier options called ‘Parisian options’ will be treated in detail as well. In discrete time, a binomial tree is used to model possible developments of the price of the underlying. By using so-called risk-neutral probabilities it will turn out to be possible to view the option price as an expectation.

To calculate this expectation for standard options it is possible to work backwards through the binomial tree or to calculate expected values at expiration time directly. In the latter case we need to know the probability of occurrence and the option value at final time for every possible final node in the tree. In this context the binomial coefficient is used to calculate the amount of different paths ending in the same node. In the case of barrier options this becomes more complicated but a relatively easy formula that replaces the binomial coefficient can be found. For Parisian options it is not possible to find a direct formula and instead we must use a recursive algorithm.

Besides using combinatorics, the price of a regular barrier option can also be found by working backwards through the binomial tree as long as all value from paths that hit the barrier is set to zero and thereby excluded from the expected value. This has led to an algorithm that is new to both the supervisor and author of this thesis.

The thesis is organized as follows. Chapter 1 gives a general introduction to options. Chapter 2 treats the Binomial No-Arbitrage Pricing model, applies the model to an example in a 1- and multiperiod situation and shows how the option price can either be calculated backwards through the tree or directly. Chapter 3 treats counting problems in the binomial tree and finds expressions needed to formulate the prices of regular barrier and Parisian options as expectations of their value at expiration time. Chapter 4 specifically discusses pricing regular barrier options and discusses numerical results that were obtained. In Chapter 5 the price formula for Parisian options and numerical results of its implementation are discussed.

Chapter 1

Options

1.1 Introduction

Financial options have been traded on the market since 1973 when the Chicago Board Options Exchange (CBOE) was opened. The amount of trade in options has grown enormously during the past few decades. This chapter will introduce basic terminology surrounding options. After that we will move on to a certain type of exotic options. Exotic options are labeled ‘exotic’ because they are more complex and less commonly traded than normal (or so-called ‘vanilla’) options. The type of options that we are interested in are regular barrier options and ‘Parisian options’, a special form of barrier options. Throughout this thesis the term ‘option’ will always refer to a financial option. Non-financial options are called real options; they are often based upon physical assets and form a way to represent possible investments.

Before explaining what options are, here is some finance terminology:

- A *derivative* is a financial instrument that is derived from a certain asset. This means that the value of a derivative is dependent on the price of an *underlying* asset.
- An *asset* is a property or object of value that can be possessed. In this thesis ‘asset’ and ‘underlying’ are synonymously used.
- Options are one of the most important types of derivatives. Other types of derivatives are futures, forwards and swaps.
- Some possible underlying assets of derivatives are commodities, equities (stocks), bonds, interest rates and indices.

1.2 Standard options

Options are a common type of derivatives. They are financial instruments that give the buyer a right, but not an obligation, to engage in a future transaction on some underlying asset.

1.2.1 Basic terminology

The following definitions are important to be familiar with:

- *Holder*: the person buying the option.
- *Writer*: the person selling the option.
- *Exercising*: when the holder enforces the agreement and actually buys from or sells to the option writer.
- *Expiration date*: after this date, the option can no longer be exercised.
- *Call option*: gives its holder a right to buy (and when exercised, gives its writer an obligation to sell).
- *Put option*: gives its holder a right to sell (and when exercised, gives its writer an obligation to buy).
- *American option*: option that can be exercised on any date up to and including the expiration date.
- *European option*: option that can be exercised only on the expiration date.
- *Strike price* (also called exercise price): the price at which the transaction takes place when the option is exercised.
- *At-the-money*: an option is at-the-money when the strike price equals the current price of the underlying asset.
- *In-the-money*: an option is in-the-money when the holder's payoff (see below) is positive if he were to exercise immediately.
- *Out-of-the-money*: when exercised immediately, the option's payoff to its holder would be negative.
- *Volatility*: a measure that puts a number on the tendency of a financial security (the underlying) to fluctuate in value. A high volatility corresponds to a value that tends to fluctuate heavily.

1.2.2 Option payoffs

Associated with owning an option is a certain payoff. Let's take a look at payoffs on the expiration date. When you are the holder of a call option (see Figure 1.1-A) and the market price of the underlying is far above the strike price, exercising your option means you can buy the underlying for the strike price and immediately sell it on the market and receive the current market price. Then your payoff is the difference between these two prices and is positive. On the other hand, a strike price below the market price means that by exercising you pay the strike price and would receive less on the market when you immediately sell the underlying. That means your profit would be negative. Since you are not obliged to exercise, you will not do anything in that case and your payoff will be zero. The last possible scenario is that the strike price and asset price are equal: exercising or not, your payoff will be zero so you are indifferent when it comes to exercising. This is all illustrated in Figure 1.1-A.

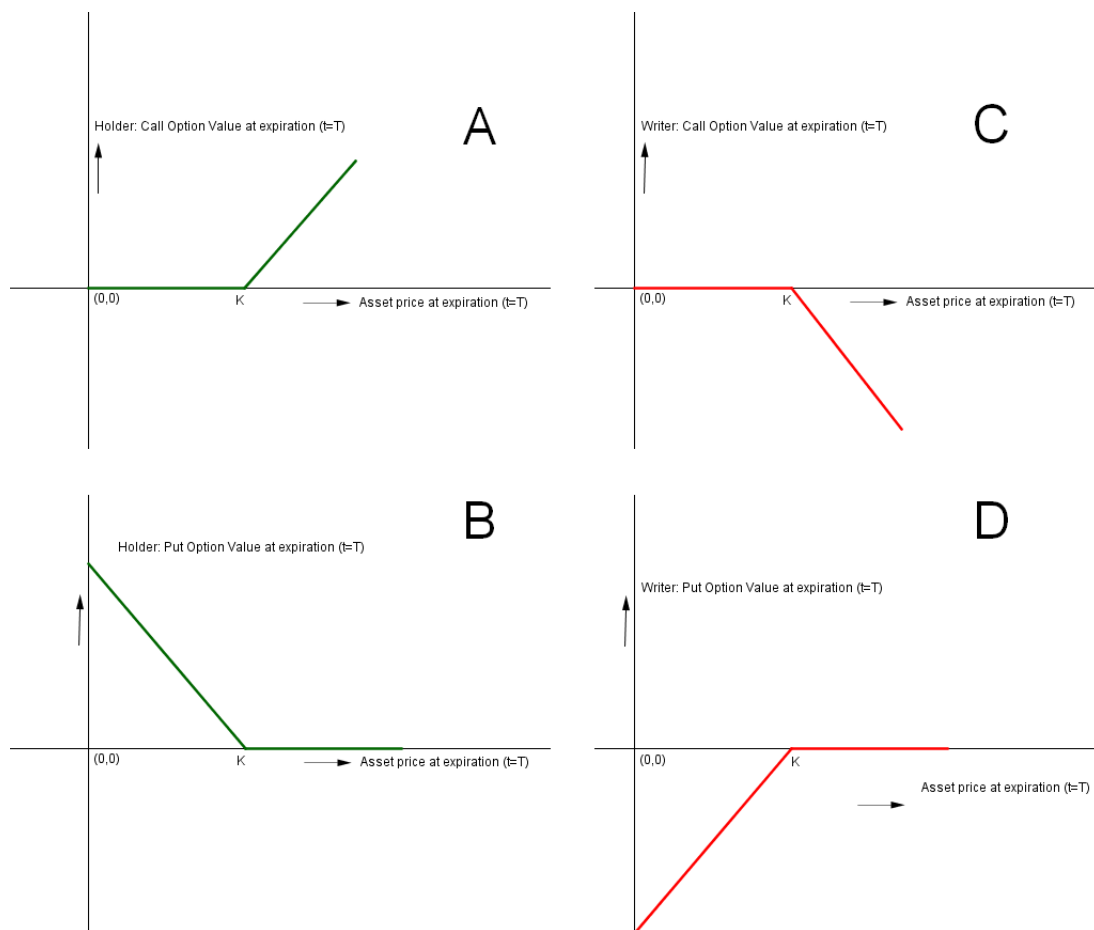


Figure 1.1: Option payoffs at expiration

The other graphs in Figure 1.1 show option payoffs for the holder of a put option (B) and also from the perspective of the option writer instead of the holder (C and D). Take for example graph D. Someone who sells a put option has the obligation to buy the underlying asset when the holder of the option decides to exercise. When the asset price is lower than the strike price, the holder will receive a positive payoff from selling his asset for a higher price than the current market price. In that case, the holder exercises and the writer is obliged to pay him the strike price. Since the asset is worth less on the market, his payoff is negative. On the other hand, when the market price is higher than the strike price, the holder will not exercise the option but rather sell his asset on the market for a higher price. The payoff to the writer is zero in that case.

1.2.3 Option prices

Options are traded before the expiration date, at a time when the payoffs as discussed above are unknown. The holder has to pay a price to obtain an option, received by the writer. (To see profit and loss of the total deal for the holder and writer we would have to shift the payoffs in Figure 1.1 to the holder and writer respectively down and up by the price that the holder paid to the writer.) Now what determines the price at which the option is sold? First of all, the possible payoffs. As we have seen, the payoff is based on the price of the underlying asset and the strike price so these two factors affect the option price. Also the expiration date (time period) has an influence on the price: the further away it is, the more uncertainty there is about where the asset price will end up. Finally, the volatility of the underlying asset price influences the option price. In case of a call option, a high asset price volatility means a significant chance that it ends up very high and similarly a chance that it ends up very low. When it ends up low, you do not lose anything because you do not exercise. When it ends up very high, you make a very large profit. This is called an asymmetric payoff and it causes call options prices to rise if volatility of the underlying rises.

Option pricing is based on several assumptions about the market and about movement of the asset price. Two approaches to calculate prices that are treated in many finance-textbooks are the Binomial Option Pricing Model and the Black-Scholes model. These are mathematically essentially the same but the Binomial Option Pricing Model uses discrete time whereas the Black-Scholes model uses continuous time. In this thesis all pricing will take place in the discrete-time framework. The binomial method is the topic of the next chapter.

1.3 Barrier options

Barrier options are a type of exotic options. The option to exercise depends on the underlying asset price crossing a certain barrier. This makes barrier options path-dependent, since their payoff depends on the path that is travelled by the price of the underlying asset. A barrier can be hit from above or from below. The consequence of hitting the barrier can be that the option starts to exist or that it stops existing. We call the latter case ‘early expiration’ because the same happens as on the expiration date: the contract ceases to exist. That means there are four types of barrier options:

- Up-and-out: an option that expires early if the underlying asset price hits a barrier

from below.

- Up-and-in: an option that comes into existence if the underlying asset price hits a barrier from below.
- Down-and-out: an option that expires early if the underlying asset price hits a barrier from above.
- Down-and-in: an option that comes into existence if the underlying asset price hits a barrier from above.

1.3.1 Parisian options

Parisian options are the same as barrier options but instead of hitting the barrier, the asset price must remain on the ‘other’ side of the barrier for a certain amount of time specified in a contract. This means that an asset price can cross a barrier, stay on the other side for some time and move back across the barrier without any consequences.

Just as with the regular barrier options, there are four different types. All four definitions change in the same way. For example, the definition for the up-and-out becomes the following.

- Up-and-out Parisian: an option that expires if the underlying asset remains constantly above a given barrier for a specified amount of time.

Note that ‘constantly’ means that the time on the other side of the barrier must be consecutive. This is illustrated for the up-and-out case in Figure 1.2, where the price moves around the barrier but does not expire until a consecutive period on the other side occurs. Parisian option contracts where the time-interval on the other side of the barrier does not necessarily have to be consecutive do exist but are not treated in this thesis.

To conclude this chapter, some remarks concerning this thesis. First of all, the only options that will be considered are call options. Furthermore, we will only reason from the perspective of the option holder, never from that of the writer. Finally in the case of regular barrier and Parisian options, the only type we consider is ‘up-and-out’. Understanding this case suffices to understand the other cases as well. Also, due to a relationship between call and put prices the calculation of a put price is very easy when the call price is known.

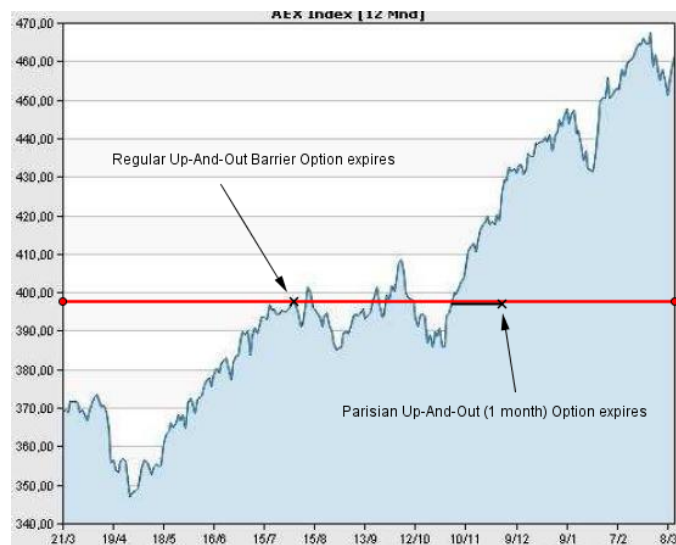


Figure 1.2: Early expiration of barrier options

Chapter 2

Binomial No-Arbitrage Pricing Model

2.1 Introduction

This chapter will treat the so called Binomial No-Arbitrage Pricing Model. Sections 2.2 and 2.3 are based upon [10]. The model leads to a powerful pricing method and is based on the assumption that there are no arbitrage opportunities. The result of this method will turn out to be very useful and is also needed to price methods for exotic options.

The method's approach is to replicate an option by trading in stock and in money markets. We will illustrate this by explaining and applying the method for a one-period binomial tree. Then we will use our results to treat an example. After that, the generalized method for an N -period model will be stated. To illustrate this we extend our 1-period example to 3 periods. In addition we insert a barrier in this 3-period model and calculate a barrier option price. Finally we will construct a non-recursive price formula based on what we have found.

Arbitrage means making a riskless profit. In theoretically perfect and efficient markets there are no arbitrage opportunities. If in reality they do occur, market movements cause them to disappear quickly. Important in this context is the following: two financial instruments that pay off exactly the same (on the expiration date) in any scenario should trade for the same price. It turns out to be possible to construct a portfolio that pays off exactly the same as an option. We know at what prices the elements of the portfolio so we can calculate the price of the portfolio and by the no-arbitrage assumption it equals the price of the option.

Another economical concept that is used in this model and is assumed prior knowledge is the time-value of money. The interest rate quantifies this and is used to so-called 'discount' an amount of money in the future to make it equivalent with an amount of money right now. For more explanation on this and on arbitrage, see [3].

2.2 One-period binomial model

We use the following notations:

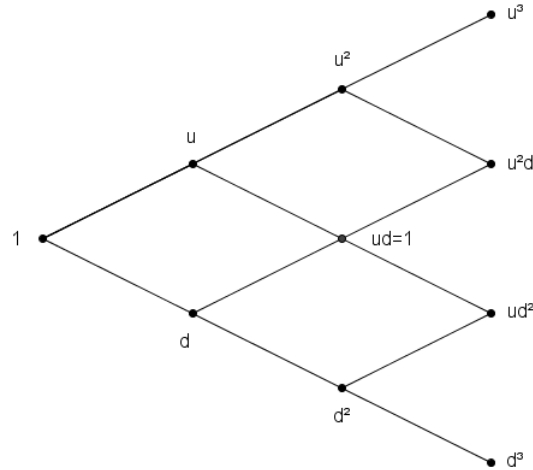


Figure 2.1: Multiplicative simple random walk

S_n	asset price at time n
H, T	heads = asset price goes up, tails = asset price goes down
$p = P(H)$	probability of heads
$q = 1 - p = P(T)$	probability of tails
$u = \frac{S_1(H)}{S_0}$	up factor
$d = \frac{S_1(T)}{S_0}$	down factor
r	interest rate
K	strike price
X_n	wealth at time n
Δ_n	position in the asset during $[n, n + 1)$
V_n	value of option at time n

Furthermore, we have the following:

- $d = \frac{1}{u}$. This is not necessary for the method to work, but a convenient and common choice for implementation. Making one up-step and one down-step ($u \cdot d = u \cdot \frac{1}{u} = 1$) gets you back at the same level you started. This means that the tree that models the asset price is a multiplicative simple random walk. Figure 2.1 illustrates this.
- $0 < d < 1 + r < u$. This condition is needed to rule out arbitrage. The interest rate r is the risk-less return you can get on money: when you have a savings account you will receive this interest without incurring any risk. It is also the interest rate that you have to pay when you borrow money from the bank. Now imagine $d > 1 + r$. In that case you could borrow money, buy stock, later on sell your stock and pay off your loan. This way one would make a profit, without enduring any risk. That situation is impossible so it is ruled out by this condition. Similar reasoning the other way around (selling stock instead of buying it) leads to the other part of the inequality, $1 + r < u$. All elements are greater than zero since stock prices must remain positive.
- $S_1(T) < K < S_1(H)$. We consider this case since it is the only interesting case. It

means the option will be worth zero in the case of $S_1(T)$ and will be worth $S_1 - K$ in the case of $S_1(H)$.

- $V_1 = (S_1 - K)^+ = \max\{0; S_1 - K\}$. This is the option payoff at time n .

The following can happen in a one-period situation:

$$\begin{array}{ccc} & & S_1(H) = uS_0 \\ & \nearrow & \\ S_0 & & \\ & \searrow & \\ & & S_1(T) = dS_0 \end{array}$$

To replicate an option, use initial wealth X_0 to buy Δ_0 stock. This will cost $\Delta_0 S_0$. You will need to borrow $\Delta_0 S_0 - X_0$ in case $X_0 < \Delta_0 S_0$ or you will have some savings left in case $X_0 > \Delta_0 S_0$. In either case, your cash position can be denoted by $X_0 - \Delta_0 S_0$. At time $t = 1$, the interest rate comes into the picture and your cash position will be given by $(1 + r)(X_0 - \Delta_0 S_0)$. The stock you own will then be worth $\Delta_0 S_1$. So your wealth at $t = 1$ is given by:

$$\begin{aligned} X_1 &= \Delta_0 S_1 + (1 + r)(X_0 - \Delta_0 S_0) \\ &= (1 + r)X_0 + \Delta_0(S_1 - (1 + r)S_0) \end{aligned}$$

In the case of no-arbitrage and to make sure that this portfolio is equivalent to the option, it must be so that $X_1(H) = V_1(H)$ and $X_1(T) = V_1(T)$. Since we know S_1 and K we know $V_1(H)$ and $V_1(T)$. (Note that we do *not* know V_1 since we do not know whether H or T occurs.) Using $V_1(H)$ and $V_1(T)$, we get:

$$X_1(H) = V_1(H) = (1 + r)X_0 + \Delta_0(S_1(H) - (1 + r)S_0)$$

$$\frac{1}{1 + r}V_1(H) = X_0 + \Delta_0 \left(\frac{S_1(H)}{1 + r} - S_0 \right) \quad (2.1)$$

Similarly,

$$\frac{1}{1 + r}V_1(T) = X_0 + \Delta_0 \left(\frac{S_1(T)}{1 + r} - S_0 \right) \quad (2.2)$$

To solve these equations an uncommon method is chosen. Two variables are introduced: $\tilde{p}$ and $\tilde{q} = 1 - \tilde{p}$. We can regard them as synthetic probabilities and we will see that they turn out to be very useful. Multiplying the equation for H by $\tilde{p}$, the equation for T by $\tilde{q} = 1 - \tilde{p}$ and adding gives:

$$\begin{aligned} \frac{1}{1 + r}(\tilde{p}V_1(H) + \tilde{q}V_1(T)) &= \tilde{p}X_0 + \tilde{p}\Delta_0 \frac{S_1(H)}{1 + r} - \tilde{p}\Delta_0 S_0 \\ &+ \tilde{q}X_0 + \tilde{q}\Delta_0 \frac{S_1(T)}{1 + r} - \tilde{q}\Delta_0 S_0 \\ &= X_0 - \Delta_0 S_0 + \frac{\Delta_0}{1 + r}[\tilde{p}S_1(H) + \tilde{q}S_1(T)] \end{aligned}$$

If we choose $\tilde{p}$ such that

$$S_0 = \frac{1}{1 + r}[\tilde{p}S_1(H) + (1 - \tilde{p})S_1(T)] \quad (2.3)$$

we have

$$X_0 = \frac{1}{1+r} (\tilde{p}V_1(H) + \tilde{q}V_1(T)) \quad (2.4)$$

which means that

$$V_0 = \frac{1}{1+r} (\tilde{p}V_1(H) + \tilde{q}V_1(T))$$

since there are no arbitrage opportunities. We have now used wealth at time zero to replicate a call option. Looking at (2.3), we see that our choice for $\tilde{p}$ results in a formula for V_0 that is just the expectation - under artificial probabilities $\tilde{p}$ and $\tilde{q}$ - of the option value at $t = 1$, discounted back to $t = 0$. It is important to realize that these are not the actual probabilities p and q but it is useful that this expectation form arises due to the choice for $\tilde{p}$ and $\tilde{q}$. Also 2.3 has the form of a discounted expected value and can be solved for $\tilde{p}$:

$$\begin{aligned} (1+r)S_0 &= \tilde{p}uS_0 + (1-\tilde{p})dS_0 \\ 1+r &= \tilde{p}u + (1-\tilde{p})d \end{aligned}$$

So

$$\tilde{p} = \frac{1+r-d}{u-d} \quad (2.5)$$

$$\tilde{q} = 1 - \tilde{p} = \frac{u-1-r}{u-d} \quad (2.6)$$

Finally, we can calculate Δ_0 , the amount of stock we need to buy at time zero to replicate the option. Subtract 2.2 from 2.1 to get:

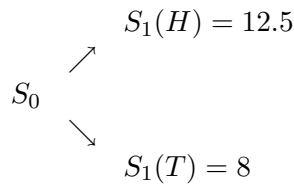
$$\begin{aligned} \frac{1}{1+r}V_1(H) - \frac{1}{1+r}V_1(T) &= X_0 + \Delta_0 \left(\frac{S_1(H)}{1+r} - S_0 \right) \\ &\quad - X_0 - \Delta_0 \left(\frac{S_1(T)}{1+r} - S_0 \right) \\ \frac{1}{1+r}(V_1(H) - V_1(T)) &= \frac{\Delta_0}{1+r}(S_1(H) - S_1(T)) \end{aligned}$$

So

$$\Delta_0 = \frac{V_1(H) - V_1(T)}{S_1(H) - S_1(T)}. \quad (2.7)$$

Example

Now, let's apply these results to an example. We take $u = 1.25$, $d = 1/u = 0.8$. Interest rate $r = 0.05$, initial asset price $S_0 = 10$ and strike price $K = 11$. So $V_1(H) = 1.25 \times 10 - 11 = 1.5$ and $V_1(T) = 0$ (since $0.8 \times 10 = 8 < 11$).



We started off with a formula for X_1 of which we can only fill in r and S_0 at this moment:

$$\begin{aligned} X_1 &= (1+r)X_0 + \Delta_0(S_1 - (1+r)S_0) \\ &= 1.05 \cdot X_0 + \Delta_0(S_1 - 10.5) \end{aligned}$$

The condition we imposed on $\tilde{p}$ led to formulas 2.5 and 2.6 for $\tilde{p}$ and $\tilde{q}$ that we can fill in.

$$\tilde{p} = \frac{1 + 0.05 - 0.8}{1.25 - 0.8} = \frac{5}{9}, \quad \tilde{q} = \frac{1.25 - 1 - 0.05}{1.25 - 0.8} = \frac{4}{9}$$

The next step is filling in 2.4 to obtain V_0 and thereby X_0 :

$$\begin{aligned} X_0 = V_0 &= \frac{1}{1.05} \left(\frac{5}{9} \cdot 1.5 + \frac{4}{9} \cdot 0 \right) \\ &= \frac{20}{21} \cdot \frac{15}{18} = \frac{300}{378} = 0.794 \end{aligned}$$

We also found a formula 2.7 for Δ_0 that we can fill in.

$$\Delta_0 = \frac{1.5 - 0}{12.5 - 8} = \frac{1}{3}.$$

Now, let's fill in the formula we started off with, to check that X_1 equals V_1 .

$$\begin{aligned} X_1(H) &= 1.05 \cdot \frac{300}{378} + \frac{1}{3} (12.5 - 10.5) \\ &= \frac{5}{6} + \frac{2}{3} = 1.5 \\ X_1(T) &= 1.05 \cdot \frac{300}{378} + \frac{1}{3} (8 - 10.5) \\ &= \frac{5}{6} - \frac{5}{6} = 0 \end{aligned}$$

So indeed, $V_1(H)$ and $V_1(T)$ equal $X_1(H)$ and $X_1(T)$. This illustrates that we can calculate an option price by discounting the expected option value at a future time, when we use artificial probabilities. We call these probabilities 'risk-neutral' and denote probability and expectation in the risk-neutral world by $\tilde{\mathbb{P}}$ and $\tilde{\mathbb{E}}$. To see why, note that the formula for V_0 can be written as:

$$V_0 = \frac{1}{1+r} \tilde{\mathbb{E}}[V_1] \rightarrow \tilde{\mathbb{E}}[V_1] = (1+r)V_0$$

This shows that expected return is the interest rate, which is the risk-free return. Therefore $\tilde{\cdot}$ is called the risk-neutral world.

2.3 Multi-period binomial model

The idea of the one-period binomial model can be extended to multiple periods. As we have seen in the example, replicating the option has led to formulas for the risk neutral probabilities $\tilde{p}$ and $\tilde{q}$ that can be filled in straight away. Using these, we can find the value of the option backwards in time by taking the discounted expected value of the possible outcomes.

We can fill in the formula for Δ and calculate our wealth to check the replication. Doing this shows that the value of the portfolio is indeed equal to the value of the option that we calculated by discounting the expected value. The general theorem is as follows (see [10]).

Theorem 2.1: Multi-period binomial model *Consider an n -period binomial asset-pricing model, with $0 < d < 1 + r < u$, and with*

$$\tilde{p} = \frac{1 + r - d}{u - d}, \quad (2.8)$$

$$\tilde{q} = 1 - \tilde{p} = \frac{u - 1 - r}{u - d}. \quad (2.9)$$

Let V_n be a random variable (a derivative that pays off at n) that depends on the coin tosses $\omega_1, \omega_2, \dots, \omega_n$. Define recursively backward in time the sequence $V_{n-1}, V_{n-2}, \dots, V_0$ by

$$V_k(\omega_1 \omega_2 \dots \omega_n) = \frac{1}{1 + r} [\tilde{p} V_{k+1}(\omega_1 \omega_2 \dots \omega_k H) + \tilde{q} V_{k+1}(\omega_1 \omega_2 \dots \omega_k T)]. \quad (2.10)$$

This way, every V_k (except V_n) depends on the coin tosses that have occurred before k . Now, define

$$\Delta_k(\omega_1 \omega_2 \dots \omega_k) = \frac{V_{k+1}(\omega_1 \omega_2 \dots \omega_k H) - V_{k+1}(\omega_1 \omega_2 \dots \omega_k T)}{S_{k+1}(\omega_1 \omega_2 \dots \omega_k H) - S_{k+1}(\omega_1 \omega_2 \dots \omega_k T)}, \quad (2.11)$$

where $0 < k < n - 1$. Now set $X_0 = V_0$ and define recursively forward the portfolio values $X_1, X_2, \dots, X_n$ by

$$X_{k+1} = \Delta_k S_{k+1} + (1 + r)(X_k - \Delta_k S_k), \quad (2.12)$$

where the X_k 's, S_k 's and Δ_k are dependent on $\omega_1 \omega_2 \dots \omega_n$, then it can be proven by induction that the following holds:

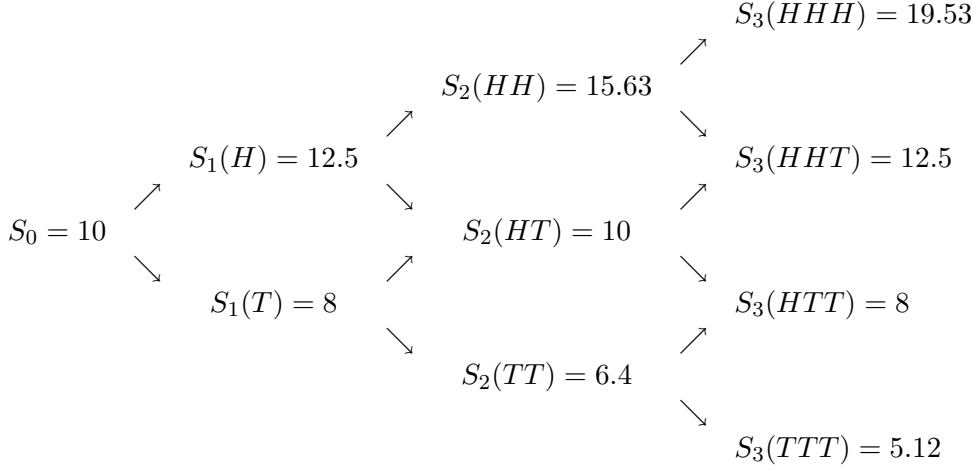
$$V_n(\omega_1 \omega_2 \dots \omega_n) = X_n(\omega_1 \omega_2 \dots \omega_n) \text{ for all } \omega_1 \omega_2 \dots \omega_n \quad (2.13)$$

So for $k = 1, 2, \dots, n$ we define, given $\omega_1 \dots \omega_k$, the price of the derivative security at time k to be $V_k(\omega_1 \dots \omega_k)$ as given in 2.10. At $t = 0$, the price of the derivative is defined to be V_0 .

Example

We will now extend the 1-period example to 3 periods to illustrate the method for multiple periods. We change the strike price K to $K = 7$. According to the general multiperiod method, we should first calculate $\tilde{p}$ and $\tilde{q}$. They are independent of K so they remain $\tilde{p} = \frac{5}{9}$ and $\tilde{q} = \frac{4}{9}$.

The following picture shows the asset-price developments that can occur during these three steps. Note that these are just multiples of u and d with S_0 . So for example $S_3(HHT) = u^2 \cdot d \cdot S_0 = 1.25^2 \cdot 0.8 \cdot 10 = 12.5$. Also $ud = 1$ so after one up and one down step, we are back at the original price level. Thus, the order in which the toss-sequence is stated does not matter, i.e. $S_3(HHT) = S_3(HTH) = S_3(THH)$, $S_2(HT) = S_2(TH)$, etc.



The next step is to define recursively backward in time V_2 , V_1 and V_0 . For this, we need all possible values of V_3 . These are calculated simply by

$$\begin{aligned}
V_3(HHH) &= (S_3(HHH) - K)^+ = 19.53 - 7 = 12.53 \\
V_3(HHT) &= V_3(HTH) = V_3(THH) = (S_3(HHT) - K)^+ = 12.5 - 7 = 5.5 \\
V_3(HTT) &= V_3(THT) = V_3(TTH) = (S_3(HTT) - K)^+ = 8 - 7 = 1 \\
V_3(TTT) &= (S_3(TTT) - K)^+ = 0
\end{aligned}$$

Now, we fill in the formula's for V :

$$V_2(HH) = \frac{1}{1.05} \cdot \left[\frac{5}{9} \cdot V_3(HHH) + \frac{4}{9} \cdot V_3(HHT) \right] = 8.96$$

Analogous for the other V_2 's gives: $V_2(HT) = 3.33$ and $V_2(TT) = 0.53$. Going one step further back, using these V_2 's we get $V_1(H) = \frac{1}{1.05} \cdot \left[\frac{5}{9} \cdot 8.96 + \frac{4}{9} \cdot 3.33 \right] = 6.15$ and in the same way $V_1(T) = 1.99$. Finally, for V_0 we obtain $V_0 = 4.09$. This is the option price that we are interested in. Just as in the 1-period example, by calculating Δ_k 's and X_k 's it can be verified that $V_k = X_k$ holds for every k .

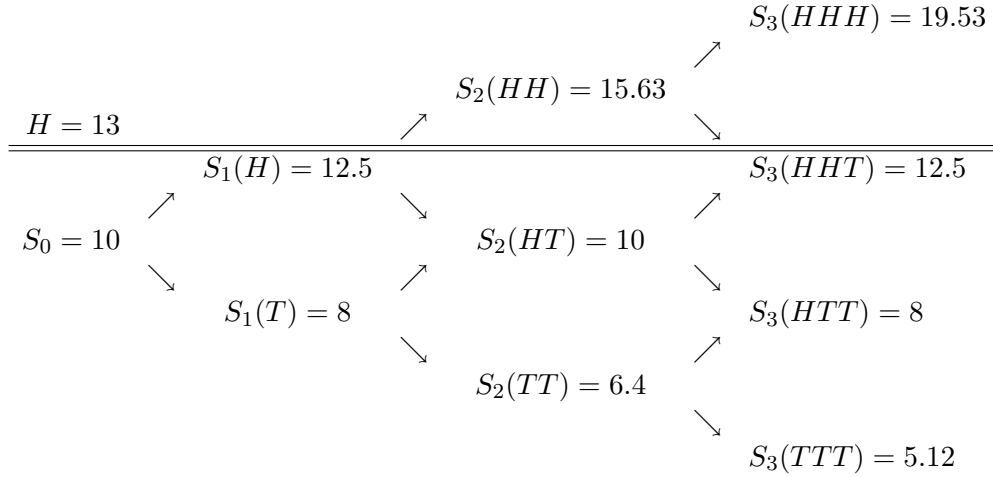
Now we know that the option value is an expected value, we could also calculate it directly instead of working backwards step by step. It would look like this:

$$\begin{aligned}
V_0 &= \frac{1}{(1+r)^3} \tilde{\mathbb{E}}[V_3] \\
&= \frac{1}{(1+r)^3} \left[\tilde{\mathbb{P}}(HHH)V_3(HHH) + \tilde{\mathbb{P}}(HHT)V_3(HHT) \right. \\
&\quad + \tilde{\mathbb{P}}(HTH)V_3(HTH) + \tilde{\mathbb{P}}(THH)V_3(THH) + \tilde{\mathbb{P}}(HTT)V_3(HTT) \\
&\quad \left. + \tilde{\mathbb{P}}(THT)V_3(THT) + \tilde{\mathbb{P}}(TTH)V_3(TTH) + \tilde{\mathbb{P}}(TTT)V_3(TTT) \right] \quad (2.14) \\
&= \frac{1}{1.05^3} [\tilde{p}^3 \cdot 12.53 + \tilde{p}^2 \tilde{q} \cdot 5.5 + \tilde{p}^2 \tilde{q} \cdot 5.5 + \tilde{p}^2 \tilde{q} \cdot 5.5 + \tilde{p} \tilde{q}^2 + \tilde{p} \tilde{q}^2 + \tilde{p} \tilde{q}^2] \\
&= 4.09
\end{aligned}$$

Observe that, for example, $V_3(HHT) = V_3(HTH) = V_3(THH)$ and also $\tilde{\mathbb{P}}_3(TTH) = \tilde{\mathbb{P}}_3(THT) = \tilde{\mathbb{P}}_3(HTT)$. Grouping these terms can simplify calculations. Recall the binomial distribution where the probability of j successes in k trials, with p being the chance

of succes, is given by $\binom{k}{j} p^j q^{k-j}$. If we regard every step as a trial where an up-step is succes, we can denote the probability of making 2 up-steps and 1 down-step by $\binom{3}{2} p^2 q^1$. The binomial coefficient $\binom{3}{2}$ groups $\tilde{\mathbb{P}}_3(TTH)$, $\tilde{\mathbb{P}}_3(THT)$ and $\tilde{\mathbb{P}}_3(HTT)$ together because $\binom{3}{2} = 3$ is the number of different paths that lead to $S_3 = 12.5$. This idea will be generalized in Section 2.5 after we have introduced the binomial tree in more detail.

To conclude this section we will look at what happens when we insert a barrier in the above example. This will serve as an introduction to the counting problems that Chapter 3 deals with.



We insert a barrier at height $H = 13$ so for every path that reaches height HH the option expires early and has zero value. That means we have (superscript B denoting the barrier case)

$$\begin{aligned} V_3^B(HHH) &= 0, V_3^B(HHT) = 0, V_3^B(HTH) = V_3^B(THH) = 5.5 \\ V_3^B(HTT) &= V_3^B(THT) = V_3^B(TTH) = 1, V_3^B(TTT) = 0. \end{aligned}$$

so when we repeat the calculation in 2.14 we can no longer group all V_3 's that have ended in the same node. Filling in 2.14 we get

$$\begin{aligned} V_0^B &= \frac{1}{(1.05)^3} [\tilde{p}^3 \cdot 0 + \tilde{p}^2 \tilde{q} \cdot 0 + \tilde{p}^2 \tilde{q} \cdot 5.5 + \tilde{p}^2 \tilde{q} \cdot 5.5 + \tilde{p} \tilde{q}^2 + \tilde{p} \tilde{q}^2 + \tilde{p} \tilde{q}^2 + 0] \\ &= 1.59 \end{aligned}$$

2.4 Binomial trees

The binomial tree serves to model the underlying asset price. As we have seen, at every time-step the price can either go up or down. Since the up- and down factors u and d are related by $d = 1/u$, we are looking at a simple random walk. Up until now we have worked with sequences of outcomes of coin tosses such as HHT . From now on we will use a different way to express in which node we are, a way that does not tell us the exact path that has been travelled. We introduce the following notations:

(k, i)	position of a node in the tree
k	total number of steps from $(0, 0)$ to a node (k, i)
n	total number of time steps to expiration of the option (from $t = 0$ to $t = T$)
i	height of a node (k, i) , we have that $-k < i < k$
$N(k, i)$	total no. of paths from $(0, 0)$ to (k, i)

Recall from the 3-period example that in order to calculate the expectation, we needed to know every possible payoff at T and for every possible payoff the probability that it occurs. Since (in the regular option case) two paths ending up at the same final value have the same probability to occur, we want to know how many different paths lead to every final value so that we can group these in the summation. Therefore we want to work with $N(k, i)$.

2.4.1 The binomial coefficient

The binomial coefficient is given by:

$$\binom{k}{j} = \frac{k!}{j!(k-j)!} \text{ for } 0 \leq j \leq k$$

In combinatorics, this is often called the ‘choose function’ because it is the number of ways in which you can choose j out of k objects, regardless of the order. Note that it is the same as $\binom{k}{k-j}$:

$$\begin{aligned} \binom{k}{k-j} &= \frac{k!}{(k-j)!(k-(k-j))!} \\ &= \frac{k!}{(k-j)!j!} = \binom{k}{j} \end{aligned}$$

We are looking at paths from $(0, 0)$ to (k, i) and in the case of paths, ‘choosing regardless of order’ means that it does not matter how you get from one node to the other, as long as you get there. Where a path ends only depends on the number of up- and down steps. When you have to end at a certain height you can either choose when all the up-steps occur or when the down steps occur and then the rest of the steps must automatically be of the other type. So to fill in the binomial coefficient we need to know how many up- and down steps we make on the way from one node to another.

2.4.2 Number of paths from $(0, 0)$ to (k, i)

Consider a path from $(0, 0)$ to (k, i) . There are k steps and you end up i points higher than where you started. This means there must be at least i up-steps. That leaves $k - i$ other steps to be taken. Now because we end exactly i nodes higher than we started, there must be as many up- as down-steps among those $k - i$ steps. Of only half of those you may choose freely: $\frac{1}{2}(k - i)$.

So, the number of paths from the origin to a node (k, i) is given by

$$\begin{aligned}
N(k, i) &= \binom{\text{total number of steps}}{\text{steps that can be chosen freely}} \\
&= \binom{\text{total number of steps}}{\text{number of up-steps}} = \binom{k}{i + \frac{1}{2}(k - i)} = \binom{k}{\frac{1}{2}(k + i)} \\
&= \binom{\text{total number of steps}}{\text{number of down-steps}} = \binom{k}{\frac{1}{2}(k - i)}
\end{aligned}$$

Instead of expressing height as i , we can also express it as $2j - k$. This notation reveals j as the total number of up-steps that have been taken. To see this, fill in $i = 2j - k$ in the two formulas above to get

$$N(k, 2j - k) = \binom{k}{j} = \binom{k}{k - j}$$

Throughout the remaining chapters we will use $(k, 2j - k)$ as well as (k, i) , depending on which is more convenient in a situation.

2.4.3 Number of paths from (k_1, i_1) to (k_2, i_2)

When dealing with combinatorial problems for barrier options, we sometimes need to know the number of paths between two arbitrary nodes. The above can easily be generalized for starting in a different node than the origin. We have seen the trick: you need to know the total number of steps and the height difference. The number to choose freely is the height difference + half of the steps that are left.

So in general the number of paths from a point (k_1, i_1) to (k_2, i_2) is given by

$$\binom{k_2 - k_1}{(i_2 - i_1) + \frac{1}{2}((k_2 - k_1) - (i_2 - i_1))} = \binom{k_2 - k_1}{\frac{1}{2}((k_2 - k_1) + (i_2 - i_1))}$$

2.5 Formula for the option price

In the multi-period example we saw that we can calculate the expectation for V_T directly and discount it back from $t = T$ to $t = 0$ instead of working backwards through the tree step by step. From now on we use another expression for the discount factor, as if we were in continuous time. It is called the compounded discount factor and given by $e^{-r\Delta t}$ (see [3] for more explanation).

Let $V_T(n, 2j - n)$ denote the option value at time T when the asset price is at $(n, 2j - n)$ and let $\mathbb{P}(n, 2j - n)$ denote the probability that the asset price ends up in that node. Being in node $(n, 2j - n)$ means the asset price has made j up-steps and $n - j$ down-steps. Note that the possible heights of the final asset price are $n, n - 2, n - 4, \dots, -n + 2, -n$. Then we have

$$\begin{aligned}
V_0 &= e^{-r\Delta t} \tilde{\mathbb{E}}[V_T] \\
&= e^{-rT} \sum_{j=0}^n \tilde{\mathbb{P}}(n, 2j - n) V_T(n, 2j - n)
\end{aligned} \tag{2.15}$$

where

$$\tilde{\mathbb{P}}(n, 2j - n) = N(n, 2j - n) \tilde{p}^j \tilde{q}^{n-j} = \binom{n}{j} \tilde{p}^j \tilde{q}^{n-j} \quad (2.16)$$

and

$$\begin{aligned} V_T(n, 2j - n) &= (S_T(n, 2j - n) - K)^+ \\ &= (S_0 u^j d^{n-j} - K)^+. \end{aligned} \quad (2.17)$$

Finally, combining this into one formula results in

$$V_0 = e^{-rT} \sum_{j=0}^n \binom{n}{j} \tilde{p}^j \tilde{q}^{n-j} (S_0 u^j d^{n-j} - K)^+ \quad (2.18)$$

From now on, we will denote the risk-neutral probabilities no longer by $\tilde{p}$ and $\tilde{q}$ and expectation under these probabilities no longer by $\tilde{\mathbb{E}}$. Rather we assume that we are always in the risk-neutral world so when p and q are used they actually denote risk-neutral probabilities.

Chapter 3

Counting problems

3.1 Introduction

As we have seen in Section 2.3 , we want to count the number of paths of the asset price from $(0,0)$ to certain nodes at final time T . This chapter deals with counting problems that arise when pricing barrier and Parisian options. In the context of binomial trees we add the following notations:

m	the height for which $i \geq m$ lies upon or above the barrier
l	no. of consecutive steps above the barrier before a Parisian option goes out
$F(k, i)$	total no. of paths from $(0,0)$ to (k, i) , not touching or crossing $i = m$
$G(k, i)$	total no. of paths from $(0,0)$ to (k, i) , less than l steps on or above m

Recall that the formula for $N(k, i)$, the number of paths from $(0,0)$ to (k, i) is just the binomial coefficient and is given by

$$N(k, i) = \binom{k}{\frac{1}{2}(k+i)} = \binom{k}{\frac{1}{2}(k-i)} \quad (3.1)$$

or when we call $i = 2j - k$,

$$N(k, 2j - k) = \binom{k}{j} = \binom{k}{k-j}. \quad (3.2)$$

Finding expressions for $F(k, i)$ and $G(k, i)$ is more difficult and needs the ‘reflection principle’, topic of Section 3.3. Section 3.4 is based upon [4] and deals with finding $F(k, i)$. Section 3.5 finds $G(k, i)$ and is based on [5]. But before we go there, we will first take a look at how counting $F(k, i)$ and $G(k, i)$ is related to the option price formula.

3.2 Counting problems and price formulae

Recall from section 2.4 that the price of a regular call option is given by:

$$V_0 = e^{-rT} \sum_{j=0}^n \mathbb{P}(n, 2j - n) V_T(n, 2j - n) \quad (3.3)$$

In the case of a barrier option, working this out gives the same expression for $\mathbb{P}(n, 2j - n)$ but a different expression for $V_T(n, 2j - n)$. (From now on the superscript B will denote the case of a regular barrier option and P a Parisian option.)

$$\begin{aligned} V_T^B(n, 2j - n) &= \mathbf{1}_{\{S_t \neq m \forall t\}} V_T(n, 2j - n) + \mathbf{1}_{\{S_t = m \text{ for some } t\}} \cdot 0 \\ &= \mathbb{P}(\text{S stays below barrier}) (S_T(n, 2j - n) - K)^+ \end{aligned}$$

Every path from $(0, 0)$ to $(n, 2j - n)$ has the same probability to occur. So the probability that a path from $(0, 0)$ to $(n, 2j - n)$ does not touch or cross the barrier is simply

$$\frac{\text{number of paths from } (0, 0) \text{ to } (n, 2j - n) \text{ that stay below the barrier}}{\text{total number of paths from } (0, 0) \text{ to } (n, 2j - n)}$$

which means that

$$\mathbb{P}(\text{S stays below barrier}) = \frac{F(k, i)}{N(k, i)}$$

So the option value results in:

$$V_T^B(n, 2j - n) = \frac{F(k, i)}{N(k, i)} (S_T(n, 2j - n) - K)^+ \quad (3.4)$$

Putting this into the formula for V_0^B gives

$$\begin{aligned} V_0^B &= e^{-rT} \sum_{j=0}^n \mathbb{P}(n, 2j - n) V_T^B(n, 2j - n) \\ &= e^{-rT} \sum_{j=0}^n \binom{n}{j} p^j q^{n-j} \frac{F(n, 2j - n)}{N(n, 2j - n)} (S_T(n, 2j - n) - K)^+ \\ &= e^{-rT} \sum_{j=0}^n N(n, 2j - n) \frac{F(n, 2j - n)}{N(n, 2j - n)} p^j q^{n-j} (S_T(n, 2j - n) - K)^+ \\ &= e^{-rT} \sum_{j=0}^n F(n, 2j - n) p^j q^{n-j} (S_T(n, 2j - n) - K)^+. \end{aligned} \quad (3.5)$$

Doing the same for a Parisian option results in:

$$V_0^P = e^{-rT} \sum_{j=0}^n G(n, 2j - n) p^j q^{n-j} (S_T(n, 2j - n) - K)^+ \quad (3.6)$$

3.3 Reflection principle

To find expressions for $F(n, 2j - n)$ and $G(n, 2j - n)$ in subsequent sections, the reflection principle is needed. It is illustrated in Figure 3.1.

According to the reflection principle [9], there are as many paths from A to B as there are paths from A' to B . To see this, call T the point at which some path from A to B first crosses or touches the x -axis. Since A' and A are each others reflections in the line L , this path from A to T has a reflection in the L -axis: a path from A' to T that has a one-to-one correspondence with the path from A to T . For every path from A to B that

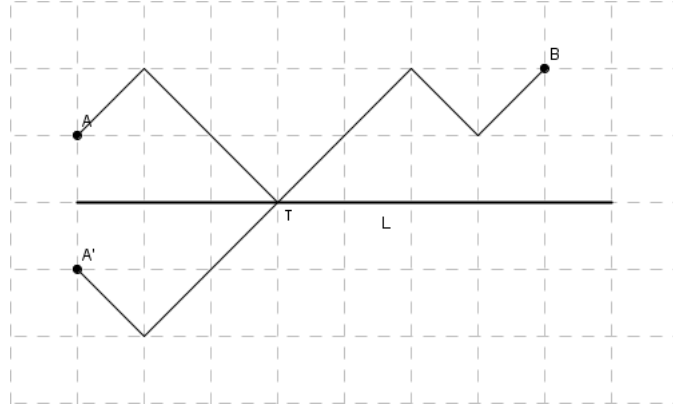


Figure 3.1: Illustration of the reflection principle

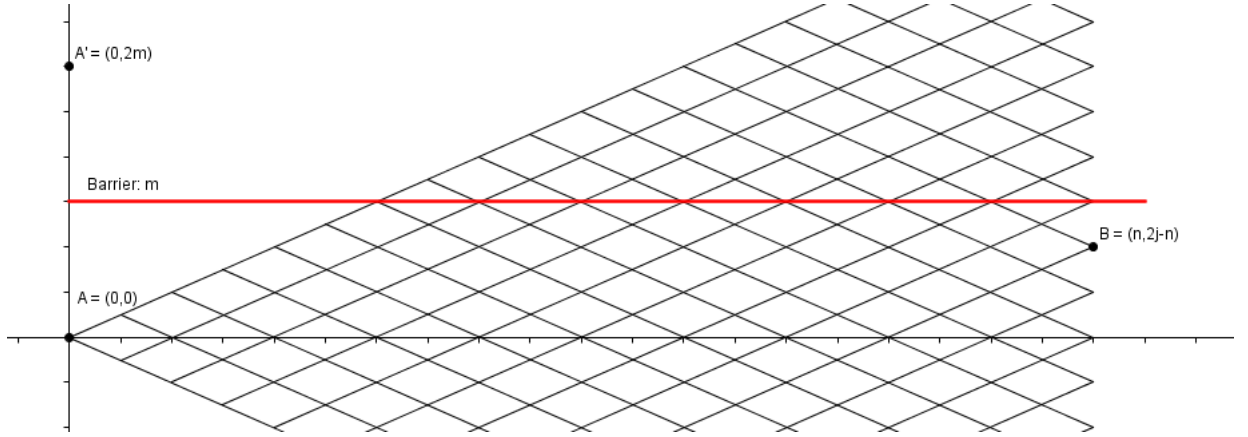


Figure 3.2: Barrier option tree: application of the reflection principle

touches or crosses the barrier, we can find a T as well as a reflected path from A' to T .

That means every path from A to B touching or crossing the barrier corresponds to a path from A' to B . The first part of this corresponding path is the reflection in L from A' to T and the second part is the same for both: T to B . A' lies above and B lies below the barrier, so *every* path from A' to B must cross the barrier. Combine this with the fact that every path from A' to B corresponds to a path from A to B that touches or crosses the barrier and we have our desired result: the total number of paths from A' to B equals the number of paths from A to B that touch or cross barrier L .

3.4 Counting paths of barrier options

This section will apply the reflection principle to count paths in the case of regular barrier options. We are looking for a formula for $F(k, i)$, as defined in 3.1, that we express in the form $F(k, 2j - k)$ here.

From Section 3.3, we know that the number of paths from $(0, 0)$ to $(k, 2j - k)$ that

touches or crosses the barrier equals the total number of paths from $(0, 2m)$ to $(k, 2j - k)$. The total number of paths from one node to another is given by the binomial coefficient. The total number of steps taken is $k - 0 = k$ and the total number of down-steps that is taken is the height difference $2m - (2j - k)$ plus half of the remaining steps $\frac{1}{2}(k - (2m - (2j - k)))$:

$$2m - (2j - k) + \frac{1}{2}(k - (2m - (2j - k))) = 2m - 2j + k - m + j = k + m - j.$$

That means the number of paths we are interested in is given by:

$$\binom{n}{k + m - j} = \binom{k}{k - (k + m - j)} = \binom{k}{j - m}$$

Since this is the number of paths that do touch and/or cross the barrier, we subtract it from the total number of paths to get an expression for $F(k, 2j - k)$:

$$F(k, 2j - k) = \binom{k}{j} - \binom{k}{j - m} \quad (3.7)$$

3.5 Counting paths of Parisian options

Obtaining an expression for $G(k, i)$ is more complicated. It is not possible to find a direct formula for the number of allowed paths to a final node. By dividing the binomial tree into four different areas, we will be able to find a recursive formula. Figure 3.3 shows the different areas.

Area I

First, consider area I. In this area, $k < l + m$ so $i < l + m$. Only after m steps, the price can be at height m and only after another l steps it is possible that the price has stayed above barrier m for a period of length l . So for every node in this area, it is simply impossible for the option to expire or have expired. Therefore, the number of paths from the origin to any point in area I that have not led to expiration is simply the total number of paths: $G(k, i) = N(k, i)$ if $k < l + m$.

Area II

Next is area II, where $i \geq l + m$ and thus $k \geq l + m$. To get there the price must not only have reached m but it must have stayed above m for at least l consecutive steps. So for $i \geq l + m$, the option must have already expired. The number of ‘allowed’ paths to nodes in this area is therefore zero: $G(k, i) = 0$ if $i \geq l + m$.

Area III

Area III consists of points below the barrier that have possibly spent a longer time period than l above the barrier H . Since every trajectory ending in area III makes its last step below the barrier, the final step does not influence what we are looking at: time spent above the barrier. Therefore we may use a recursive relation, using the fact that the last step was either an up- or down-step: the number of steps to a point (k, i) in III is given by the number of steps to $(k - 1, i + 1)$ plus the number of steps to $(k - 1, i - 1)$. So $G(k, i) = G(k - 1, i - 1) + G(k - 1, i + 1)$ if $i < H, k > m + l$.

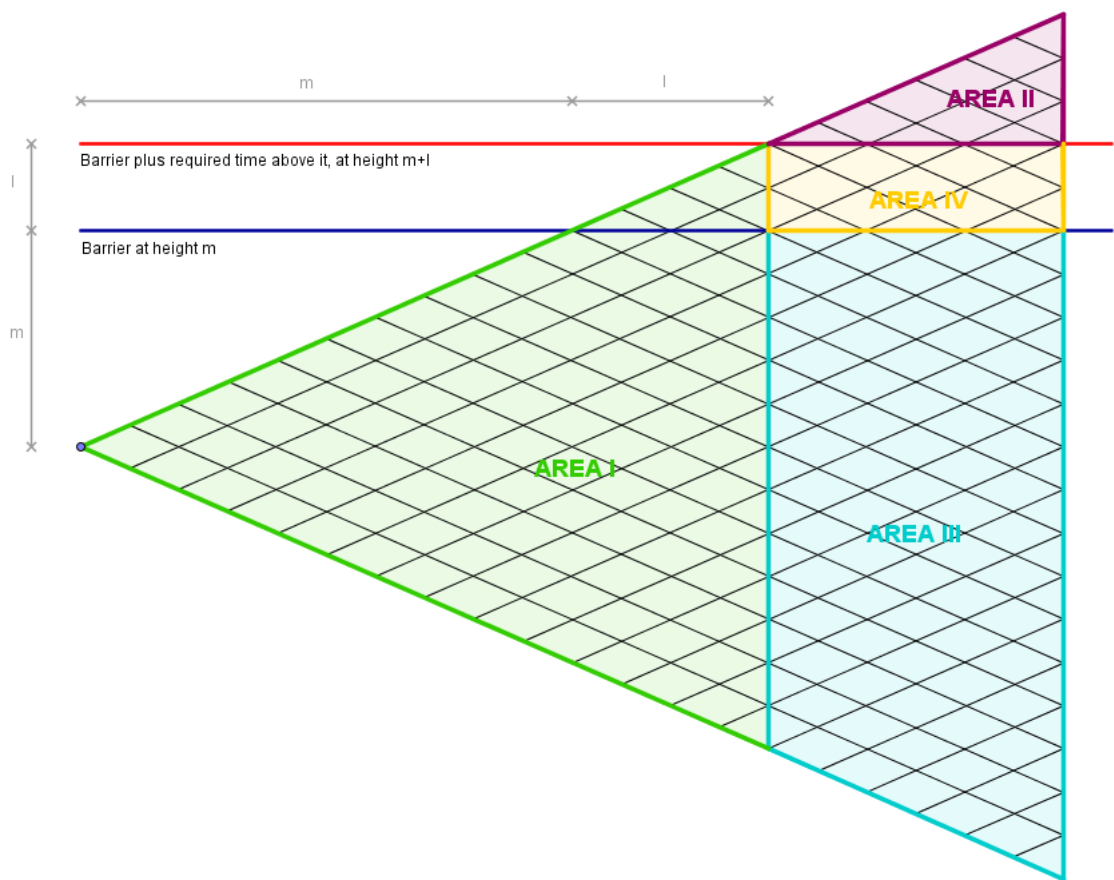


Figure 3.3: Binomial tree - four areas needed to construct $G(k, i)$

Area IV

Now we arrive at area IV. This is the most difficult one. To every node in this area lead paths that have already led to expiration as well as paths that have not caused expiration yet. Since it is possible that expiration occurs at the final step prior to the node, we cannot just use the simple recursive relation that we have used for area III. So in order to calculate $G(k, i)$, we need to exclude paths that have already expired or expire in the last step.

The area we were looking at is nodes (k, i) with $m \leq i < l + m$, $k \geq l + m$. The following formula provides us with an expression for $G(k, i)$: *The number of paths from the origin to a node in area IV that have stayed above the barrier less than l steps, is given by*

$$G(k, i) = \sum_{h=0}^{0.5(l-(i-m))} G(k - 2h - (i - m) - 1, m - 1) \cdot \left[\binom{2h + i - m}{h + i - m} - \binom{2h + i - m}{h + 1 + i - m} \right]. \quad (3.8)$$

To explain this, start by noting that all paths to a node in this area that *have not expired* can be split up into two parts. Part one is a path from the origin that arrives at a node at height $m - 1$ after let's say $k - x - 1$ steps and then makes an up-step. Hence, this is a path from the origin to the barrier: $(0, 0) \rightarrow (k - x, m)$. Part two starts on the barrier and - using less than l steps - ends in the node we are looking at. This means it does $(k - x, m) \rightarrow (k, i)$ (for which we have $m \leq i < l + m$). Multiplying the number of possible 'part one's with the number of possible 'part two's gives us the number of non-expired paths from $(0, 0)$ to (k, i) in area 4 that have remained above the barrier since they were in node $(k - x, m)$.

Part I - $(0, 0) \rightarrow (k - x, m)$ with the conditions: the path has not caused expiration and the last step is an up-step.

The total number of non-expired paths from $(0, 0)$ to $(k - x, m)$ is the sum of the number of non-expired paths from $(0, 0)$ to $(k - x - 1, m - 1)$ and the number of non-expired paths from $(0, 0)$ to $(k - x - 1, m + 1)$: $G(k - x - 1, m - 1) + G(k - x - 1, m + 1)$. But remember the condition that the last step is an up-step, we may not include the paths coming through $G(k - x - 1, m + 1)$. Therefore the number we are looking for is given by $G(k - x - 1, m - 1)$.

Part II - $(k - x, m) \rightarrow (k, i)$ with the condition: the path has consecutively stayed above the barrier.

(Later on we restrict possible values for x to assure that the path has not yet expired.) The number of possible paths from one node to another not crossing the barrier, is given by the total number of paths minus the number of paths that *do* cross the barrier. The total number of paths, $(k - x, m) \rightarrow (k, i)$, is given by the usual binomial coefficient:

$$\binom{k - (k - x)}{i - m + \frac{1}{2}(k - (k - x) - (i - m))} = \binom{x}{\frac{1}{2}(x + (i - m))}$$

The number of paths that *cross* barrier m is the number of paths that *touch or cross* the line $m - 1$ and can be found using the reflection principle. The reflection of $(k - x, m)$ in the line $m - 1$ is $(k - x, m - 2)$, so the number of paths crossing barrier m is the total number of paths from $(k - x, m - 2)$ to (k, i) and is given by the binomial coefficient:

$$\binom{k - (k - x)}{i - (m - 2) + \frac{1}{2}(k - (k - x) - (i - (m - 2)))} = \binom{x}{\frac{1}{2}(x + i - m + 2)}$$

So finally, the number of Part II paths we are looking for is given by:

$$\binom{x}{\frac{1}{2}(x + (i - m))} - \binom{x}{\frac{1}{2}(x + i - m + 1)} \quad (3.9)$$

Multiplying the number of possible paths for part I and part II, we get the following expression:

$$G(k - x - 1, m + 1) \left[\binom{x}{\frac{1}{2}(x + (i - m))} - \binom{x}{\frac{1}{2}(x + i - m + 1)} \right] \quad (3.10)$$

To obtain this equation we have - out of the blue - said that we are on the barrier after $k - x$ steps and that we stay on and/or above the barrier after $k - x$. But there are multiple possibilities for x that should all be taken into account. The number of paths we are looking for is a sum of the above expression over a certain range of values for x . Now let's look at the possible values for x . At (k, i) , we are $i - m$ nodes above m . That means the last possible node to reach the barrier and stay above it is $(k - (i - m), m)$, which leads to $x \geq i - m$. On the other hand, we have been above the barrier for less than l steps so the first possible node to reach and stay above or on the barrier is $(k - l + 1, m)$, leading to the condition $x \leq l - 1 \Leftrightarrow x < l$. That means we have $i - m \leq x < l$.

Furthermore we have to take into account that height m is either even or uneven and can only be reached after an amount of steps that is also even or uneven. That means we cannot let x run over all values between $i - m$ and l ; we have to skip every other number. To avoid this, we can use an alternative expression involving some number $2h$ so that it is possible to sum for a range of values for h . We make the replacement $x = 2h + i - m$. Filling this into the constraints for x gives:

$$i - m \leq 2h + i - m < l \Leftrightarrow 0 \leq 2h < l - (i - m) \Leftrightarrow 0 \leq \frac{l - (i - m)}{2}$$

Making the replacement for x in 3.10 and summing over all possible values for h results in the expression:

$$G(k, i) = \sum_{h=0}^{\frac{l - (i - m)}{2}} G(k - 2h - (i - m) - 1, m - 1) \left[\binom{2h + i - m}{h + i - m} - \binom{2h + i - m}{h + 1 + i - m} \right]$$

which equals 3.8. This means we have found an expression - albeit recursive - for $G(k, i)$, the number of paths from $(0, 0)$ to (k, i) for which a Parisian option has not yet expired.

Chapter 4

Regular barrier option prices

This chapter will discuss two methods to calculate prices of regular barrier options. The first method is based upon counting paths. The price formula can be given right away, based upon results from Chapter 3. The second method, topic of Section 4.2, uses a different approach and requires explanation in this chapter. Section 4.3 discusses the implementation of both methods in Matlab and section 4.4 will discuss the numerical results that were obtained using both methods.

4.1 Method I: counting paths

In Section 3.2 we found a formula for the value of a regular barrier option, depending on $F(n, 2j - n)$, denoting the number of paths from $(0, 0) \leftarrow (k, i)$ for which the option has not yet expired:

$$V_0^B = e^{-rT} \sum_{j=0}^n F(n, 2j - n) p^j q^{n-j} (S_T(n, 2j - n) - K)^+. \quad (4.1)$$

Later in Section 3.4 we found an expression for $F(k, 2j - k)$:

$$F(k, 2j - k) = \binom{k}{j} - \binom{k}{j - m} \quad (4.2)$$

Again, this expression is a discounted expectation of the final option payoffs. For every final option payoff the probability of finishing in that node is the probability to take exactly enough up- and downsteps to end up there times the number of ways in which you can do that.

To reduce computations we can specify this formula a little bit further. We currently sum over all possible final nodes but we know that in certain nodes the option payoff will be zero. These are on the upper end nodes that lie above the barrier and on the lower end nodes for which the asset price is lower than the strike price. On the upper end this means we only need to sum up to $j = m - 1$ since the option is out at $j = m$. On the lower end, we need to find for which j we have that $Su^j d^{n-j} > K$. Rewriting

this inequality gives:

$$\begin{aligned}
u^j d^{n-j} &> \frac{K}{S} \\
(u/d)^j &> \frac{K}{Sd^n} \\
j \log(u/d) &> \log(K/Sd^n) \\
j &> \frac{\log(K/Sd^n)}{\log(u/d)}
\end{aligned}$$

Any value for j must be an integer so if we call $a := \left\lceil \frac{\log(K/Sd^n)}{\log(u/d)} \right\rceil$ then we have $V_N(n, 2j - n) = 0$ for all $j < a$. Now we may write the price formula as

$$V_0^B = e^{-rT} \sum_{j=a}^{m-1} F(n, 2j - n) p^j q^{n-j} (S_T(n, 2j - n) - K). \quad (4.3)$$

4.2 Method II: recursively backwards

This method uses different calculations to do essentially the same as the counting method. Calculating the option price is all about the expected value and when it comes to the barrier case, it is all about making sure that paths that hit the barrier *do not contribute any value to the option price*. In case of Method I, value from expiring paths that expire early is eliminated by looking at the number of paths to a certain node that are allowed. But we can also eliminate value from expiring paths while working backwards in the tree. As long as we make sure that *every* path that touches or crosses the barrier adds zero to the expected option value.

Recall from Section 2.3 that the value of an option after k steps is the discounted expectation of the option value after $k + 1$ steps. There we said the following: V_n is a random variable that pays off at n and depends on the coin tosses $\omega_1, \omega_2, \dots, \omega_n$. The value of V_k is defined recursively backward in time for $k = n - 1, n - 2, \dots, 0$ as follows:

$$V_k(\omega_1 \omega_2 \dots \omega_k) = \frac{1}{1+r} [\tilde{p} V_{k+1}(\omega_1 \omega_2 \dots \omega_k H) + \tilde{q} V_{k+1}(\omega_1 \omega_2 \dots \omega_k T)].$$

This way, every V_k (except V_n) depends on the coin tosses that have occurred before k . Since we know every possible V_n we can calculate backwards to find V_0 . Basically this is also a way of covering every path through the tree. Now how do we make sure that paths that hit the barrier at some point do not contribute to the expectation?

The essential observation we need to make is that every path that causes the option to expire must do so by making one of the following steps:

$$\begin{aligned}
(m-1, m-1) &\rightarrow (m, m) \\
(m-1+2, m-1) = (m+1, m-1) &\rightarrow (m+2, m) \\
(m+3, m-1) &\rightarrow (m+4, m), \\
(m+5, m-1) &\rightarrow (m+6, m), \text{ etc.}
\end{aligned}$$

This is illustrated in Figure 4.1. The red arrows represent all steps that cause expiration. The expected value of the option when the asset price is at the end of any of those red

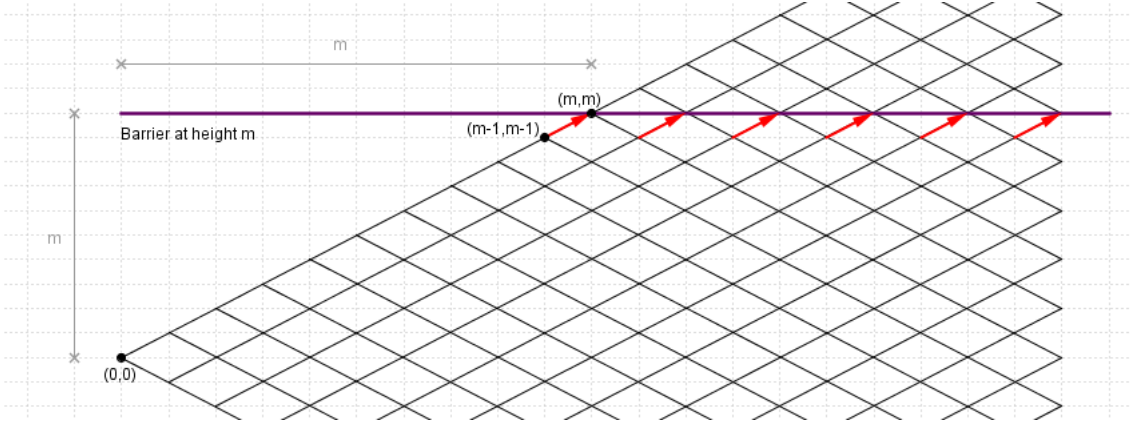


Figure 4.1: Possible expiration-steps for a regular barrier option

arrows must be zero. When we define in our recursive scheme $V_k(k, m) = 0$ for every k on the barrier then this causes all value arising from expired paths to be removed from the expectation.

In Section 2.3 we calculated backwards point by point but we can also work backwards using vectors. At every iteration, we set the option value in nodes on or above the barrier to zero. Figure 4.2 gives an algorithmic description of this. Vector $\mathbf{V}_t$ gets shorter and eventually V_0 is a single number. We use $h = \frac{T}{n}$ and let k steps correspond time t , so $t = h \cdot k$.

4.3 Matlab implementation

The example that I have implemented in Matlab uses values taken from Costabile [5]. The following constants are used:

US interest rate	$r = 0.056$
Japan interest rate	$\gamma = 0.007$
volatility	$\sigma = 0.13$
initial asset price	$S_0 = 1/120.5$
strike price	$K = 1/125$
barrier height	$H = 1/110$
lifetime in years	$T = 0.5$

Based on Black Scholes are expressions for u and d taken from [6]. The same goes for p and q but these are modified by [5] for the underlying involving also γ . The formula for n as used by [5] is based on Boyle and Lau [2] who have showed this to be a convenient choice to avoid ‘bumping up against the barrier’. We have the following expressions:

1. Define the following vectors:

$$\mathbf{S}_t = \begin{pmatrix} S_t(k, k) \\ S_t(k, k-2) \\ \vdots \\ S_t(k, -k+2) \\ S_t(k, -k) \end{pmatrix} = \begin{pmatrix} S_0 u^k d^0 \\ S_0 u^{k-2} d^2 \\ \vdots \\ S_0 u^2 d^{k-2} \\ S_0 u^0 d^k \end{pmatrix} \text{ and} \quad (4.4)$$

$$\mathbf{V}_t = \begin{pmatrix} V_t(k, k) \\ V_t(k, k-2) \\ \vdots \\ V_t(k, -k+2) \\ V_t(k, -k) \end{pmatrix} = (\mathbf{S}_t - K)^+ \cdot (\mathbf{S}_t < H). \quad (4.5)$$

The vector $(\mathbf{S}_t < H)$ is a vector with logicals consisting of zeros and ones. This vector sets the option value in final nodes on or above the barrier to zero. The expression $\cdot \times$ denotes a pointwise vector multiplication.

2. Set $T = t$.
3. For every $t \geq 0$:

4. Take two vectors out of $\mathbf{V}_t$:

$$\mathbf{V}_t^{up} = \begin{pmatrix} V_t(k, k) \\ V_t(k, k-2) \\ \vdots \\ V_t(k, -k+2) \end{pmatrix}, \mathbf{V}_t^{down} = \begin{pmatrix} V_t(k, k-2) \\ \vdots \\ V_t(k, -k+2) \\ V_t(k, -k) \end{pmatrix}. \quad (4.6)$$

5. Set $\mathbf{S}_{t-h}$:

$$\mathbf{S}_{t-h} = d \cdot \begin{pmatrix} S_t(k, k) \\ S_t(k, k-2) \\ \vdots \\ S_t(k, -k+2) \end{pmatrix} = u \cdot \begin{pmatrix} S_t(k, k-2) \\ \vdots \\ S_t(k, -k+2) \\ S_t(k, -k) \end{pmatrix}. \quad (4.7)$$

6. Calculate $\mathbf{V}_{t-h}$:

$$\mathbf{V}_{t-h} = e^{-rh} \cdot (p\mathbf{V}_t^{up} + q\mathbf{V}_t^{down}) \cdot (\mathbf{S}_{t-h} < H) \quad (4.8)$$

where $(\mathbf{S}_{t-h} < H)$ is again a vector with logicals and $\cdot \times$ denotes pointwise multiplication.

7. Set $t = t - h$.
8. The option price at $t = 0$ is now given by $V_0 = \mathbf{V}_0$.

Figure 4.2: Algorithmic description of Method II

number of time steps	$n = \left\lfloor \frac{m^2 \sigma^2 T}{\log^2(H/S)} \right\rfloor$
length of one time step	$h = \frac{T}{n}$
up factor	$u = e^{\sigma \sqrt{h}}$
down factor	$d = e^{-\sigma \sqrt{h}}$
probability of up-step	$p = \frac{e^{(r-\gamma)h} - d}{u - d}$
probability of down-step	$d = 1 - p = \frac{u - e^{(r-\gamma)h}}{u - d}$

After setting the constants and calculating these variables we need to implement a calculation of the actual price. Both ways to do this, as discussed in Section 4.1 and Section 4.2, have been implemented.

4.3.1 Implementation of Method I

This method requires some carefull programming, especially if one wants to make it as quick as possible. Remember the formula for $F(n, 2j - n)$. We are going to rewrite this expression:

$$\begin{aligned}
F(n, 2j - n) &= \binom{n}{j} - \binom{n}{j - m} \\
&= \frac{n!}{(n - j)!j!} - \frac{n!}{(j + m)!(n - (j + m))!} \\
&= \frac{n! \prod_{i=1}^m (j + i)}{(n - j)!(j + m)!} - \frac{n! \prod_{i=1}^m (n - j - m + i)}{(j + m)!(n - j)!} \\
&= \frac{n!}{(n - j)!(j + m)!} \left[\prod_{i=1}^m (j + i) - \prod_{i=1}^m (n - j - m + i) \right] \\
&= \frac{n!}{(n - j)!j!} \cdot \frac{1}{\prod_{i=1}^m (j + i)} \left[\prod_{i=1}^m (j + i) - \prod_{i=1}^m (n - j - m + i) \right] \\
&= N(n, 2j - n) \cdot \left[1 - \prod_{i=1}^m \frac{(n - j - m + i)}{(j + i)} \right] \tag{4.9}
\end{aligned}$$

Filling this into V_0^B gives the following formula:

$$V_0^B = e^{-rT} \sum_{j=0}^n N(n, 2j - n) \cdot \left[1 - \prod_{i=1}^m \frac{(n - j - m + i)}{(j + i)} \right] p^j q^{n-j} (S_T(n, 2j - n) - K)^+.$$

Matlab cannot evaluate $N(k, i)$ for the numbers we are working with because the value for N simply gets too large for large n . Naturally the formula also includes a term that is going to get very small for large n : $p^j q^{n-j}$ (since $p < 1$ and $q < 1$). We program evaluation of these two terms multiplied by eachother in order to keep working with reasonable numbers. The formula below shows which three vectors we construct so that the final value can easily be calculated. The Matlab code can be found in Appendix A.1.

$$V_0^B = e^{-rT} \sum_{j=0}^n \underbrace{\frac{n!}{(n - j)!j!} p^j q^{n-j}}_{\text{Pf}} \underbrace{\left[1 - \prod_{i=1}^m \frac{(n - j - m + i)}{(j + i)} \right]}_{\text{Cf}} \underbrace{(S_T(n, 2j - n) - K)}_{\text{Vf}}$$

				Method I		Method II	
m	n	a	$\lfloor \frac{1}{2}(n+m) \rfloor$	V_0^B	t_I	V_0^B	t_{II}
10	101	49	55	1.4241e-004	0.0007	1.4241e-004	0.0098
20	406	199	213	1.4003e-004	0.0015	1.4003e-004	0.0163
32	1041	515	536	1.4060e-004	0.0018	1.4060e-004	0.0384
40	1626	805	833	1.4046e-004	0.0034	1.4046e-004	0.0686
50	2541	1261	1295	1.4067e-004	0.0046	1.4067e-004	0.1386

Table 4.1: Prices and computation times for Method I & II

4.3.2 Implementation of Method II

Implementation of this method is extremely easy in terms of the amount of code that is needed. Matlab code for this method can also be found in Appendix A.1. Before starting the recursion, 2 vectors are calculated. One contains all possible asset prices after n steps ($\mathbf{Sf}$) and the other contains all possible option payoffs after n steps ($\mathbf{Vf}$, the same as above in Method I). During the recursion we can keep working with these 2 vectors without needing to store any other vectors. At every time-step, two vectors are drawn out of V_k , multiplied by p and q , discounted and compared to barrier H to obtain a vector for V_{k-1} . The vector gets shorter and shorter and eventually results in a single value: V_0^B .

4.4 Results

This section will present a table with prices that were found for different values of m . All the values in Table 4.4 are equal to the values that Lerouge [7] found based on Costabile [5], using a window period of 0 days. We are interested in whether Method I or Method II performs faster so the table presents calculation times for both methods. From these results we can conclude that Method I performs significantly faster than Method II.

Looking at the code, it is the `for`-loop that causes Method II to be slower. Method I however also contains `for`-loops but apparently these do not take a lot of time to work through. After a closer look at why this is the case, we find that it is caused by the fact that Method II must always loop over n and Method I calculates $\mathbf{Pf}$ and $\mathbf{Cf}$ only for values between a and $\lfloor \frac{1}{2}(n+m) \rfloor$. Looking at Table 4.4, we see that they lie closely together compared to the total number of steps n .

These two values lie close together when a is relatively high and $\lfloor \frac{1}{2}(n+m) \rfloor$ is relatively low. Remember that a is the number of up-steps that need to be taken to be in-the-money at maturity. In this case we are working with $S_0 = 0.0083$ and $K = 0.0080$ so in order to end in-the-money we are not allowed to take many down-steps. This means a is quite high. And then there is $\lfloor \frac{1}{2}(n+m) \rfloor$: making more up-steps than this will surely cause you to end up on or above the barrier. This number of steps will depend on the height of the barrier compared to the initial price, in our case $H = 0.0091$ and $S_0 = 0.0083$. S_0 and H lie quite close together and indeed $\lfloor \frac{1}{2}(n+m) \rfloor$ is relatively small.

So the amount of work done by Method II is constant relative to n but the amount of work that Method I needs to do depends on the values a and $\frac{1}{2}(n+m)$. We can try to put in some other numbers that give Method I a hard time and compare methods

m	100	150	200
n	621	1397	2485
a	0	223	609
$\lfloor \frac{1}{2}(n+m) \rfloor$	360	773	1342
$V_{0,I}^B$	0.0075	0.0075	0.0075
t_I	0.0089	0.0197	0.0383
$V_{0,II}^B$	0.0075	0.0075	0.0075
t_{II}	0.0233	0.0578	0.1323

Table 4.2: Results for $K = 0.0008$, $H = 0.0120$.

m	100	150	200
n	2429	5467	9719
a	588	1793	3606
$\lfloor \frac{1}{2}(n+m) \rfloor$	1264	2808	4959
$V_{0,I}^B$	0.0069	NaN	NaN
t_I	0.0397	...	...
$V_{0,II}^B$	0.0069	0.0069	0.0069
t_{II}	0.1259	0.8430	1.9974

Table 4.3: Results for $K = 0.0008$, $H = 0.0100$.

again. First of all we set the strike price very low at $K = 0.0008$, so that a will be low. $S_0 = 0.0083$ stays the same. Now for $H = 0.0100$ and $H = 0.0120$ we will vary m and look at the results.

Table 4.3 shows what happens when we set $H = 0.0120$ and take $m = 100, 150, 200$. This indeed causes Method II to perform relatively better compared to Table 4.1: Method I is now 3-4 times faster instead of 10-30 times faster. However, the values for K and H that we are using to make this point are not very realistic. Using the values from [5] values shows that Method I is simply better. Besides, the fact that a and $\lfloor \frac{1}{2}(n+m) \rfloor$ lie not too far apart is no coincidence; it is due to the choice for n based on m , S and H .

In Table 4.4 we see that for very large n , Method I is no longer able to find a price. This is due to a part of the calculation that Matlab cannot handle when n gets too big. Method II still does the job. Basically this means that only when it comes to large n and m - leading to many elements between a and $\lfloor \frac{1}{2}(n+m) \rfloor$ - Method II performs better.

Chapter 5

Parisian option prices

Parisian option were introduced in Section 1.3.1. Section 5.1 will state the pricing formula for these options using the counting paths-approach and Section 5.2 will discuss the way I implemented the algorithm in Matlab. Section 5.3 introduces the problem of finding a relation between regular barrier and Parisian option prices, based on the height of the barrier and the required time above the barrier. Section 5.4 will present results that were obtained with Matlab.

5.1 Counting paths

In Section 3.5 we have found the following algorithm to count the number of paths from $(0,0)$ to (n,i) that do not cause the option to expire early:

- Area I: $G(k,i) = N(k,i)$ if $k < l + m$
- Area II: $G(k,i) = 0$ if $i \geq l + m$
- Area III: $G(k,i) = G(k-1,i-1) + G(k-1,i+1)$ if $i < H, k > m + l$
- Area IV: if $m \leq i < l + m$ and $k \geq l + m, G(k,i) =$

$$\sum_{h=0}^{\frac{1}{2}(l-(i-m))} G(k-2h-(i-m)-1, m-1) \left[\binom{2h+i-m}{h+i-m} - \binom{2h+i-m}{h+1+i-m} \right]$$

This is input for the price formula we found in Section 3.2:

$$V_0^P = e^{-rT} \sum_{j=0}^n G(n, 2j-n) p^j q^{n-j} (S_T(n, 2j-n) - K)^+ \quad (5.1)$$

As we did in the case of barrier options, we also define a as the minimum number of up-steps needed to be in-the-money at maturity:

$$a = \left\lceil \frac{\log(K/Sd^n)}{\log(u/d)} \right\rceil$$

In this case we could let the sum run up to $\lfloor \frac{1}{2}(n+l+m) \rfloor$ since any path that makes more than this many up-steps ends in node higher than $m+l$, which means it has surely

expired. The formula may be written as:

$$V_0^P = e^{-rT} \sum_{j=a}^{\lfloor \frac{1}{2}(n+l+m) \rfloor} G(n, 2j-n) p^j q^{n-j} (S_T(n, 2j-n) - K) \quad (5.2)$$

5.2 Matlab implementation

Unlike $F(n, i)$ in the regular barrier option case, the formula for $G(n, i)$ is recursive. Programming $G(n, i)$ as the algorithm above results in a function that frequently calls upon itself and is unable to handle large n . Therefore I wrote a function that constructs a large matrix with a tree in it that contains values for $G(n, i)$ for each node in the tree. Again, this function could not deal with large n , simply because the number $G(n, i)$ became too large to handle. Eventually the following was done:

$$V_0^P = e^{-rT} \sum_{j=0}^n \underbrace{\frac{G(n, 2j-n)}{N(n, 2j-n)}}_A \underbrace{N(n, 2j-n) p^j q^{n-j}}_{Pf} \underbrace{(S_T(n, 2j-n) - K)}_{Vf}$$

Again, the letters below the formula show the variable names used in Matlab. The code that calculates V_0^P can be found in Appendix A.2. Note that Vf and Pf are identical to the Vf and Pf calculated in the case of a regular barrier. The values for $\frac{G(n, 2j-n)}{N(n, 2j-n)}$ are found by constructing a large matrix and finally taking out the bottom row. The numbers in the matrix represent for every node the probability that the option still exists when the asset price is in that node. Values range from 0 to 1 so the problem of values that are too large or too small is avoided.

5.3 A relation with regular barrier prices?

We will see below how much time it takes to compute a Parisian price compared to a regular barrier price. This raises the question of whether we could somehow find a Parisian price in a less complicated way, by doing regular barrier price calculations. Parisian options are worth more than regular barrier options when they have the same barrier height because the barrier option has a higher chance of expiring early and ending up worthless. Because of the window period, a Parisian option expires less often on the same underlying. Just as the window period leads to less early expiration and a higher option value, so does a higher barrier. That means a regular barrier option with a higher barrier is more expensive than the ‘same’ option with a lower barrier.

This leads to the idea that we could try to find a barrier height H_{impl} related to H , for which a regular barrier option has the same price as the Parisian option. We would mainly expect this implied barrier height H_{impl} to depend on the window period because the longer the window period, the lower the chance of early expiration. In [1] the implied barrier concept is introduced for option pricing in continuous time, based on window periods. Besides window periods the authors also look at different expiration times. When T increases we would expect a lower Parisian price due to a higher probability of early expiration. This also leads to a lower implied barrier.

5.4 Results

Table 5.1 shows prices that were found with the code from Appendix A.2. By ‘wp’ the window period is denoted in days. Using 360 days a year, l is determined by $l = \frac{wp/360}{h}$ where h denotes the step size and the value for l that is found is rounded off. The values that I found equal those found by Lerouge [7] based upon Costabile [5].

		wp = 0 days		wp = 5 days		wp = 10 days		wp = 15 days	
m	n	l	V_0^P	l	V_0^P	l	V_0^P	l	V_0^P
10	101	0	1.4241e-004	3	1.9738e-004	6	2.2668e-004	8	2.4648e-004
20	406	0	1.4003e-004	11	2.0135e-004	23	2.3739e-004	34	2.6236e-004
32	1041	0	1.4060e-004	29	2.0569e-004	58	2.4019e-004	87	2.6907e-004
40	1626	0	1.4046e-004	45	2.0737e-004	90	2.4162e-004	135	2.7084e-004
50	2541	0	1.4067e-004	71	2.0897e-004	141	2.4381e-004	212	2.7258e-004

Table 5.1: Parisian option: prices for different m and l

Table 5.2 shows computational times with this code for the three window periods other than zero. Table 5.3 shows the computational times for a window period of 0 days, the situation of a regular barrier option. Therefore it also shows the times we found in Chapter 4 with Method I. Clearly this motivates the question of whether a relation can be found so that the Parisian price can be calculated by calculating some regular barrier price.

		wp = 5 days		wp = 10 days		wp = 15 days	
m	n	l	time	l	time	l	time
10	101	3	0.0632	6	0.0186	8	0.0157
20	406	11	0.1356	23	0.2045	34	0.2692
32	1041	29	0.9408	58	1.3970	87	1.8790
40	1626	45	2.3103	90	3.5675	135	4.8488
50	2541	71	5.8620	141	9.0907	212	12.8555

Table 5.2: Parisian option: computational times

		Parisian method		Chapter 4 - Method I	
m	n	l	time	l	time
10	101	0	0.0054	0	0.0012
20	406	0	0.0778	0	0.0031
32	1041	0	0.5051	0	0.0076
40	1626	0	1.2207	0	0.0125
50	2541	0	2.9718	0	0.0198

Table 5.3: Computational times compared for $l = 0$

Now let’s look at some results from trying to find a relation between regular barrier and Parisian prices as explained in Section 5.3. Figure 5.1 shows the implied barrier for

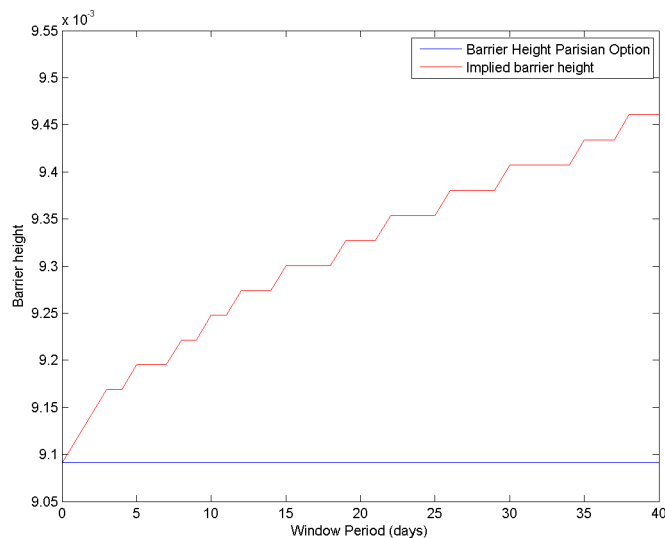


Figure 5.1: Relation between window period and implied barrier

different window periods. S and the barrier height of the Parisian option H_{par} were kept constant. This shows a linear relationship between the window period and the implied barrier height. The line is bumpy because of the way the while-loop is constructed to find H_{impl} and because we are working in discrete time.

Figure 5.2 is a surface plot of implied barriers. H_{impl} was calculated for a range of window periods as well as a range of different values for the expiration time T . As in the previous figure, we see that the implied barrier is positively related to the window period but we are unable to observe the expected negative relationship between the T -axis and the implied barrier.

One of the reasons we might not be able to observe such a relationship is that the interval for T we are looking at is $[0, 2]$. The program that is used does not work for large T (which leads to a large n) unless the tree grid is made coarse by setting m small. Doing this reduces the illustration of the linear relationship along the ‘window period’-axis and also does not show a negative relationship with the expiration time when $T = [0, 5]$. Besides the programming constraint on n and the resulting tradeoff between m and T we are using different numerical values than the numerical example in [1] so specific results cannot be compared. Therefore we cannot draw any conclusions about the relation between the implied barrier and the expiration time. Only Figure 5.1 serves as a clear illustration of the implied barrier concept in discrete time when it comes to the window period.

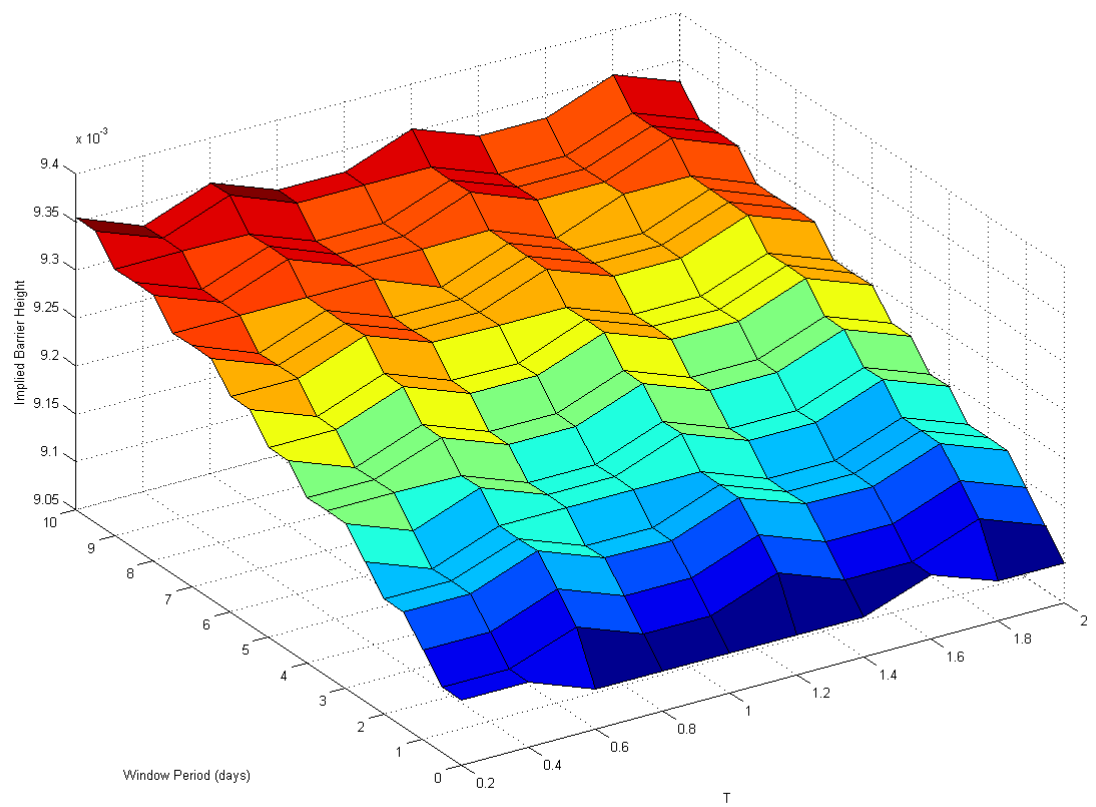


Figure 5.2: Implied barrier as function of window period and T

Chapter 6

Conclusion

This thesis has worked towards pricing methods for regular barrier and Parisian options in discrete time. With the Binomial No-Arbitrage Pricing Method as starting point, we found a direct price formula for standard options in at the end of Chapter 2. After looking into combinatorical formulas in Chapter 3 we eventually arrived at general pricing formulae for regular barrier and Parisian options. Both formulas have been implemented in Matlab and their results have been discussed in Chapters 4 & 5. Reference values from other literature indicates that the methods were implemented correctly.

In Chapter 4, an alternative recursive method to calculate regular barrier option prices was developed. Supervisor and author of this thesis do not know whether this method is new but neither of them has encountered the method in existing literature. Implementation of the method is very easy and results in a program that can handle a very large number of time-steps. Numerical results show that clever implementation of the ‘counting paths’-method results in a faster program than the recursive method. Both methods use loops which slow down calculations but due to certain choices for parameters, the combinatorical method loops over a relatively small amount of values whereas the recursive method must always loop over the total number of time-steps.

Further research on the recursive method that has been found was beyond the time-scope of this thesis. It would be interesting to work out the computational cost of the method in more detail and optimize it in terms of the number of multiplications that is needed and the amount of memory that is used. Also its concept could be applied to trinomial trees, which form a way of discrete asset price modelling that has not been discussed in this thesis.

Finally, in Section 5.4 we have shown that in discrete time, the relation between the window period and implied barrier height as discussed in [1] can be found by numerical experiment. The proposed relation between expiration time and implied barrier height could not be illustrated with the program available. Further research could look at both relations in discrete-time compared to continuous time in more detail.

Appendix A

Matlab code

A.1 Regular barrier option

%This file calculates the option price of a regular barrier option using
%two different methods.

%Set Constants

m=100	%height of barrier
r=0.056;	%US interest rate
gamma=0.007;	%Japan interest rate
sigma=0.13;	%volatility
S=1/120.5;	%initial asset price
K=1/125;	%strike price
H=1/110;	%height of barrier
T=0.5;	%lifetime

%Calculate variables

n=floor(T*(m*sigma/log(H/S))^2)	%number of time-steps
h=T/n;	%size of each step
u=exp(sigma*sqrt(h));	%up factor
d=1/u;	%down factor
p=(exp((r-gamma)*h)-d)/(u-d);	%probability of up-step
q=1-p;	%probability of down-step

% van=max(ceil(log(K/(S*d^n))/log(u/d)),0)

% tot=floor(0.5*(n+m))

%%%

%Method I: Counting paths

tic;

%Construct vectors with final asset prices and option payoffs

Sf=S*u.^(n:-2:-n);

Vf1=(Sf-K).*((Sf<H).*(Sf>K));

%Calculate the min. no of up-steps to be in the money at maturity

a=max(ceil(log(K/(S*d^n))/log(u/d)),0);

```

%Construct vector with probabilities for asset price to end at points that
%are in the money and below the barrier
Pf=zeros(n+1,1);
for k = a:floor(0.5*n)
    N=(n-k+1):1:n;
    D=1:1:k;
    Pf(k+1)=prod(p*q*N./D)*q^(n-2*k);
end
for k = ceil(0.5*n):floor(0.5*(n+m))
    N=(k+1):1:n;
    D=1:1:n-k;
    Pf(k+1)=prod(p*q*N./D)*p^(2*k-n);
end
Pf=flipud(Pf);

%Construct a 'correction' vector that eliminates value from expired paths.
Cf=zeros(n+1,1);
for j = n-floor(0.5*(n+m)):n-a
    N=n-j-m+1:1:n-j;
    D=j+1:1:j+m;
    Cf(j+1)=1-prod(N./D);
    XX2(j+1)=Cf(j+1);
end

%Multiply the 3 vectors pointwise, sum and discount to get V_0
Vfe=exp(-r*T)*sum(Vf1'.*Pf.*Cf)
t1=toc

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Method II: recursively backwards

tic;
%Construct vectors with final asset prices and option payoffs
Sf=S*u.^(n:-2:-n);
Vf2=(Sf-K).*((Sf<H).*(Sf>K));
%Calculate backwards to find V_0
for i=1:1:n
    Sf=d*Sf(1:end-1);
    Vf2=exp(-r*h)*(p*Vf2(1:end-1)+q*Vf2(2:end)).*(Sf<H);
end
Vf2
t2=toc
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

A.2 Parisian option

```
function price = paris(m,wp)
```

```

%Constants
r=0.056; %US interest rate
gamma=0.007; %Japan interest rate
sigma=0.13; %volatility
S=1/120.5; %initial asset price
K=1/125; %strike price
H=1/110; %height of barrier
T=0.5; %lifetime

%Calculate values that are needed later on
n=floor(T*(m*sigma/log(H/S))^2); %number of time-steps
h=T/n; %size of each step
u=exp(sigma*sqrt(h)); %up factor
d=1/u; %down factor
p=(exp((r-gamma)*h)-d)/(u-d); %probability of up-step
q=1-p; %probability of down-step

diy=360; %number of days per year
wpd=wp/diy; %window period (in years)
l=round(wpd/h); %number of steps in window period

%Number of up-steps needed to be in the money at maturity.
min_itmam = max(ceil(log(K/(S*d^n))/log(u/d)),0);

%Generate 'still-existing'-probability-matrix
GNIf = zeros(n,2*n+1);
%Two outer branches
for N=1:n
    GNIf(N,n+1-N)=1;
    GNIf(N,n+1+N)=1;
end
%AREA 2
GNIf(:,(n+1+m+1:n+n+1))=0;
%AREA 1
for N=1:(l+m-1)
    for i=-n:(l+m-1)
        if (floor((N+i)/2)==((N+i)/2))&&(N>abs(i))&&(i+n~=0)
            GNIf(N,i+n+1)=1;
        end
    end
end
for N=(l+m):n
    for i=-n:(l+m-1)
        %AREA 3
        if (i<m)&&(mod(N+i,2)==0)&&(N>abs(i))
            GNIf(N,i+n+1)=(.5*(N-i)/N)*GNIf(N-1,i+n+2)+(.5*(N+i)/N)*GNIf(N-1,i+n);
        %AREA 4
        elseif (i>=m)&&(mod(N+i,2)==0)&&(N>=abs(i))

```

```

        t1=0.5*(N-i);t2=i-m+1;
        k=1:ceil((1-i+m)/2-1);
        KV=cumprod([prod([t1+m:0.5*(N+i)]./[N-i+m:N]),(t1-k+1)./(N-2*k-t2+1).*
        ... (t1-k+m)./(N-2*k+2-t2).*(2*k-2+t2)./k.*(2*k+t2-1)./(k+t2)]];
        GNIf(N,i+n+1)=KV*GNIf(N-2*[0,k]-t2,n+m);
    end
end
end
%Take out 'final' probabilities
Ef=fliplr(GNIf(n,1:2:end))';
clear GNIf;

%Calculate final asset prices and option values
Sf=fliplr(S*u.^(-n:2:m+1-1));
Sf=[zeros(1,n+1-length(Sf)),Sf]; %vector with final underlying asset prices
Vf=(Sf-K).*(Sf>K); %vector with final values of option, given existence

```

A.3 Finding a relation

%This m-file runs "bar_imp_wp" as function of T. This means its result
 %is a matrix with implied barrier values for different T's and different
 %window periods.

```

%Vector with T's
T=0.2:0.2:2;
no_of_wp=20;

%Matrix to put in results
RM=zeros(no_of_wp+1,2*length(T));

%Evaluate for every T
for i=1:length(T)
    RM(:,2*i-1:2*i)=bar_imp_wp(T(i),no_of_wp);
end

%Take out implied barrier vectors
A=RM(:,2:2:2*length(T));

%Make surfaceplot of results
surf(T,0.5*RM(:,1),A)
xlabel('T')
ylabel('Window Period (days)')
zlabel('Implied Barrier Height')
saveas(gca,'surf.png');

```

```

function rangewp = bar_imp_wp(T,no_of_wp)
%Calculate replicating barrier height for 41 different window periods.

```

```

results=zeros(no_of_wp+1,3);

%Set constants for Parisian option
H=1/110;
m=10;
%Set other constants
r=0.056;           %US interest rate
gamma=0.007;       %Japan interest rate
sigma=0.13;        %volatility
K=1/125;           %strike price
S=1/120.5;         %initial asset price
diy=360;           %no of days in a year

for wp=0:no_of_wp
%Determine n,l,u,d,p,q,a
n=floor(T*(m*sigma/log(H/S))^2)
h=T/n;
wpd=0.5*wp/diy;
l=round(wpd/h);
u=exp(sigma*sqrt(h));
d=1/u;
p=(exp((r-gamma)*h)-d)/(u-d);
q=1-p;
min_itmam = max(ceil(log(K/(S*d^n))/log(u/d)),0);

%Calculate price of Parisian option
p_paris = paris(n,T,u,d,p,q,K,S,r,l,m,min_itmam);
%Calculate price of regular barrier option
p_barrier = barrier(n,h,u,d,p,q,K,S,H,r,m);

%Run barrier.m for different m's to see which barrier gives a price
%that lies closest to the price of the parisian option.
p_bar=[]; Hbar=[]; i=1; p_bar(i)=p_barrier; Hbar(i)=H;
while p_bar(i)<=p_paris
    i=i+1;
    Hbar(i)=S*u^(m+i-1);
    p_bar(i)= barrier(n,h,u,d,p,q,K,S,Hbar(i),r,m+i-1);
end

%Set implied barrier in results-vector
results(wp+1,:)=[wp,H,Hbar(end)];
end

rangewp = [results(:,1),results(:,3)];

```

Bibliography

- [1] Anderluh, J.H.M. & Weide, J.A.M. van der (2004): *The Implied Barrier Concept*. Lecture Notes in Computer Science, Springer-Verlag.
- [2] Boyle, P.P. & Lau, S.H. (1994): *Bumping up against the barrier with the binomial method*. The Journal of Derivatives.
- [3] Brealey, R.A., Myers, S.C. & Allen, F. (2005): *Principles of Corporate Finance*. McGraw-Hill.
- [4] Costabile, M. (2000): *A discrete algorithm for pricing double barrier options*. Decisions in Economics and Finance, Springer-Verlag.
- [5] Costabile, M. (2002): *A combinatorial approach for pricing Parisian options*. Decisions in Economics and Finance, Springer-Verlag.
- [6] Cox, J.C., Ross, S.A. & Rubinstein, M. (1979): *Option Pricing: A Simplified Approach*. Journal of Financial Economics.
- [7] Lerouge, S. (2009): *Une approche combinatoire pour le pricing des options Parisiennes*. ENSTA ParisTech.
- [8] Lyuu, Y.D. (1998): *Very fast algorithms for barrier option pricing and the Ballot problem*. The Journal of Derivatives.
- [9] Feller, W. (1950): *An introduction to probability theory and its applications - Volume I*. John Wiley & Sons
- [10] Shreve, S.E. (2005): *Stochastic Calculus for Finance I: The Binomial Asset Pricing Model*. Springer Finance.