A comparative study of tree-based models and their applications in modern finance

by

Johannes Barend van Biljon

Submitted in partial fulfillment of the requirements for the degree

Magister Scientiae

in the Department of Mathematics and Applied Mathematics
in the Faculty of Natural and Agricultural Sciences

University of Pretoria
Pretoria

Supervisor: Prof. E. Maré

May 2014

Declaration

I, the undersigned, declare that the dissertation, which I hereby submit for the degree Magister Scientiae at the University of Pretoria, is my own work and has not previously been submitted by me for a degree at this or any other tertiary institution.

Signature: ....
J.B. van Biljon

Date: 20 May 2014 ....

Abstract

A comparative study of tree-based models and their applications in modern finance

J.B. van Biljon
Supervisor:Prof. E. Maré
University of Pretoria
Pretoria

Thesis: MSc (Financial Engineering)
May 2014

The Binomial option pricing model plays an integral role in modern finance due to its simplicity to implement and pedagogical value. There are two ways of extending the Binomial model on one source of underlying risk. The first is to expand the number of possible states after each time step which results in the multinomial model. The second is to increase the number of sources of underlying risk. In this dissertation, the extension of the Binomial model in both cases is discussed.

Numerical investigation is done to evaluate convergence patterns and computational intensity of a number of non-vanilla options. These include rainbow, basket and digital options, as well as convertible bonds. Theoretical and actual convergence is discussed and compared.

Acknowledgements

I would firstly like to thank Professor Eben Maré for his valuable insight, help and most of all his enthusiasm and positive encouragement.

I would also like to thank Anzel and my family for their patience, support and encouragement over the past few years. I would not have been able to finish without you.

Finally I would like to thank the Risk Model Development team and management at Standard Bank for their patience and assistance.

Contents

List of Figures

List of Tables

Nomenclature

Variable	Description
T	Time period of option
t	Current time
$f(t,S)$	Option value
$S_i(t)$	Share price i at time t
$X_i(t)$	Log transformed share price i at time t
$Y_i(t)$	Log transformed share price i at time t
σ_i	Volatility of asset i
ρ_{ij}	Correlation between returns of asset i and j
K	Strike
r	Annual continuously compounded risk-free rate of interest
k	Kurtosis
p_l	Probability of movement l
q_l	Risk neutral probability of movement l
u_l	Size of jump l
d_l	Size of downward jump l
U_l	Log transformed Size of jump l
λ	Recovery rate of Convertible Bond
δ	Probability of default of Convertible Bond
c	Conversion factor of Convertible Bond
$O(.)$	Order notation

Table 1: Nomenclature

Acronym	Description
CRR	Cox, Ross and Rubinstein
RB	Rendleman and Bartter
BS	Black-Scholes
BEG	Boyle, Evnine and Gibbs
KR	Kamrad and Ritchken
MGF	Moment Generating Function
PDE	Partial Differential Equation
ATM	At The Money
OTM	Out of the Money
ITM	In The Money

Table 2: Acronyms

Chapter 1

Introduction

1.1 Objectives

The role of financial modeling in pricing complex financial derivatives has played a pivotal role in expanding the financial industry over the past few decades.

Many modern derivatives cannot be priced using closed-form solutions, leaving lattice type models or simulations. Lattice models are effective in their simplicity and offer a viable alternative to running time consuming simulations [50]. Changes in computing power in more recent times has seen the time taken by the simulations decrease substantially.

Due to the popularity of these complex products, a large amount of investigation and research has gone into these pricing methods. In Cox, Ross and Rubinstein (CRR) [17] the Binomial tree method to price European and American type options is introduced. The simplicity of the method, as well as the fact that the method had the ability to accurately price path dependent options, made it one of the cornerstones of modern quantitative finance.

Most introductions to quantitative finance start with the binomial option pricing model on one source of underlying risk. This dissertation will provide a view on tree-based models, expanding the Binomial model on one underlying source of risk into higher dimensions and also increasing the number of states into multinomial modeling. The aim of this dissertation is to investigate lattice models in multiple dimensions. The investigation will attempt to add to the pedagogy of the lattice literature by visually representing multidimensional lattice problems.

The bulk of research available has been focussed on either the expansion of the Binomial model into higher states, or looking at using the Binomial model to price derivatives with multiple sources of underlying risk. This dissertation covers both these topics, and looks at the implications of combining higher order lattice models, with derivatives with multiple sources of underlying risk.

The second chapter will focus on explaining and deriving the Binomial model in its most basic CRR [17] form for one source of underlying risk. This Binomial model will then be expanded into the multinomial world of k states. The multinomial model will be derived, and in particular, focus will be given to deriving and implementing these models for 2,3,4,5,6 and 7 states. Particular focus will be placed on examining the convergence patterns of the multinomial models, as well as comparing theoretical and actual convergence. A large amount of work has been done on investigating convergence patterns of tree based models. Work from Easton [19], Andricopoulos, Duck, Newton and Widdicks [4], Chen, Chen and Chung [15], Tian [58] and Omberg [50] is used.

Proofs showing that the Black-Scholes [8] partial differential equation can be derived from tree based models will then be described for the Binomial model. The chapter will conclude by briefly looking at other combinatorial lattice models, mainly by studying the implementation and convergence of the adaptive mesh model.

Chapter three looks at some of the options that depend on multiple sources of underlying risk. The analytical results for these will be used in later chapters when implementing the models discussed in the dissertation. Focus will be placed on Basket and Rainbow type options, where the derivative is dependent on more than one underlying share. Min and Max options, as well as Product options and Digital options will be some of those discussed.

This chapter will also investigate how tree based models are used when looking at convertible bonds. Convertible bonds are dependent on interest rates, credit and equity. This makes these types of options ideal for an investigation in multiple sources of risk. A number of lattice models will be investigated and implemented. The first model that will be discussed is the quadrinomial model [20], followed by the Tsiverotis and Fernandes[59] model. The Ayache, Forsyth and Vetzal [5] model; the Hung and Wang [29] model; and the Chambers Lu Model [14] models will be compared. Finally the k-factor spread model [3] will be discussed.

Chapter four will commence by investigating how trees can be used when problems exist that depend on more than one source of risk. We introduce the Product of Trees concept from Luenberger [43] and discuss the positives and negatives. In Luenberger's paper, he examines the binomial product of trees approach on two sources of underlying risk. This method will be investigated in depth, and extended into higher dimensions. This method can be extended in two ways: increasing the underlying trees' dimensions or increasing the sources of risk. The chapter will conclude by showing how the product of trees can be seen as the basis of the novel approach of looking at higher dimension option pricing.

Chapter five will expand on the work done by Luenberger, by looking at the work done by Boyle et al. [11], Kamrad and Ritchken [37] and Korn and Muller [40]. The main focus is increasing the number of underlying sources of risk. The models will be implemented, and again convergence and efficiency will be examined. The computational intensity will be a very important discussion point, as increasing the number of states and the number of sources of risk concurrently, causes a dramatic increase in computational effort.

Chapter six concludes the investigation with a summary of results obtained. The results of the dissertation will be discussed in detail, and whether the objectives of the dissertation were met will be discussed. Ideas for further investigation and analysis will be provided.

To conclude the introduction, this dissertation aims to provide a comprehensive overview of lattice models by covering the following main objectives:

1. Survey existing literature surrounding the derivation of Binomial trees.
2. Explore the expansion of the Binomial tree, both in terms of expanding the number of states and the number of sources of risk.
3. Investigate some options that are dependent on multiple sources of risk.
4. Provide an in-depth analysis on the Product of Trees approach [43].
5. Introduce three other methods that can value options on multiple sources of risk: the BEG model [11], the Kamrad and Ritchken approach [37] and the Decoupled approach [40].
6. Compare the accuracy and efficiency of the models when implemented on the options discussed using Matlab.
7. Investigate the convergence patterns and rates of these models.
8. Increase the amount of literature available on the topic of expanding the Binomial Tree.

In the conclusion of this dissertation, reference will be made back to the objectives outlined above.

The following flow diagram provides a view of the layout and interactions of the dissertation:

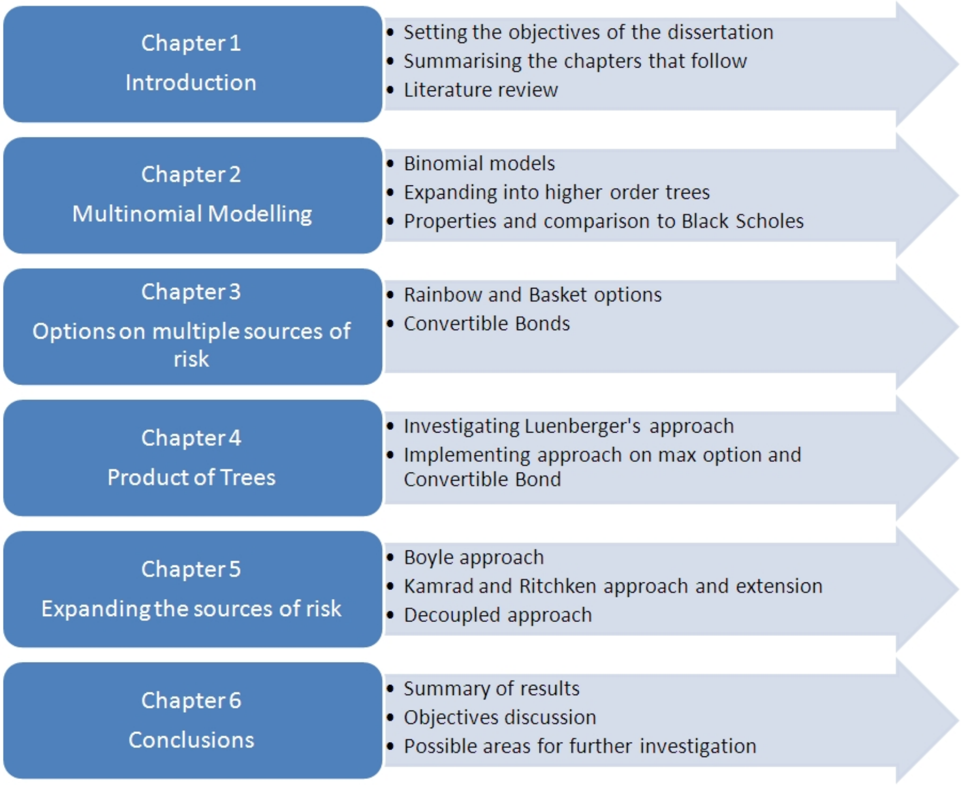

Chapter 1
Introduction
• Setting the objectives of the dissertation
• Summarising the chapters that follow
• Literature review
Chapter 2
Multinomial Modelling
• Binomial models
• Expanding into higher order trees
• Properties and comparison to Black Scholes
Chapter 3
Options on multiple sources of risk
• Rainbow and Basket options
• Convertible Bonds
Chapter 4
Product of Trees
• Investigating Luenberger's approach
• Implementing approach on max option and Convertible Bond
Chapter 5
Expanding the sources of risk
• Boyle approach
• Kamrad and Ritchken approach and extension
• Decoupled approach
Chapter 6
Conclusions
• Summary of results
• Objectives discussion
• Possible areas for further investigation

Figure 1.1: Flow of Disseration

Chapter 2

Multinomial Lattice Models

In this chapter the seminal work of Cox, Ross and Rubinstein (CRR) [17] is expanded into the k states of a multinomial lattice. Section 2.1 derives the Binomial lattice and looks at the differences between the CRR, Rendleman Bartter(RB) [54] and the Jarrow-Rudd [31] formulations as well as discussing the properties and assumptions of the model. Section 2.2 looks at expanding the Binomial lattice by increasing the number of states. The multinomial model is derived, and implemented on European options. Theoretical convergence is investigated along the lines of Zhou [23], and these are compared to actual convergence in Section 2.3. Section 2.4 investigates the Black-Scholes model as the limiting case for the Binomial model. Section 2.5 expands on the multinomial model by looking at combinatorial models, and in particular the adaptive mesh model by Ahn, Figlewski and Gao[1].

2.1 The Binomial Lattice

2.1.1 The One Period Binomial Model

If we consider a share that is currently worth $S(0)$, then after one time period, the share can be worth either $uS(0)$ or $dS(0)$ where u is the magnitude of the up movement ($u > 1$) and d is the magnitude of the down movement ($d < 1$). The probability of the share moving to $uS(0)$ is p and the probability of going to $dS(0)$ is $(1 - p)$. The nature of the variable p will be discussed in this section. This one step binomial tree can be seen in Fig. 2.1.

Hull [27] defines a derivative as "*An instrument whose price depends on, or is derived from, the price of another asset*". In the case of the one step tree, the underlying asset is the share.

Suppose there is a derivative on this share and we denote the value of this derivative as f. If the share price moves to $uS(0)$ the value of the derivative

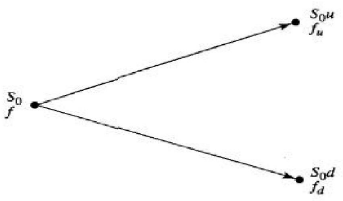


Figure 2.1: One step Binomial Tree

will be f_u and if the share price moves to $dS(0)$ the value of the derivative will be f_d.

Suppose also that there is a risk free interest rate r (this can be assumed to be cash).

We would like to replicate the payoff of the derivative by using only the underlying share and the risk free asset. Due to the no arbitrage property, if we can replicate the derivative, a unique price for the derivative will exist.

In order to replicate the payoff, we construct a portfolio Π, consisting of a long position in δ shares and one option. We need to calculate δ if the portfolio is riskless.

$$\Pi = \delta S(0) - f$$

$$\Pi_u = \delta S(0)u - f_u$$

$$\Pi_d = \delta S(0)d - f_d.$$

For the portfolio to be riskless, we have that

$$\Pi_u = \Pi_d.$$

This gives us the following:

$$\delta = \frac{f_u - f_d}{S(0)u - S(0)d}. \tag{2.1.1}$$

If we look at the present value of the portfolio we find it to be

$$(\delta S(0)u - f_u)e^{-rT},$$

and since this is equal to the cost of setting up the portfolio, we find after some algebra that:

$$f = e^{-rT}[pf_u + (1-p)f_d], \tag{2.1.2}$$

where

$$p = \frac{e^{rT} - d}{u - d}. \tag{2.1.3}$$

This can be generalised to the following when the time step is of length Δt:

$$f = e^{-r\Delta t}[pf_u + (1-p)f_d], \tag{2.1.4}$$

where

$$p = \frac{e^{r\Delta t} - d}{u - d}. \tag{2.1.5}$$

Eq. (2.1.4) is the backward induction equation. This equation is incredibly important, and can be interpreted as follows: The price of the derivative is equal to the expected value of the function of the derivative using the underlying share, discounted back using the risk free rate.

The only assumption that is required to hold is that no arbitrage opportunities exists [27]. If one sees the variable p as the probability of an up movement in the share price, the concept of the risk neutral world becomes important.

Hull [27] defines the risk neutral world as the world where "*all individuals are indifferent to risk*". This means that the expected return on any security is the risk free rate. This concept translates into risk neutral valuation, where the world can be assumed to be risk neutral, but where prices that are derived in the risk neutral world can be translated directly into the risky world.

2.1.2 Finding u and d

If we wish to specify the magnitude of the movements, i.e. u and d, we can do so by matching the volatility of the underlying share to the two parameters. From the previous section, we know that the expected value after one time step is the following:

$$pS(0)u + (1-p)S(0)d. \tag{2.1.6}$$

This must be equal to the expected return after one time period of the share, so we find:

$$pS(0)u + (1-p)S(0)d = e^{\mu\Delta t S(0)}. \tag{2.1.7}$$

We can then solve for p as:

$$p = \frac{e^{r\Delta t} - d}{u - d}. \tag{2.1.8}$$

The second moment is the variance σ^2. If we match the variance as well we find:

$$pu^2 + (1-p)d^2 - [pu + (1-p)d]^2 = \sigma^2 \Delta t. \tag{2.1.9}$$

Substituting Eq. (2.1.8) into the equation above, we find:

$$e^{\mu\Delta t}(u+d) - ud - e^{2\mu\Delta t} = \sigma^2 \Delta t. \tag{2.1.10}$$

If we use the Taylor expansion of $e^{\Delta t}$:

$$e^{\Delta t} = 1 + \Delta t + \frac{\Delta t^2}{2!} + ..., \tag{2.1.11}$$

and we ignore the terms of order Δt^2 or higher, we find:

$$(1+\mu\Delta t)(u+d) - ud - (2\mu\Delta t) = \sigma^2 \Delta t. \tag{2.1.12}$$

A solution to Eq. (2.1.10) is:

$$u = e^{\sigma\sqrt{\Delta t}} \tag{2.1.13}$$

$$d = e^{-\sigma\sqrt{\Delta t}}. \tag{2.1.14}$$

2.1.3 The Rendleman Bartter Binomial Lattice

Throughout this dissertation, models are introduced that offer extensions of both the CRR model, as well as models that extend the Rendleman Bartter (RB) model. This section will derive the RB model and highlight the differences between the RB and the CRR model.

In the RB case, instead of fixing the magnitude of the up and down movements the probabilities of the up and down movements are fixed to $\frac{1}{1+k^2}$ where $k \geq 0$. The difference in the final trees will be that the CRR tree will be symmetric, and due to the equal weight, the RB tree will not be symmetric.

The magnitude of the up and down movements in the RB tree becomes:

$$u = e^{(r-0.5\sigma^2)\Delta t + k\sigma\sqrt{\Delta t}} \tag{2.1.15}$$

$$d = e^{(r-0.5\sigma^2)\Delta t - k^{-1}\sigma\sqrt{\Delta t}}. \tag{2.1.16}$$

Another formulation is the Jarrow-Rudd formulation. This is a specific case of the RB model, where $k = 1$.

During the next few sections, all three formulations will be used. The table above shows the differences between the models:

Model	CRR	RB	JR
Probability	$\frac{e^{r\Delta t}-d}{u-d}$	$\frac{1}{1+k^2}$	0.5
Magnitude of up movement	$e^{\sigma\sqrt{\Delta t}}$	$e^{(r-0.5\sigma^2)\Delta t+k\sigma\sqrt{\Delta t}}$	$e^{(r-0.5\sigma^2)\Delta t+\sigma\sqrt{\Delta t}}$
Magnitude of down movement	$e^{-\sigma\sqrt{\Delta t}}$	$e^{(r-0.5\sigma^2)\Delta t-k^{-1}\sigma\sqrt{\Delta t}}$	$e^{(r-0.5\sigma^2)\Delta t-\sigma\sqrt{\Delta t}}$

Table 2.1: CRR, RB and JR comparison

2.1.4 Pricing an option using the Binomial Model

If we wish to use the Binomial model to derive the value of derivatives as defined in section we need to define a few equations. These will be used throughout the dissertation. Benninga and Wiener [7] show that one can see the value of an option as follows:

$$PO(n,j) = Max(su^j d^{n-j} - X, 0), \tag{2.1.17}$$

where n is the step and j is the state, u the magnitude of an up movement and d of a down movement. X is the strike and PO is short for Pay Off.

We can now define the price of a call option as follows:

$$C(X) = \sum_{j=0}^{n} p^j (1-p)^{j-n} \binom{n}{j} Max[su^j d^{n-j} - X, 0], \tag{2.1.18}$$

where

$$\binom{n}{j} = \frac{n!}{j!(n-j)!}$$

denotes the number of j up movements in n steps.

Finally we can write this as [7]:

$$C = e^{-r(T-t)} \sum_{i=j}^{n} \frac{n!}{i!(n-i)!} p^i (1-p)^{n-i} (Su^i d^{n-i} - X). \tag{2.1.19}$$

2.2 Multinomial Model

The previous section dealt with lattices where there were two possible movements, either up or down. These movements can be described as the different states of the world.

The method of expanding the Binomial model of two states into models of higher states is analogous to that of Boyle et al [11]. Boyle uses a moment matching approach that boils down to matching the moments of the discrete distribution to the continuous distribution we are trying to replicate. The next section aims to look at a general description of k state multinomial lattices using a moment matching technique. These include higher order trees, and we focus on the Binomial, Trinomial, Pentanomial, Hexanomial and Heptanomial lattices. The extension of a Binomial lattice to k-state variables and the convergence of the multinomial model has been looked at in detail by Zhou [23], and we draw from these texts to understand the model.

We will start off by looking at the derivation of the model using Jabbour et al. [30], and then in the subsequent sections we will look at different implementations of the model. For each implementation we will look at how the model performs by using it to price a simple European Put option described in Jabbour et al. with the following characteristics from Table 2.2:

Parameter	OTM	ATM	ITM
S	100	100	100
K	90	100	110
σ	0.3	0.3	0.3
r	0.05	0.05	0.05
T	1	1	1
Black Scholes Price Put	5.3081	9.3542	14.6553

Table 2.2: Vanilla Put Parameters

The problem is taken from Jabbour et al. in order for the results to be compared. The results will be scrutinised in the final convergence section.

2.2.1 Error

The concept of error is very important to understand in general, and especially when dealing with discrete type models modeling continuously distributed share prices, errors will creep in. In this section, we shall briefly look at the different types of error that can be encountered drawing from the work done by Shea [55].

2.2.1.1 Distribution Error

When using a lattice to find the option price, the lattice approximates the continuous log normal distribution with only a finite set of probabilities. Distribution error results from the difference between the discrete and continuous distributions [55]. Even though in our lattice examples, the discrete and continuous distributions' moments match, we still get distribution error.

2.2.1.2 Non-Linearity Error

The aim of a lattice is to find an approximation for the average price of an option over a certain period, by using a finite number of nodes. If the payoff function is non-linear, and we try to approximate this by a few number of nodes, the discrepancy that arises, is called the non-linearity error [55]. Initially we are only implementing the models on vanilla options, where the payoff functions are linear, we do not encounter this type of error with the multinomial models, but we will touch on the subject again when pricing the barrier options with the Adaptive Mesh Model.

In the next section where the models on multiple sources of risk are introduced and implemented on the non-linear payoff of a digital option, the non-linearity error will be apparent.

2.2.2 Derivation

We start by looking at the process followed by a stock price S :

$$dS = rSdt + \sigma SdW, \tag{2.2.1}$$

where S is the share price, σ is the volatility and r is the risk free interest rate. Using Ito, we let $X = ln(S)$ under the risk neutral measure and from here we know that $X(t)$ follows the following process:

$$dX(t) = \alpha dt + \sigma dW,$$

where $\alpha = r - 0.5\sigma^2$ and W is standard Brownian motion. (See Appendix A for Ito Calculus and how the process is derived).

We introduce the variable $\tilde{\mathrm{X}} = X_t - X_0 = ln(S_t/S(0))$ with mean αt and variance $\sigma^2 t$. We know that in a time period $S(0)$ can move to $u_j S(0)$ for $j = 1, 2...n$

Thus, the discrete distribution of the share price process looks like this :

$$\begin{cases} U_1 \text{ with probability } p_1 \\ U_2 \text{ with probability } p_2 \\ \qquad\qquad\qquad\qquad . \\ \qquad\qquad\qquad\qquad . \\ \qquad\qquad\qquad\qquad . \\ U_n \text{ with probability } p_n \end{cases}, \tag{2.2.2}$$

where $U_j = ln(u_j)$. u_j is the magnitude of a movement.

In order to find u and d in Section 2.1.2, we matched the expected value and variance of the lattice and the log normal distribution. These are the first two fo the so called moments. Appendix B gives greater insight into the Moment Generating function and Moments.

To specify the multinomial lattice we need to set the first k central moments of the discrete distribution equal to the first k central moments of the continuous log normal distribution. We start our derivation by looking at how to obtain the central moments.

Taking a look at the first moment, we find:

$$M_1 = \Sigma_{j=1}^{n} p_j U_j = m.$$

The k'th order central moment for the variable $\tilde{\text{X}}$ can be found using:

$$m_k = \Sigma_{j=1}^{n} p_j (U_j - m)^k = \Sigma_{j=1}^{n} p_j w_j^k \text{ for k = 1,2... L.}$$

where $w_j = U_j - m$. and L the number of moments to be matched.

Since we are working with the symmetrical normal distribution, all odd central moments will be equal to 0.

Thus, $m_1 = 0$ and the first moment, $m_1 = E[\tilde{X}] = \alpha\Delta t$, and the second, $m_2 = Var[\tilde{X}] = \sigma^2\Delta t$.

The other central moments can be obtained by applying the Taylor series expansion to the Moment Generating Function (MGF) as follows. (Again, see Appendix B for the mathematics behind the MGF).

$$M(t) = \Sigma_{j=0}^{\infty} M^{(j)}(0) \frac{t^j}{(j!)}.$$

If we look at the standard normal distribution, $W \sim N(0,1)$, we find a MGF that looks like this:

$$M(t) = e^{0.5t^2},$$

which we can write in the form:

$$M(t) = \Sigma_{j=0}^{\infty} \frac{t^{2j}}{(j!)2^j}. \tag{2.2.3}$$

$M^{(j)}(0)$ represents the j-order central moment

We know that the k^{th} central moment represents the coefficient before t^k in Eq. (2.2.3) multiplied by k!. Thus we have that:

$$m_k^W = \begin{cases} 0 \text{ if k is odd} \\ \frac{k!}{2^{(k/2)}(k/2)!} \text{ if k is even.} \end{cases} \tag{2.2.4}$$

To specify the multinomial model we need to solve the following set on non-linear equations with respect to p_j and w_j:

$$m_k = \Sigma_{j=1}^n p_j w_j^k = m_k^{\tilde{X}} \text{ for k = 1,2... L.} \tag{2.2.5}$$

In other words, our aim is to solve the following system of equations

$$[P]^T[W^k] = m_k^W \text{for k = 1,2...L,}$$

where $[W^k] = [w_j^k]_{j=1}^n$ and $[W^0] = [1]_{j=1}^n$.

To generalise, the procedure of specifying the n-order multinomial model is to set up n+1 non-linear equations, i.e. where L=n. The first equations will be that the sum of the probabilities add up to one and the other n will be found by matching the moments of the discrete distribution to the underlying continuous distribution. We will then end up with two solution vectors $[P] = [p_j]_{j=1}^n$ and $[W] = [w_j]_{j=1}^n$. One can see that this will not have a unique solution since we have 2n unknown and only n+1 equations. One should thus introduce additional constraints to find a unique solution.

A way to make the tree recombine, is to impose the additional constraint of

$$\Delta_{j+1} = \Delta j \text{ for j = 1,2,3...n-2,}$$

where $\Delta_i = w_{i+1} - w_i$.

One way of solving the system is by optimizing the following:

$$min_{[W],[P]}|[P^T][W^K] - m_K^W|,$$

subject to

$[P]^T[W^k] = m_k^W$ for k = 1,2,3...L and

$\Delta_{j+1} = \Delta j$ for j = 1,2,3...n-2

and where K is first even number greater than n.

Once the system is solved, the usual backward induction procedure can be followed to obtain the option price:

$$f = e^{-r\Delta t}\Sigma_{j=1}^n p_j f_j = e^{-r\Delta t}[P]^T[W], \tag{2.2.6}$$

where

$$f_j = f(Se^{(r-0.5\sigma^2)\Delta t + w_j\sigma\sqrt{\Delta t}}, t).$$

This equation is just a slightly different formulation to Eq. (2.1.4).

2.2.3 Properties of Multinomial Lattices

This section discusses two important properties of multinomial lattices, drawing from the work done by Yamada and Primbs [52]. Yamada and Primbs propose and prove the following two conditions:

1. If the skewness and kurtosis are fixed over each time period T/N, where N is the total time, then the multinomial lattice converges to a Gaussian distribution as T/N tends to 0.
2. If the annualised skewness and kurtosis are fixed then the multinomial lattice converges to a Poisson distribution with the same skewness and kurtosis.

Yamada and Primbs prove this for the case of the Pentanomial model.

2.2.4 The Binomial Model

We derived the specifications for the binomial model in the Section 2.1, but we can now write the equations in the Multinomial Form shown earlier like this:

$$\begin{aligned} p_1 + p_2 &= 1 \\ p_1 w_1 + p_2 w_2 &= 0 \\ p_1 w_1^2 + p_2 w_2^2 &= 1, \end{aligned} \tag{2.2.7}$$

where $w_1 = (\frac{lnu_1 - \alpha\Delta t}{\sigma\sqrt{\Delta t}})$, $w_2 = (\frac{lnu_2 - \alpha\Delta t}{\sigma\sqrt{\Delta t}})$, $p_1 = p$ and $p_2 = 1 - p$.

Including the constraint:

$$p_1 w_1^3 + p_2 w_2^3 = 0$$

gives us four equations and four variables, and solving for this by using the minimisation procedure outlined in the Multinomial derivation, we find the parameters in Table 2.3.
This is the Jarrow-Rudd specification of the Binomial model. From Fig. 2.2, Fig. 2.3 and Fig. 2.4 we can see that the model converges to the Black Scholes price as the number of iterations is increased. The Matlab code can be seen

p_1	0.5
p_2	0.5
w_1	-1
w_2	1

Table 2.3: Binomial Specification - Jarrow-Rudd

in Appendix C.1.

The convergence patterns are exactly the same for the ITM and OTM options. The ATM option has a slightly different pattern, but seems to converge at the same rate.

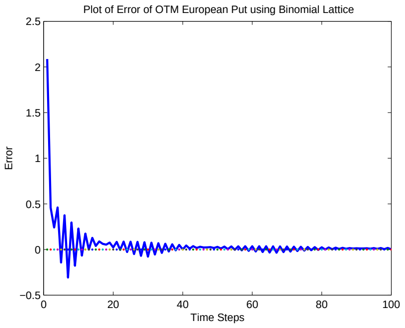


Figure 2.2: Convergence of OTM Binomial Model

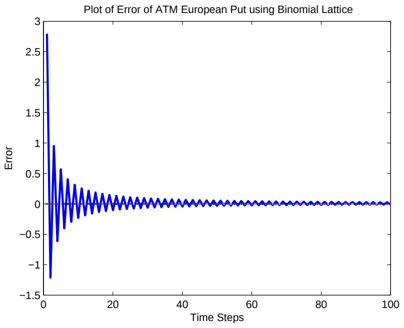


Figure 2.3: Convergence of ATM Binomial Model

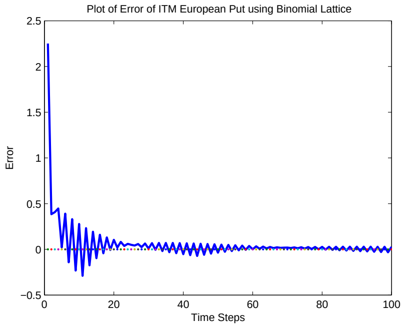


Figure 2.4: Convergence of ITM Binomial Model

2.2.5 The Trinomial Model

Trinomial Trees provide another way of looking at discrete representation of share price movements, and another way to price options. In this model we have three probabilities of share price movements, and three possible moves.

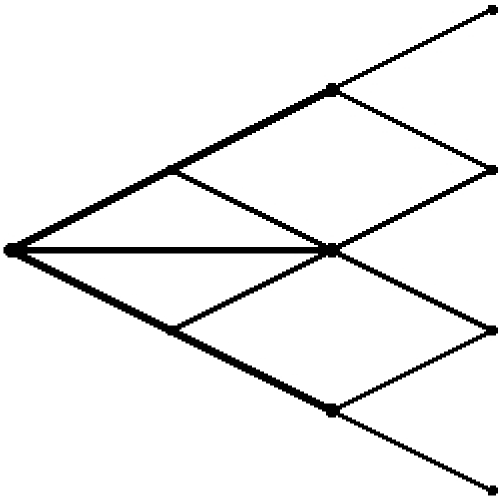


Figure 2.5: Two step Trinomial Tree

There are a number of ways of deriving the Trinomial model, but we will be looking at the most natural way of seeing it as being a two step Binomial model. Then we will use the moment matching technique to obtain the model. From the Binomial model Eq. (2.1.4) we had that:

$$f = e^{-r\Delta t}[pf_{u_1} + (1-p)f_{u_2}].$$

If we look at two steps of the Binomial model as one step, we find that the following must hold:

$$f_{u_1} = e^{-r\Delta t}[pf_{u_1u_1} + (1-p)f_{u_1u_2}]$$

$$f_{u_2} = e^{-r\Delta t}[pf_{u_1u_2} + (1-p)f_{u_2u_2}].$$

Substituting these equations into f, we find the following:

$$f = e^{-r\Delta t}[pe^{-r\Delta t}[pf_{u_1u_1} + (1-p)f_{u_1u_2}] + (1-p)e^{-r\Delta t}[pf_{u_1u_2} + (1-p)f_{u_2u_2}]]$$

$$f = e^{-2r\Delta t}[p^2 f_{u_1u_1} + 2p(1-p)f_{u_1u_2} + (1-p)^2 f_{u_2u_2}].$$

Since we are looking at two time steps, f simplifies to the following:

$$f = e^{-r\Delta t}[p^2 f_{u_1u_1} + 2p(1-p)f_{u_1u_2} + (1-p)^2 f_{u_2u_2}].$$

We know from the Binomial model and from the fact that we are using two time steps that

$$p = \frac{e^{r\Delta t/2} - d}{u_1 - u_2},$$

or

$$p = \frac{e^{r\Delta t/2} - e^{-\sigma\sqrt{\Delta t/2}}}{e^{\sigma\sqrt{\Delta t/2}} - e^{-\sigma\sqrt{\Delta t/2}}}.$$

Now, if we substitute p into f, we have our solution. We define the following:

$$\begin{aligned}
p_{u_1} = p^2 &= \left(\frac{e^{r\Delta t/2} - e^{-\sigma\sqrt{\Delta t/2}}}{e^{\sigma\sqrt{\Delta t/2}} - e^{-\sigma\sqrt{\Delta t/2}}}\right)^2 \\
p_{u_3} = (1-p)^2 &= \left(\frac{-e^{r\Delta t/2} + e^{\sigma\sqrt{\Delta t/2}}}{e^{\sigma\sqrt{\Delta t/2}} - e^{-\sigma\sqrt{\Delta t/2}}}\right)^2 \\
p_m &= 1 - p_{u_1} - p_{u_3},
\end{aligned} \tag{2.2.8}$$

This is the final specification for the Trinomial model.

Next we show that we can look at the model in terms of moment matching and get a similar answer. In one time period, we have the following situation:

$$S(0) = \begin{cases} u_1 S(0) \\ u_2 S(0) \\ u_3 S(0) \end{cases} \tag{2.2.9}$$

From the condition that the probabilities sum to one, and using moment matching, we can find the following set of equations:

$$\begin{aligned}
p_1 + p_2 + p_3 &= 1 \\
p_1 w_1 + p_2 w_2 + p_3 w_3 &= 0 \\
p_1 w_1^2 + p_2 w_2^2 + p_3 w_3^2 &= 1 \\
p_1 w_1^3 + p_2 w_2^3 + p_3 w_3^3 &= 0 \\
p_1 w_1^4 + p_2 w_2^4 + p_3 w_3^4 &= k \\
p_1 w_1^5 + p_2 w_2^5 + p_3 w_3^5 &= 0.
\end{aligned} \tag{2.2.10}$$

We now have 6 equations and 6 variables. We know that the fourth moment is the kurtosis and thus from our assumption that the share follows a log-normal distribution, we know that the kurtosis is 3.

Solving the set of equations gives us the parameters in Table 2.4.

p_1	$\frac{1}{6}$
p_2	$\frac{2}{3}$
p_3	$\frac{1}{6}$
w_1	$-\sqrt{3}$
w_2	0
w_3	$\sqrt{3}$

Table 2.4: Trinomial Specification

The probabilities and jump parameters obtained in Table 2.4 match those found by Jabbour et al. [30], Heston and Zhou [23] and Alford and Webber [2].

From Fig. 2.6, Fig. 2.7 and Fig. 2.8 we see that the convergence of the Trinomial model looks quite different to the Binomial model. The Matlab code can be seen in Appendix C.1.

As with the Binomial model, the convergence patterns are exactly the same for the ITM and OTM options. The ATM option has a slightly different pattern, but seems to converge at the same rate.

A certain amount of smoothing in the size of the oscillations has taken place when comparing to the Binomial model. This is particularly evident for the ATM option.

In the next three sections we show how the Pentanomial, Hexanomial and Heptanomial models are derived. The Quadnomial model is not investigated as, when using the moment matching technique explored above, it simplifies to exactly the trinomial model.

2.2.6 The Pentanomial Model

For the Pentanomial model, we continue exactly as with the Binomial and Trinomial models.

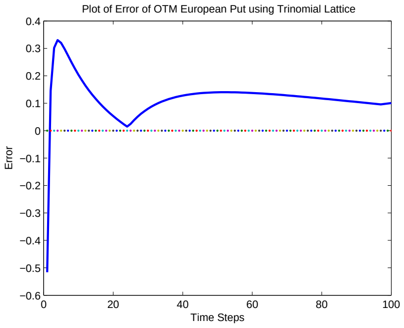


Figure 2.6: Convergence of OTM Trinomial Model

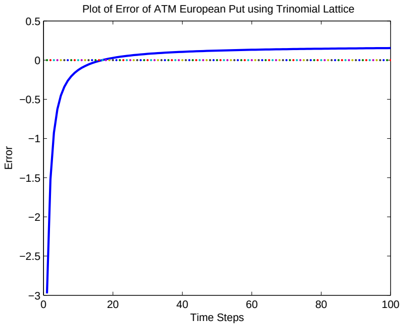


Figure 2.7: Convergence of ATM Trinomial Model

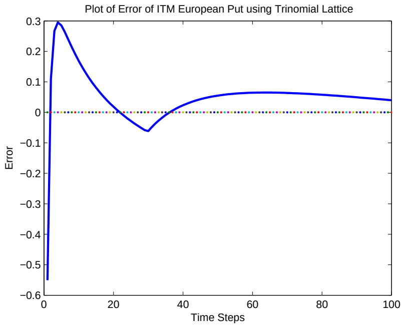


Figure 2.8: Convergence of ITM Trinomial Model

In one time period, we have the following situation:

$$S(0) = \begin{cases} u_1 S(0) \\ u_2 S(0) \\ u_3 S(0) \\ u_4 S(0) \\ u_5 S(0) \end{cases} \tag{2.2.11}$$

From the condition that the probabilities sum to one, and using moment matching, we can find the following set of equations:

$$\begin{aligned}
p_1 + p_2 + p_3 + p_4 + p_5 &= 1 \\
p_1 w_1 + p_2 w_2 + p_3 w_3 + p_4 w_4 + p_5 w_5 &= 0 \\
p_1 w_1^2 + p_2 w_2^2 + p_3 w_3^2 + p_4 w_4^2 + p_5 w_5^2 &= 1 \\
p_1 w_1^3 + p_2 w_2^3 + p_3 w_3^3 + p_4 w_4^3 + p_5 w_5^3 &= 0 \\
p_1 w_1^4 + p_2 w_2^4 + p_3 w_3^4 + p_4 w_4^4 + p_5 w_5^4 &= 3 \\
p_1 w_1^5 + p_2 w_2^5 + p_3 w_3^5 + p_4 w_4^5 + p_5 w_5^5 &= 0 \\
p_1 w_1^6 + p_2 w_2^6 + p_3 w_3^6 + p_4 w_4^6 + p_5 w_5^6 &= 15 \\
p_1 w_1^7 + p_2 w_2^7 + p_3 w_3^7 + p_4 w_4^7 + p_5 w_5^7 &= 0 \\
p_1 w_1^8 + p_2 w_2^8 + p_3 w_3^8 + p_4 w_4^8 + p_5 w_5^8 &= 105 \\
p_1 w_1^9 + p_2 w_2^9 + p_3 w_3^9 + p_4 w_4^9 + p_5 w_5^9 &= 0.
\end{aligned}$$

Now we have 10 equations in 10 unknowns.

Solving the set of equations gives us the parameters in Table 2.5.

p_1	0.013333
p_2	0.213334
p_3	0.546666
p_4	0.213334
p_5	0.013333
w_1	-2.738608
w_2	-1.369304
w_3	0
w_4	1.369304
w_5	2.738608

Table 2.5: Pentanomial Specification

The probabilities and jump parameters obtained in Table 2.5 match those found by Jabbour et al [30], but not Heston and Zhou [23] and Alford and Webber [2]. This is due to the solution of the underlying system not being unique. Further investigation can be done regarding the sensitivities and stability of these weights.

We can find the value of the option using backward induction. Fig. 2.9, Fig. 2.10 and Fig. 2.11 depict the convergence of the Pentanomial model. The Matlab code can be seen in Appendix C.1.

As with the Binomial and Trinomial models, the convergence patterns are exactly the same for the ITM and OTM options. The ATM option has a slightly different pattern, but seems to converge at the same rate.

The convergence pattern of the Pentanomial model seems closer to the Trinomial model than to the Binomial model.

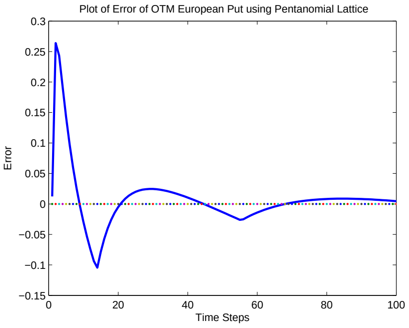


Figure 2.9: Convergence of OTM Pentanomial Model

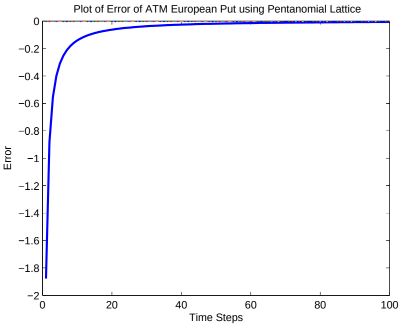


Figure 2.10: Convergence of ATM Pentanomial Model

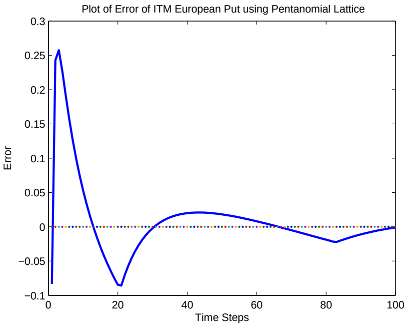


Figure 2.11: Convergence of ITM Pentanomial Model

2.2.7 The Hexanomial Model

For the Hexanomial model, we continue exactly as with the Pentanomial model.

In one time period, we have the following situation:

$$S(0) = \begin{cases} u_1 S(0) \\ u_2 S(0) \\ u_3 S(0) \\ u_4 S(0) \\ u_5 S(0) \\ u_6 S(0) \end{cases} \tag{2.2.12}$$

From the condition that the probabilities sum to one, and using moment matching, we can find the following set of equations:

$$\begin{aligned}
p_1 + p_2 + p_3 + p_4 + p_5 + p_6 &= 1 \\
p_1 w_1 + p_2 w_2 + p_3 w_3 + p_4 w_4 + p_5 w_5 + p_6 w_6 &= 0 \\
p_1 w_1^2 + p_2 w_2^2 + p_3 w_3^2 + p_4 w_4^2 + p_5 w_5^2 + p_6 w_6^2 &= 1 \\
p_1 w_1^3 + p_2 w_2^3 + p_3 w_3^3 + p_4 w_4^3 + p_5 w_5^3 + p_6 w_6^3 &= 0 \\
p_1 w_1^4 + p_2 w_2^4 + p_3 w_3^4 + p_4 w_4^4 + p_5 w_5^4 + p_6 w_6^4 &= 3 \\
p_1 w_1^5 + p_2 w_2^5 + p_3 w_3^5 + p_4 w_4^5 + p_5 w_5^5 + p_6 w_6^5 &= 0 \\
p_1 w_1^6 + p_2 w_2^6 + p_3 w_3^6 + p_4 w_4^6 + p_5 w_5^6 + p_6 w_6^6 &= 15 \\
p_1 w_1^7 + p_2 w_2^7 + p_3 w_3^7 + p_4 w_4^7 + p_5 w_5^7 + p_6 w_6^7 &= 0 \\
p_1 w_1^8 + p_2 w_2^8 + p_3 w_3^8 + p_4 w_4^8 + p_5 w_5^8 + p_6 w_6^8 &= 105 \\
p_1 w_1^9 + p_2 w_2^9 + p_3 w_3^9 + p_4 w_4^9 + p_5 w_5^9 + p_6 w_6^9 &= 0 \\
p_1 w_1^{10} + p_2 w_2^{10} + p_3 w_3^{10} + p_4 w_4^{1}0 + p_5 w_5^{1}0 + p_6 w_6^{10} &= 945 \\
p_1 w_1^{11} + p_2 w_2^{11} + p_3 w_3^{11} + p_4 w_4^{11} + p_5 w_5^{11} + p_6 w_6^{11} &= 0.
\end{aligned}$$

Now we have 12 equations in 12 unknowns.
Solving the set of equations gives us the parameters in Table 2.6.

p_1	0.003316
p_2	0.081193
p_3	0.415492
p_4	0.415492
p_5	0.081193
p_6	0.003316
w_1	-3.189031
w_2	-1.913419
w_3	-0.637806
w_4	0.637806
w_5	1.913419
w_6	3.189031

Table 2.6: Hexanomial Specification

The probabilities and jump parameters obtained in Table 2.6 match those found by Jabbour et al. [30].

Fig. 2.12, Fig. 2.13 and Fig. 2.14 depict the convergence of the Hexanomial model. The Matlab code can be seen in Appendix C.1.

Once again, as with the previous models, the convergence patterns are exactly the same for the ITM and OTM options. The ATM option has a slightly different pattern, but seems to converge at the same rate.

The convergence pattern of the Hexanomial model seems to match that of the Binomial model.

2.2.8 The Heptanomial Model

For the Heptanomial Model, we continue as earlier.

In one time period, we have the following situation:

$$S(0) = \left\{ \begin{array}{l} u_1 S(0) \\ u_2 S(0) \\ u_3 S(0) \\ u_4 S(0) \\ u_5 S(0) \\ u_6 S(0) \\ u_7 S(0) \end{array} \right. \tag{2.2.13}$$

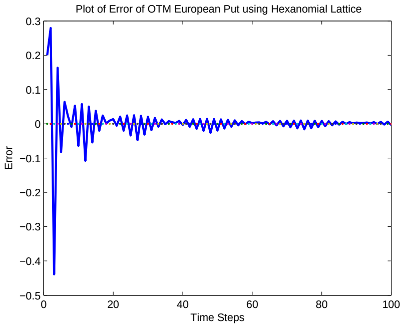


Figure 2.12: Convergence of OTM Hexanomial Model

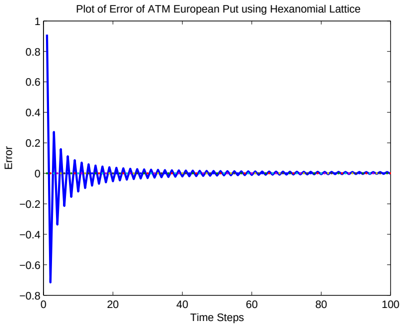


Figure 2.13: Convergence of ATM Hexanomial Model

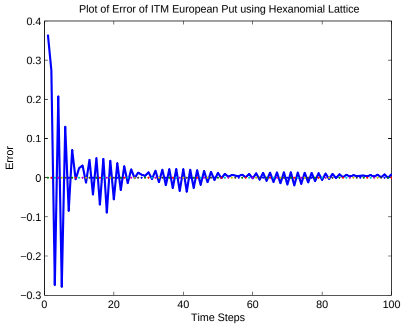


Figure 2.14: Convergence of ITM Hexanomial Model

From the condition that the probabilities sum to one, and using moment matching, we can find the following set of equations:

$$\begin{aligned}
p_1 + p_2 + p_3 + p_4 + p_5 + p_6 + p_7 &= 1 \\
p_1 w_1 + p_2 w_2 + p_3 w_3 + p_4 w_4 + p_5 w_5 + p_6 w_6 + p_7 w_7 &= 0 \\
p_1 w_1^2 + p_2 w_2^2 + p_3 w_3^2 + p_4 w_4^2 + p_5 w_5^2 + p_6 w_6^2 + p_7 w_7^2 &= 1 \\
p_1 w_1^3 + p_2 w_2^3 + p_3 w_3^3 + p_4 w_4^3 + p_5 w_5^3 + p_6 w_6^3 + p_7 w_7^3 &= 0 \\
p_1 w_1^4 + p_2 w_2^4 + p_3 w_3^4 + p_4 w_4^4 + p_5 w_5^4 + p_6 w_6^4 + p_7 w_7^4 &= 3 \\
p_1 w_1^5 + p_2 w_2^5 + p_3 w_3^5 + p_4 w_4^5 + p_5 w_5^5 + p_6 w_6^5 + p_7 w_7^5 &= 0 \\
p_1 w_1^6 + p_2 w_2^6 + p_3 w_3^6 + p_4 w_4^6 + p_5 w_5^6 + p_6 w_6^6 + p_7 w_7^6 &= 15 \\
p_1 w_1^7 + p_2 w_2^7 + p_3 w_3^7 + p_4 w_4^7 + p_5 w_5^7 + p_6 w_6^7 + p_7 w_7^7 &= 0 \\
p_1 w_1^8 + p_2 w_2^8 + p_3 w_3^8 + p_4 w_4^8 + p_5 w_5^8 + p_6 w_6^8 + p_7 w_7^8 &= 105 \\
p_1 w_1^9 + p_2 w_2^9 + p_3 w_3^9 + p_4 w_4^9 + p_5 w_5^9 + p_6 w_6^9 + p_7 w_7^9 &= 0 \\
p_1 w_1^{10} + p_2 w_2^{10} + p_3 w_3^{10} + p_4 w_4^{10} + p_5 w_5^{10} + p_6 w_6^{10} + p_7 w_7^{10} &= 945 \\
p_1 w_1^{11} + p_2 w_2^{11} + p_3 w_3^{11} + p_4 w_4^{11} + p_5 w_5^{11} + p_6 w_6^{11} + p_7 w_7^{11} &= 0 \\
p_1 w_1^{12} + p_2 w_2^{12} + p_3 w_3^{12} + p_4 w_4^{12} + p_5 w_5^{12} + p_6 w_6^{12} + p_7 w_7^{12} &= 10395 \\
p_1 w_1^{13} + p_2 w_2^{13} + p_3 w_3^{13} + p_4 w_4^{13} + p_5 w_5^{13} + p_6 w_6^{13} + p_7 w_7^{13} &= 0.
\end{aligned}$$

Now we have 14 equations and 14 unknowns. Solving the set of equations gives us the parameters in Table 2.7:

p_1	0.000802
p_2	0.026810
p_3	0.233813
p_4	0.477150
p_5	0.233813
p_6	0.026810
p_7	0.000802
w_1	-3.594559
w_2	-2.396373
w_3	-1.198186
w_4	0.000000
w_5	1.198186
w_6	2.396373
w_7	3.594559

Table 2.7: Heptanomial Specification

The probabilities and jump parameters obtained in Table 2.7 match those found by Jabbour et al. [30], but not Heston and Zhou [23] and Alford and Webber [2]. This is due to the solution of the underlying system not being unique. Further investigation can be done regarding the sensitivities and stability of these weights.

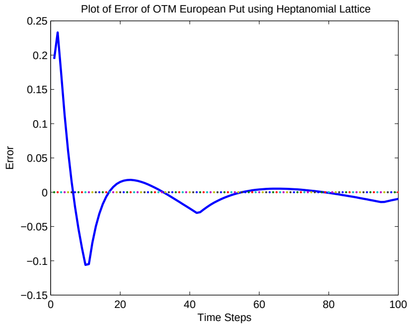


Figure 2.15: Convergence of OTM Heptanomial Model

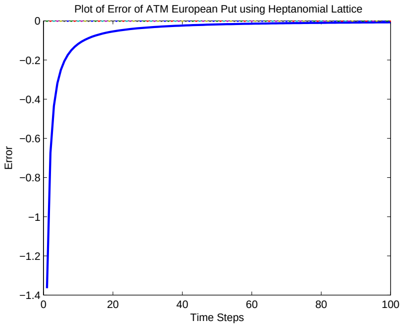


Figure 2.16: Convergence of ATM Heptanomial Model

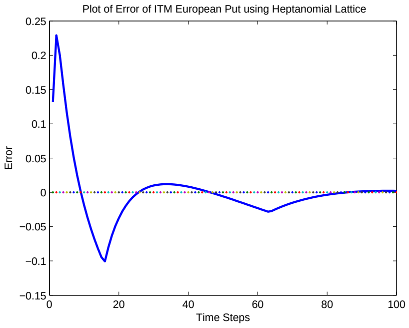


Figure 2.17: Convergence of ITM Heptanomial Model

We can find the value of the option using backward induction. Fig. 2.15, Fig. 2.16 and Fig. 2.17 depict the convergence of the Heptanomial model. The Matlab code is given in Appendix C.1.

The convergence patterns are exactly the same for the ITM and OTM options. The ATM option has a slightly different pattern, but seems to converge at the same rate. The convergence pattern of the Heptanomial model seems to match that of the Trinomial and Pentanomial models.

2.3 Convergence

In this section we will look at the simple illustrative example of the Multinomial model and scrutinise its convergence to the Black-Scholes price. Using a program written in Matlab (see Appendix C for code), we find the following results, which agree with the results from Jabbour et al. [30]. The graphs depicting convergence have already been shown in each of the previous sections.

Steps	Lattice:	2	3	5	6	7	BS
25	Value	5.39427	5.33274	5.32799	5.27405	5.32470	5.3081
	Error	0.08612	0.02464	0.01989	-0.03405	-0.01660	
	Time(seconds)	0.03662	0.19728	0.25742	0.23716	0.24198	
100	Value	5.30984	5.40859	5.31259	5.30433	5.29833	5.3081
	Error	0.00174	0.10049	0.00449	-0.00377	-0.00977	
	Time(seconds)	0.73475	1.93549	4.08907	5.66867	6.47078	
250	Value	5.30471	5.43184	5.30873	5.31327	5.30533	5.3081
	Error	-0.00338	0.12374	0.00063	0.00518	-0.00277	
	Time(seconds)	8.91598	21.17442	57.65794	85.77526	87.8444	

Table 2.8: Multinomial Lattice Convergence

We can observe a number of trends from the Figures and Table 2.8. It is interesting to note the convergence of the models from the figures. The models with an odd number of nodes (Trinomial, Pentanomial and Heptanomial) all converge with a steady tendency to the Black-Scholes price, whereas the even numbered nodes (Binomial and Hexanomial) converge with more frequent jumps from the positive to the negative side.

From Table 2.8 one can conclude that with a simple problem such as the European put, it is quite difficult to say if any of the models is superior to the others. It is clear that the greater the number of nodes, the longer the programs takes to run, but in most cases the greater the order of the lattice model, the faster the convergence. It thus makes sense to investigate the theoretical convergence, and compare that to the actual convergence by the multinomial model.

2.3.1 Theoretical Convergence

Zhou [23] investigates the theoretical convergence of multinomial models. The following proof stems from their work.

Before we state the formal theorem, we can define our model as we did in the Section 2.2.2 by matching the first q moments of the normal distribution. The theorem is then as follows:

Convergence Theorem

If a payoff function is 2q times continuously differentiable, then the multinomial approximation of the discrete solution $\tilde{f}$, converges to the continuous solution f at a rate of $O(\Delta t^{(q+1)/2-1})$. In mathematical terms this becomes:

$$\tilde{f} = f + O(\Delta t^{(q+1)/2-1}). \tag{2.3.1}$$

This implies that we will have a local error of $O(\Delta t^{(q+1)/(2)})$.

Proof:

We start by making a change of variables

$$U(h,t) = e^{-rt} f(S,t), \tag{2.3.2}$$

where

$$h = (ln(S) - (r - \sigma^2/2)t)/\sigma.$$

Our multinomial expression for Eq. (2.2.6) then becomes:

$$U(h,t) = \Sigma_{j=1}^{n} p_j U(h + w_j\sqrt{\Delta t}, t + \Delta t). \tag{2.3.3}$$

If we assume that we have a smooth payoff function, we known that U is 2q + 2 times differentiable [23]. We wish to show that the local error is $\Delta t^{(q+1)/2}$ and we can do this by showing :

$$U(h, t - \Delta t) = \Sigma_{j=1}^{n} p_j U(h + w_j\sqrt{\Delta t}, t) + O(\Delta t^{(q+1)/2}). \tag{2.3.4}$$

If we write the left-hand side into its Taylor expansion, we find :

$$U(h, t - \Delta t) = U(h,t) + \Sigma_{k=1}^{q/2} \Delta t^k \frac{\partial^k}{\partial t^k} \frac{U(h,t)}{k!} + O(\Delta t^{(q+1)/2}).$$

Denoting the first term of Eq. (2.3.4) by F and writing it as a Taylor expansion, we find:

$$F = \Sigma_{j=1}^{n} p_j U(h + w_j\sqrt{\Delta t}, t). \tag{2.3.5}$$

$$F = U(h,t) + \Sigma_{j=1}^{n}\Sigma_{k=1}^{q} p_j w_j^k \Delta t^{k/2} \frac{\partial^k}{\partial h^k}\frac{U(h,t)}{k!} + O(\Delta t^{(q+1)/2}). \qquad (2.3.6)$$

Substituting Eq. (2.2.5) into Eq. (2.3.6) we find :

$$F = U(h,t) + \Sigma_{k=1}^{q/2} \Delta t^k \frac{1}{2^k}\frac{\partial^{2k}}{\partial h^{2k}}\frac{U(h,t)}{k!} + O(\Delta t^{(q+1)/2}). \qquad (2.3.7)$$

From the heat equation we know that

$$U_{hh}(h,t)/2 = -U_t(h,t).$$

If we differentiate this w.r.t t once and w.r.t h twice we find

$$U_{hht}(h,t)/2 = -U_{tt}(h,t)$$

$$U_{hhhh}(h,t)/2 = -U_{hht}(h,t)$$

$$U_{tt}(h,t) = (\frac{-1}{2})(\frac{-1}{2})U_{hhhh}(h,t).$$

In general we can conclude that differentiating k times w.r.t t and differentiating 2k times w.r.t h we have

$$\frac{\partial^k}{\partial t^k}U(h,t) = (\frac{1}{2^k})(\frac{\partial^{2k}}{\partial h^{2k}})U(h,t).$$

Substituting this expression into Eq. (2.3.7) gives us

$$F = U(h,t) + \Sigma_{k=1}^{q/2} \Delta t^k \frac{\partial^k}{\partial t^k}\frac{U(h,t)}{k!} + O(\Delta t^{(q+1)/2}).$$

This is equal to the expression we found for Eq. (2.3.4) , thus proving that our approximation $U(h,t)$ has a local error of $O(\Delta t^{(q+1)/2})$.

2.3.2 Comparing Theoretical and Actual Convergence

In this section we will now look at how the theoretical convergence theorem that we proved holds up in practice. Theoretically, we can see from Table 2.9 the convergence rates.

Lattice	q	Convergence
Binomial	3	$O(\Delta)$
Trinomial	5	$O(\Delta^2)$
Pentanomial	7	$O(\Delta^3)$
Heptanomial	9	$O(\Delta^4)$
Hexanomial	11	$O(\Delta^5)$

Table 2.9: Theoretical Convergence

We now implement our European put example once again and compare the practical and theoretical convergence for each of the Lattice Models. For each model, we look at convergence when the number of time steps is doubled. We define the error ratio as the result when the absolute error for the "double time step run" is divided by the previous run. From Table 2.9 we know that for the Binomial model, our error ratio should be 2 , for the Trinomial 4, for the Pentanomial 8, for the Hexanomial 16 and for the Heptanomial 32. From Tables 2.10 to 2.14 we can clearly see that the models do not live up to the theoretical convergence we expected. In some cases the convergence is much larger, and in others much smaller, but no clear convergence exists. The main reason for this might be that the payoff functions are not smooth enough for the theorem to hold. It makes sense to try and introduce a way of smoothing the payoff functions in order to reach the expected convergence.

Steps	Binomial Value	Error	Error Ratio
10	5.53768786472352	0.22958786472352	
20	5.33144996543878	0.02334996543878	9.83247128675564
40	5.30945742124915	0.00135742124916	17.20170908869366
80	5.33268196832785	0.02458196832785	0.05522020169627
160	5.31874527073696	0.01064527073696	2.30919146494733
320	5.31260826800110	0.00450826800110	2.36127726532069

Table 2.10: Binomial Convergence

Steps	Trinomial Value	Error	Error Ratio
10	5.51199292644306	0.20389292644306	
20	5.36128475261777	0.05318475261777	3.83367255477145
40	5.43596220121732	0.12786220121732	0.41595367599979
80	5.42529706660100	0.11719706660100	1.09100171980105
160	5.43764598630592	0.12954598630592	0.90467539707664
320	5.43601413041785	0.12791413041785	1.01275743252712

Table 2.11: Trinomial Convergence

Steps	Pentanomial Value	Error	Error Ratio
10	5.27846667670083	0.02963332329917	
20	5.30302162631378	0.00507837368622	5.83519944181748
40	5.31858232100833	0.01048232100833	0.48447034604110
80	5.31622986910020	0.00812986910020	1.28935913716847
160	5.31211339253374	0.00401339253374	2.02568501133647
320	5.30781739316973	0.00028260683027	14.20132885665680

Table 2.12: Pentanomial Convergence

Steps	Hexanomial Value	Error	Error Ratio
10	5.24397859378882	0.06412140621118	
20	5.32194195361391	0.01384195361391	4.63239568631014
40	5.30496532962900	0.00313467037100	4.415760502911860
80	5.31657650221880	0.00847650221880	0.36980706075309
160	5.31287458717324	0.00477458717324	1.77533719905738
320	5.31068679917161	0.00258679917161	1.84575100596899

Table 2.13: Hexanomial Convergence

Steps	Heptanomial Value	Error	Error Ratio
10	5.20223307063900	0.06412140621118	
20	5.32313251784773	0.01503251784773	7.04252810030782
40	5.28444861357936	0.02365138642064	0.63558717363855
80	5.30714899448915	0.00095100551085	24.86987315089002
160	5.30332582207298	0.00477417792702	0.19919775202942
320	5.30845465303084	0.00035465303084	13.46154554427224

Table 2.14: Heptanomial Convergence

2.4 The Black-Scholes PDE as limiting case

For any study on option pricing to be complete, it needs to explore the seminal work of Black and Scholes [8]. This section will discuss the derivation of the Black Scholes PDE, as well as the underlying assumptions. Emphasis will be placed on seeing the Black Scholes equation as a limiting case for the Binomial method explored earlier.

2.4.1 The Black-Scholes Equation and Assumptions

If we look at the derivation of the Binomial Method in Section 2.1.1 once again, one can highlight the corresponding assumptions of the Black Scholes model.

When the portfolio of shares and options is constructed, there are two assumptions:

$$\Pi = \delta S(0) - f. \tag{2.4.1}$$

1. Assumption of liquidity, i.e. one can buy and sell shares/options.

2. Assume that there is no preference to buying or selling.

Once we look at the portfolio when time has passed, i.e. at the portfolio after an up or down movement, there is another assumption that there have not been any transaction costs.

$$d\Pi = \delta dS(0) - df.$$

If we then look at the driving process for the shares given in Eq. (2.2.1), we assume that the share returns are log normally distributed.

$$dS = rSdt + \sigma SdW.$$

Thus, using Ito again with $\delta = \frac{\partial f}{\partial S}$ we find:

$$d\Pi = (-\frac{\partial f}{\partial t} - 0.5\frac{\partial^2 f}{\partial S^2}\sigma^2 S^2)dt. \tag{2.4.2}$$

$$\tag{2.4.3}$$

Since we assume that no arbitrage is possible we know that the portfolio must earn the same return as the risk free rate:

$$d\Pi = r\Pi dt. \tag{2.4.4}$$

If we substitute Eq. (2.4.3) and Eq. (2.4.1) into Eq. (2.4.4) and if we assume that trading is continuous we find:

$$rf = r\frac{\partial f}{\partial S}S + +0.5\frac{\partial^2 f}{\partial S^2}S\sigma^2 + \frac{\partial f}{\partial t}.$$

This is the familiar Black-Scholes PDE [8].

The assumptions mentioned in the derivation are summarised as follows:

1. Assumption of liquidity, i.e. one can buy and sell shares/options
2. There are no transaction costs or taxes
3. The share returns are log normally distributed
4. There are no dividends during the life of the option
5. There are no arbitrage opportunities
6. Trading is contiguous
7. The risk free rate is constant

This method of deriving the Black-Scholes PDE is called the Arbitrage free of hedging approach. There are three other ways of deriving the equation:

1. The Probability of Default or CAPM Approach.
2. The Present Value Approach.
3. The Limiting case of the Binomial Approach.

It is interesting to see how the Black-Scholes equation can be seen as the limiting case in continuous time of the Binomial method. In this section the Black-Scholes PDE will be derived from both the Cox, Ross and Rubinstein Binomial method, as well as the k-state multinomial model.

2.4.2 The limiting case of the Binomial approach

As in the derivation of the Cox, Ross and Rubinstein tree, we have that the magnitude of up and down movements are defined as $u = e^{\sigma\sqrt{\Delta t}}$ and $d = e^{-\sigma\sqrt{\Delta t}} = \frac{1}{u}$. The share price at time t is defined as $S(t)$, and after one time period $S(t+\Delta t)$. Also $S(t+\Delta t) = uS(t)$ with probability p and $S(t+\Delta t) = dS(t)$ with probability $(1-p)$, where

$$p = \frac{e^{r\Delta t} - d}{u - d}. \tag{2.4.5}$$

From the valuation equation derived earlier, we know that a derivative $V(t, S(t))$ can be valued as follows:

$$\begin{aligned} V(t,S(t)) &= e^{-r\Delta t}[pV(t+\Delta t, uS(t)) + (1-p)V(t+\Delta t, dS(t))] \\ &= e^{-r\Delta t}[p(V(t+\Delta t, uS(t)) - V(t+\Delta t, dS(t))) + V(t+\Delta t, dS(t))]. \end{aligned} \tag{2.4.6}$$

Using the Taylor expansion, we can expand the expressions in Eq. (2.4.6), as follows:

$$\begin{aligned} V(t+\Delta t, uS(t)) &= V(t,S(t)) + \frac{\partial V(t,S(t))}{\partial S(t)}(uS(t) - S(t)) \\ &\quad + 0.5\frac{\partial^2 V(t,S(t))}{\partial S(t)^2}(uS(t)-S(t))^2 + \frac{\partial V(t,S(t))}{\partial t}\Delta t \\ &= V(t,S(t)) + \frac{\partial V(t,S(t))}{\partial S(t)}S(t)(u-1) \\ &\quad + 0.5\frac{\partial^2 V(t,S(t))}{\partial S(t)^2}S(t)(u-1)^2 + \frac{\partial V(t,S(t))}{\partial t}\Delta t. \end{aligned} \tag{2.4.7}$$

Similarly we can write:

$$\begin{aligned} V(t+\Delta t, dS(t)) &= V(t,S(t)) + \frac{\partial V(t,S(t))}{\partial S(t)}S(t)(d-1) \\ &\quad + 0.5\frac{\partial^2 V(t,S(t))}{\partial S(t)^2}S(t)(d-1)^2 + \frac{\partial V(t,S(t))}{\partial t}\Delta t. \end{aligned} \tag{2.4.8}$$

The exponential functions can also be expanded:

$$\begin{aligned} e^{r\Delta t} &= 1 + r\Delta t \\ u &= 1 + \sigma\sqrt{\Delta t} + 0.5\sigma^2\Delta t \\ d &= 1 - \sigma\sqrt{\Delta t} + 0.5\sigma^2\Delta t. \end{aligned}$$

It is clear that $(u-1)^2 = (d-1)^2 = \sigma^2\Delta t$ to $O(\Delta t)$. Thus if we subtract Eq. (2.4.8) from Eq. (2.4.7) we find:

$$V(t+\Delta t, uS(t)) - V(t+\Delta t, dS(t)) = (u-d)\frac{\partial V(t,S(t))}{\partial S(t)}S(t) \qquad (2.4.9)$$

Substituting back into Eq. (2.4.6), we find:

$$\begin{aligned}
V(t,S(t))e^{r\Delta t} &= p((u-d)\frac{\partial V(t,S(t))}{\partial S(t)}S(t)) + V(t+\Delta t, dS(t)) \\
&= \frac{1+r\Delta t-(1-\sigma\sqrt{\Delta t}+0.5\sigma^2\Delta t)}{u-d}(u-d)\frac{\partial V(t,S(t))}{\partial S(t)}S(t)) + V(t+\Delta t, dS(t)) \\
&= (r\Delta t+\sigma\sqrt{\Delta t}-0.5\sigma^2\Delta t)\frac{\partial V(t,S(t))}{\partial S(t)}S(t)) + V(t,S(t)) \\
&\quad + \frac{\partial V(t,S(t))}{\partial S(t)}S(t)(d-1) + 0.5\frac{\partial^2 V(t,S(t))}{\partial S(t)^2}S(t)(d-1)^2 + \frac{\partial V(t,S(t))}{\partial t}\Delta t \\
&= (r\Delta t+\sigma\sqrt{\Delta t}-0.5\sigma^2\Delta t)\frac{\partial V(t,S(t))}{\partial S(t)}S(t)) + V(t,S(t)) \\
&\quad + \frac{\partial V(t,S(t))}{\partial S(t)}S(t)(-\sigma\sqrt{\Delta t}+0.5\sigma^2\Delta t) + 0.5\frac{\partial^2 V(t,S(t))}{\partial S(t)^2}S(t)\sigma^2\Delta t \\
&\quad + \frac{\partial V(t,S(t))}{\partial t}\Delta t \\
&= (r\Delta t)\frac{\partial V(t,S(t))}{\partial S(t)}S(t)) + V(t,S(t)) + 0.5\frac{\partial^2 V(t,S(t))}{\partial S(t)^2}S(t)\sigma^2\Delta t + \frac{\partial V(t,S(t))}{\partial t}\Delta t.
\end{aligned}$$

Thus we can write:

$$\begin{aligned}
V(t,S(t))(1+r\Delta t) &= (r\Delta t)\frac{\partial V(t,S(t))}{\partial S(t)}S(t)) + V(t,S(t)) + 0.5\frac{\partial^2 V(t,S(t))}{\partial S(t)^2}S(t)\sigma^2\Delta t \\
&\quad + \frac{\partial V(t,S(t))}{\partial t}\Delta t \\
V(t,S(t))(r\Delta t) &= (r\Delta t)\frac{\partial V(t,S(t))}{\partial S(t)}S(t)) + +0.5\frac{\partial^2 V(t,S(t))}{\partial S(t)^2}S(t)\sigma^2\Delta t \\
&\quad + \frac{\partial V(t,S(t))}{\partial t}\Delta t.
\end{aligned}$$

If we divide by Δt on both sides we get:

$$rV(t,S(t)) = (r)\frac{\partial V(t,S(t))}{\partial S(t)}S(t)) + +0.5\frac{\partial^2 V(t,S(t))}{\partial S(t)^2}S(t)\sigma^2 + \frac{\partial V(t,S(t))}{\partial t}.$$

This is the familiar Black-Scholes PDE [8].

2.5 Combinatorial Trees

2.5.1 Adaptive Mesh Model

The adaptive mesh model (AMM) was introduced by Ahn, Figlewski and Gao[1] to improve efficiency and accuracy of a trinomial tree. The main idea behind the AMM is that we use a finer tree structure in the neighbourhood of some critical region (this finer mesh can be seen in Fig. 2.18. In most cases this will be the barrier of a Barrier Option. The reason for this is that problems occur when the initial asset price lies close to the barrier. The finer tree helps to alleviate this problem. In other words, the AMM tries to decrease the Non-Linearity error described earlier.

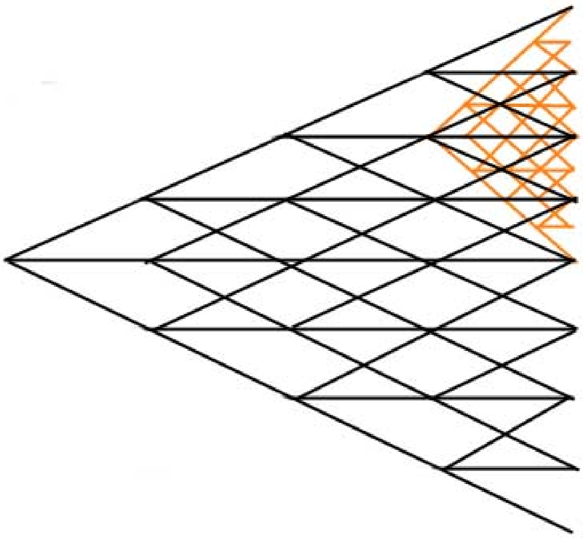


Figure 2.18: Adaptive Mesh Grafted to Trinomial Model

2.5.2 Derivation

Let $X = ln(S)$ under the risk neutral measure. We know that X(t) follows the following process:

$$dX(t) = \alpha dt + \sigma dW, \tag{2.5.1}$$

where $\alpha = r - 0.5\sigma^2$ and W is standard Brownian Motion.

The first step in deriving the Adaptive Mesh Model, is to derive the coarse trinomial tree as a base. It is necessary to define the tree slightly differently than in the multinomial section, because it then makes implementing the finer tree easier to understand. If we let Δt be the discrete time step (i.e. T/N) and h the price jump, we get the following:

$$X_{t+\Delta t} = \begin{cases} X+h & \text{with probability } p_{u_1} \\ X & \text{with probability } p_{u_2} \\ X-h & \text{with probability } p_{u_3} \end{cases} \tag{2.5.2}$$

To simplify the derivation we look at the mean adjusted value of the log of the asset price. In other words, introduce $X' = X - \alpha t$. This helps the derivation, because X' will be symmetric. If we look at the mean, variance, and by using the fact that the probabilities sum to one, we can deduce values for p_{u_1}, p_{u_2} and p_{u_3}.

$$p_{u_1} + p_{u_2} + p_{u_3} = 1$$

$$E[X'(t+\Delta t) - X'(t)] = 0 = p_{u_1}h + p_{u_2}0 + p_{u_3}(-h)]$$

$$E[(X'(t+\Delta t) - X'(t))^2] = \sigma\Delta t = p_{u_1}h^2 + p_{u_2}0 + p_{u_3}(-h)^2.$$

If we solve this set of three equations, we find that

$$\begin{cases} p_{u_1} &= \frac{\sigma^2 h}{2h^2} \\ p_{u_2} &= 1 - \frac{\sigma^2 h}{h^2} \\ p_{u_3} &= \frac{\sigma^2 h}{2h^2}. \end{cases} \tag{2.5.3}$$

Since X' is symmetrical as in the multinomial derivation, we can set its fourth moment equal to the kurtosis of the Normal Distribution. Another important aspect is that all the odd moments will be 0.

$$E[(X'(t+\Delta t) - X'(t))^4] = 3\sigma^4\Delta t^2 = p_{u_1}h^4 + p_{u_2}0 + p_{u_3}(-h)^4$$

Applying this to the solutions we found for the probabilities, we find:

$h = \sigma\sqrt{3\Delta t}$
$p_{u_1} = 1/6$
$p_{u_2} = 2/3$
$p_{u_3} = 1/6.$

From this we can write out the process:

$$X'_{t+\Delta t} - X'_t = \begin{cases} h & p_{u_1} = 1/6 \\ 0 & p_{u_2} = 2/3 \\ -h & p_{u_3} = 1/6 \end{cases} \tag{2.5.4}$$

From which we can write the process of X :

$$X_{t+\Delta t} - X_t = \begin{cases} \Delta t + h & p_{u_1} = 1/6 \\ \Delta t & p_{u_2} = 2/3 \\ \Delta t - h & p_{u_3} = 1/6 \end{cases} \tag{2.5.5}$$

Finally we can write the solution of the option price as follows:

$$\begin{split} f = exp(-r\Delta t)[p_{u_1} f(X + \alpha\Delta t + h, t + \Delta t) + \\ p_{u_2} f(X + \alpha\Delta t, t + \Delta t) + p_{u_3} f(X + \alpha\Delta t - h, t + \Delta t)]. \end{split} \tag{2.5.6}$$

This result agrees with the trinomial model we found using the moment matching technique and using two time steps of a binomial lattice illustrated earlier. We now need to generate the finer lattice onto the coarse lattice. We do this by grafting the fine lattice in the region of the barrier by implementing the fine lattice with each price step being half of the previous step. One should then also have time steps that are a quarter of the next step so as to keep the relationship between price and time steps constant. So, if we have mesh at a level called M, and the coarse lattice is called level 0, we have that $h_M = h/2^M$ and $\Delta t_M = \Delta t/4^M$.

Once the mesh is grafted onto the coarse lattice, it should also be connected back to the coarse lattice. One can also add more levels of mesh as required. As an example, we look at Fig. 2.19 by Rebib [53]. We can see that a two level mesh has been implemented. The first one starts at half the price step ($h_1 = h/2$) and a time step of a quarter of the original time step ($\Delta t_1 = \Delta t/4$). This is repeated for the second level, where we find that $h_2 = h/4$ and $\Delta t_2 = \Delta t/16$. The figure makes it clear that the initial price is then the middle node of the last level.

From Figlewski and Gao [1], we see as an example the adaptive mesh model being implemented on a European Down and Out Call option. The results are very favourable and looking at a comparison between a trinomial, a two level mesh, a four level and a eight level mesh, we find that the mesh model handles the Barrier option extremely well. Fig. 2.19 represents a two level mesh down and out barrier option.

Figlewski and Gao find that the error rate for the AMM model after 60 steps is equivalent to a Trinomial model after 5000 steps, proving that at least for a Barrier option, the AMM approach is superior.

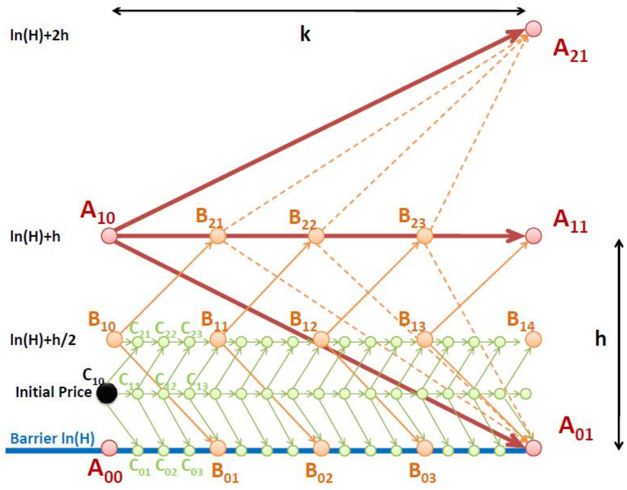


Figure 2.19: Adaptive Mesh Grafted to Trinomial Tree from Rebib [53]

Chapter 3

Options on multiple sources of risk

This chapter focuses on some of the most prevalent derivatives involving multiple sources of underlying risk. Some examples of options that depend on multiple sources of risk are: Basket options, Rainbow options and Convertible Bonds.

Basket and Rainbow options both depend on multiple source of underlying risk. Rainbow options can be defined as "all options whose payoff depends on more than one underlying risky asset; each asset is referred to as a colour of the rainbow" [51]. A multitude of call and put options on the maximum or minimum of a number of assets can be constructed. Basket options are similar to rainbow options, but where the payoff is a function of the underlying assets that are weighted. A common example of a rainbow option is an exchange option, where one asset can be exchanged for another, and an example of a basket option is a spread option. A Basket option can thus be seen as a vanilla option on several assets, whereas a rainbow option is an option with a more complex payoff structure.

Another type of security that involves multiple sources of underlying risk is the Convertible bond. Convertible bonds depend on three main sources of underlying risk: credit risk (probability of default), equity risk (share price movements) and interest rate risk (movements in the short rate). This makes these types of derivatives ideal to look at for this dissertation.

In the next sections we will discuss the work on multinomial modelling done by Boyle. Boyle applies the multinomial approach to valuing derivatives that depend on the minimum/maximum of several risky assets. These derivatives are an ideal example to illustrate the application of the product of trees in higher dimensions. This chapter starts with a discussion of the work done by Stulz [57]. Stulz derived analytical solutions for options on the minimum-maximum of two underlying assets, and thus we can use these exact results to compare convergence of our method. Magrabe [45] provided closed from solutions for European options with n underlying assets. Both Stulz and the other authors focus on European options, whereas the lattice model developed by Boyle can also accommodate the early exercise property of American options.

The different methods for valuing convertible bonds will be explored. Focus is given to the tree-based models. The Hung and Wang model [29] will be implemented using the product of tree approach in the next chapter.

3.1 Basket and Rainbow Options

The work done by Magrabe [45], and extensions by Stulz [57] make it possible to define analytical solutions for options on the maximum of several underlying assets. Heynen and Kat [24] derived analytical solutions for 2 asset Cash or Nothing or Digital options.

Magrabe was the first to value a so-called exchange option with the following payoff:

$$P(S,T) = max(S_1 - S_2, 0).$$

This can be seen as a call on share 1 with a stochastic strike. The value of this option is given by:

$$f(S,t) = S_1 e^{-q_1(T-t)} N(d_1) - S_2 e^{-q_2(T-t)} N(d_2),$$

where

$$d_1 = \frac{\frac{S_1 e^{(r-q_1)(T-t)}}{S_2 e^{(r-q_2)(T-t)}} + 0.5(\sigma_1^2 + \sigma_2^2 + 2\rho\sigma_1\sigma_2)^2(T-t)}{(\sigma_1^2 + \sigma_2^2 + 2\rho\sigma_1\sigma_2(T-t))\sqrt{(T-t)}}$$
$$d_2 = \frac{\frac{S_1 e^{(r-q_1)(T-t)}}{S_2 e^{(r-q_2)(T-t)}} - 0.5(\sigma_1^2 + \sigma_2^2 + 2\rho\sigma_1\sigma_2)(T-t)^2}{(\sigma_1^2 + \sigma_2^2 + 2\rho\sigma_1\sigma_2(T-t))\sqrt{(T-t)}}.$$

Stulz consequently extended the work to two underlying assets, and Johnson [33] extended the work to any number of underlying assets.

The dissertation will focus on the following three Rainbow/Basket type options:

1. European Max Option

$$P(S,T) = max(S_i - X, 0) \text{ for } i = 1 \text{ to } n. \qquad (3.1.1)$$

2. European Product Option

$$P(S,T) = max(\prod_{i=1}^{n}(S_i)^{\frac{1}{n}} - X, 0) \text{ for } i = 1 \text{ to } n.$$

3. European Digital Option

$$P(S,T) = \begin{cases} G & \text{if } S_i - X_i \geq 0 \\ 0 & \text{if } S_i - X_i < 0 \end{cases}. \tag{3.1.2}$$

3.2 Convertible Bonds

A convertible bond can be defined as "hybrid securities that confer upon the owner the right to receive a fixed income stream during the life of the convertible with the embedded right to forego the fixed income stream and irrevocably convert at the holder's option into a prescribed amount of the issuer's equity any time during the life of the instrument" [60].

The fact that the income generated from the convertible is uncertain, and that the bond is in fact both an interest and an equity option at the same time, makes this a very difficult instrument to price. There are a number of methods that can be used to model convertible bonds, and much research has been done on the topic, including papers by: Hariparsad [20], Tsiverotis and Fernandes [59], Ayache, Forsyth and Vetzal [5], Hung and Wang [29], Chambers Lu Model [14] and Barclays [3].

We will focus on the so-called equity valuation methods, as opposed to the firm valuation methods [20]. A few different models will be discussed in the next sections, as well as some practical results in the next chapter. The focus will be on tree based approaches. The product of trees approach will then be used to model the credit and market risk, and the results will be compared to the other approaches.

3.2.1 Quadrinomial Model

The Quadrinomial model explained in Hariparsad [20], splits the Convertible into separate components, but instead of modeling them separately, they are modelled together. This section will show that this approach is analogous to the product of trees approach introduced in the next chapter, where instead of having two underlying shares as risky assets, we have a share and interest rate.

Hariparsad assumes that the probabilities of up and down movements in both the underlying share price and interest rate trees are 0.5. We will show that the results he obtains are the same as obtained in Chapter 4, if the probabilities are taken to be 0.5.

We can derive the expected values and variances of both the share price and the interest rate over one time period:

$$
\begin{aligned}
E[S] &= 0.5(S_1 + S_2) \\
E[r] &= 0.5(r_1 + r_2) \\
\sigma_S &= 0.25(S_1 - S_2)^2 \\
\sigma_r &= 0.25(r_1 - r_2)^2
\end{aligned}
$$

Once again we know that $\sigma_{Sr} = \rho_{Sr}\sigma_S\sigma_r$, thus we find:

$$\sigma_{Sr} = 0.25\rho_{Sr}(S_1 - S_2)(r_1 - r_2). \tag{3.2.1}$$

If we shuffle around the definition for covariance, we can substitute and reach the result:

$$
\begin{aligned}
E[Sr] &= \sigma_{Sr} + E[S]E[r] \\
E[Sr] &= 0.25\rho_{Sr}(S_1 - S_2)(r_1 - r_2) + 0.25(S_1 + S_2)(r_1 - r_2 \\
E[Sr] &= 0.25(1 + \rho_{Sr})S_1 r_1 + 0.25(1 - \rho_{Sr})S_1 r_2 \\
&\qquad + 0.25(1 - \rho_{Sr})S_2 r_1 + 0.25(1 + \rho_{Sr})S_2 r_2.
\end{aligned}
$$

From this equation we can see the joint probabilities in Table 3.1:

p_{11}	$\frac{(1+\rho_{Sr})}{4}$
p_{12}	$\frac{(1-\rho_{Sr})}{4}$
p_{21}	$\frac{(1-\rho_{Sr})}{4}$
p_{22}	$\frac{(1+\rho_{Sr})}{4}$

Table 3.1: Joint Probabilities For Quadrinomial Tree

We can now use this to calculate the so called Rollback value:

$$RB = e^{-r_0\Delta t}[E[Sr]]. \tag{3.2.2}$$

This Rollback value is analogous to the backward induction formula from Section 2.1 (Eq. (2.1.4)).

In order to determine the value of the convertible bond, we need to look at the structure of the derivative. Once the bond matures, the investor must decide whether to covert the bond into shares, or to receive the redemption value of the bond. The number of shares that the investor can receive is determined by the conversion ratio (c) and the value of the bond by the redemption value F. We can thus write the value of the derivative at each node as follows:

$$Value = Max[Min(RB, C_t), \alpha S_t, P_t], \tag{3.2.3}$$

where C_t is the Call Price and P_t is the put value at time t.

3.2.2 Alternative Model Approaches

There are a multitude of different approaches to value the convertible bond available in modern literature. The following are some of the possible options:

1. Tsiverotis and Fernandes [59]

   In order to discuss the k-factor spread model in a later section, a quick introduction to the Tsiverotis and Fernandes models is provided here.

   Tsiverotis and Fernandes use the Black-Scholes model and introduce credit risk into the model by incorporating a credit spread. Tsiverotis and Fernandes argue that only the cash component of the derivative is exposed to credit risk, and the equity part is not. Their solution is to split the derivative into a cash part and an equity part.

   If we define the Cash part as CP and the Equity part as EP, then we know that the convertible bond f has the value $f = CP + EP$. Tsiverotis and Fernandes then use the Black-Scholes equation to value all the components.

2. Ayache, Forsyth and Vetzal [5]

   The Ayache, Forsyth and Vetzal (AFV) [5] approach, extends the TF model by allowing for some recovery on the share price if default occurs.

3. Hung and Wang [29]

   The Hung and Wang model incorporates a probability of default by building binomial trees for both risky and risk free interest rates. The risk free interest rates are used for the share price modeling whereas the risky interest rates are used in the bond modeling. Hung and Wang's model does not allow for correlation between the share price and interest rates. Hung and Wang's tree has 6 nodes:

Probability	Description
p_{011}	Default, S = 0, r goes up
p_{021}	Default, S = 0, r goes down
p_{112}	No Default, S goes up, r goes up
p_{122}	No Default, S goes up, r goes down
p_{212}	No Default, S goes down, r goes up
p_{222}	No Default, S goes down, r goes down

Table 3.2: Joint Probabilities For Hung Wang Model

The Hung and Wang model does not allow for correlation between the interest rate and equity. This hurdle is overcome by the Chambers and Lu model.

4. Chambers and Lu Model [14]

Chambers and Lu [14] also incorporate credit risk into their model, but provide for correlation between interest rates and equity.

The Chambers and Lu model calculates the Interest Rate tree using the Ho-Lee [25] model. The output is single period risk free interest rates. The magnitude of the up movement is given by $e^{2\sigma\sqrt{\Delta t}}$. By transforming the treasury term structure, and using a probability of an up movement as 0.5, the tree can be derived.

The second step is to define the equity risk tree. The tree is derived as with the binomial share price tree in Section 2.1.

The probabilities for the movements between the shares and interest rates are given in Table 3.3.

Description	Probability
S goes up, r goes up	$0.5(\hat{p} + \sqrt{\hat{p}(1-\hat{p})}\rho)$
S goes up, r goes down	$0.5(\hat{p} - \sqrt{\hat{p}(1-\hat{p})}\rho)$
S goes down, r goes up	$0.5(1-\hat{p} - \sqrt{\hat{p}(1-\hat{p})}\rho)$
S goes down, r goes down	$0.5(1-\hat{p} + \sqrt{\hat{p}(1-\hat{p})}\rho)$

Table 3.3: Joint Probabilities For Chambers and Lu Model

$\hat{p}$ is defined as the risk adjusted probability:

$$\hat{p} = \frac{e^{r\Delta t}/(1-\delta) - d}{u - d}, \tag{3.2.4}$$

where u and d are the up and down magnitudes and δ is the probability of default.

Once default occurs,the recovery rate λ is constant, but the probability of default can differ across each time period. To calculate the δ over the first time period, Chambers and Lu solve the following equation:

$$e^{-\hat{r_0}} = [1(1-\delta_1) + \lambda\delta_1]e^{-r_0}, \tag{3.2.5}$$

where $\hat{r_0}$ is the risky interest rate over the first time period, and r_0 is the riskless interest rate. If we solve for δ_1, we find the following:

$$\delta_1 = (1 - r_0 - \hat{r_0})/(1-\lambda) \tag{3.2.6}$$

If the recovery is 0, the default rate is just the spread between the risky and riskless bond yields.

This process can be expanded to the two year bonds and further, and thus the risky interest rate tree can be determined. The value of the convertible bond can be determined using Eq. (3.2.3). The Rollback value is now given by:

$$RB = (1-\delta_i)E[Sr] + \delta_i\lambda F, \tag{3.2.7}$$

for i = 1 to T, and where δ_i is the probability of default for period i.

5. K Factor Spread [3]

The K Factor credit spread model introduces a spread function that can be incorporated into the Tsiverotis and Fernandes model. The model, originally named the Barclays model, uses a function of the form:

$$CreditSpread = h \times S^{-k}, \tag{3.2.8}$$

where h is a function of the credit spread and k represents the relationship between the credit spread and the share price.

It is clear that there are a number of methods that can be used to value a convertible bond. Using a tree based approach helps with the visualisation of the problem, and can incorporate correlation between the risk factors, as well as the default rates. The K-Factor spread model would be the preferred approach as this takes all of the improvements from the previous models into account, but also includes a non constant credit spread. In the next chapter we will implement the Chambers and Lu model basing it on the product of trees visual method.

Chapter 4

Product of Trees

This chapter explores the product of trees concept by Luenberger [43]. His method can be seen as an alternative to Boyle's moment matching approach introduced in the previous section. Boyle's model will be discussed in detail in the next section. If more than one underlying risky asset needs to be considered at once, Luenberger models each separately using a simple Binomial tree, and then combines these to form the so-called product of trees. The main problem Luenberger states is that when the trees are presented together the number of nodes at each step is more than the number of underlying sources of risk, and thus the traditional replication arguments are invalid and the probabilities cannot be uniquely defined.

This hurdle is overcome by introducing an additional parameter using the marginal utility function and its properties. Section 1 of this chapter will step by step introduce Luenberger's solution to the problem, and Section 2 will expand the research by looking at the use of higher order trees instead of Binomial trees.

4.1 Luenberger's Product of Trees

4.1.1 One step Binomial Product of Trees Derivation

We start off by looking at two assets over a single time period. Suppose we have assets X and Y, and we can represent these assets on two binomial trees as discussed in Section 2.1. This is depicted in Fig. (4.1). The probability of an up movement $(p_1^X p_1^Y)$ and the probability of a down movement $(p_2^X p_2^Y)$ between the two assets may be correlated. These are shown in Table 4.1. We can then combine these two lattices to form the product lattice as in Fig. 4.2. We can combine the individual probabilities to form the joint probabilities.

We can now also define the magnitude of the up and down movements. As in

p_{11}	X moves up and Y moves up
p_{12}	X moves up and Y moves down
p_{21}	X moves down and Y moves up
p_{22}	X moves down and Y moves down

Table 4.1: Defining the probabilities

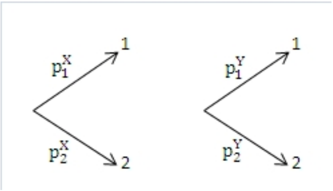


Figure 4.1: X and Y Trees

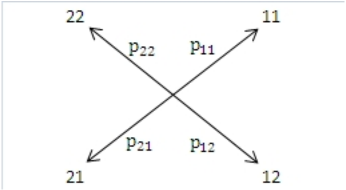


Figure 4.2: Product of X and Y

Section 2.1, we notate an up movement with u and a down movement with d. Thus we have the notation in Table 4.2.

u^X	Up movement for X
d^X	Down movement for X
u^Y	Up movement for Y
d^Y	Down movement for Y

Table 4.2: Defining the magnitudes

We also define the correlation between X and Y as ρ and the covariance of the log of the returns of X and Y as σ_{XY}. We can define the probabilities in terms of the probabilities of the individual lattices as follows:

$$
\begin{aligned}
p_{11} + p_{12} &= p_1^X \\
p_{21} + p_{22} &= p_2^X \\
p_{11} + p_{21} &= p_1^Y \\
p_{12} + p_{22} &= p_2^Y \\
p_{11} + p_{12} + p_{21} + p_{22} &= 1.
\end{aligned}
$$

If we investigate the covariance of the log of the return, we can add an additional equation. We know that the covariance can be written as [27]:

$$\sigma_{XY} = E[XY] - E[X]E[Y]. \tag{4.1.1}$$

We then just need to calculate $E[XY]$ and $E[X]E[Y]$.

If we define U^X, D^X, U^Y, D^Y as the natural logarithms of the respective movements we can derive the formula for the covariance as follows:

$$
\begin{aligned}
E[X] &= p_1^X U^X + p_2^X D^X \\
E[Y] &= p_1^Y U^Y + p_2^Y D^Y.
\end{aligned}
$$

We can then multiply these to form

$$
\begin{aligned}
E[X]E[Y] = {} & p_1^X p_1^Y U^X U^Y + p_1^X p_1^Y U^X U^Y \\
& + p_1^X p_2^Y U^X D^Y + p_2^X p_1^Y D^X U^Y + p_2^X p_2^Y D^X D^Y.
\end{aligned}
$$

We also know that:

$$E[XY] = p_{11} U^X U^Y + p_{12} U^X D^Y + p_{21} D^X U^Y + p_{22} D^X D^Y. \tag{4.1.2}$$

Thus we can write

$$
\begin{aligned}
\sigma_{XY} = {} & (p_{11} - p_1^X p_1^Y) U^X U^Y + (p_{12} - p_1^X p_2^Y) U^X D^Y \\
& + (p_{21} - p_2^X p_1^Y) D^X U^Y + (p_{22} - p_2^X p_2^Y) D^X D^Y.
\end{aligned}
$$

When the two underlying assets are independent, the probabilities can be written as:

$$p_{ij} = p_i^X p_j^Y \tag{4.1.3}$$

for $i = 1, 2$ and $j = 1, 2$.

We know from Section 2.1 that unique risk neutral probabilities can be determined for both X and Y using the risk free asset and the underlying asset. If we add a risk free asset to the product of trees, we do not have unique probabilities, as there are only three assets (X, Y and the risk free) and we have four nodes to specify the probabilities uniquely.

Luenberger solves this problem by incorporating a utility portfolio problem. The first question that stems from adding a utility function is how this uniquely determines a fourth equation, when the utility function can be a multitude of different functions. This question is answered by the fact that there are a number of conditions, that when occur, the probabilities are independent of the utility function. The next section will focus on determining these conditions.

4.1.2 Utility Functions

Suppose we have a portfolio with an initial wealth of W_0 and we can invest in assets X, Y and the risk free asset. If we invest x_X in asset X and x_Y in asset Y, we will have 1 - x_X - x_Y to invest in the risk free asset.We can define the returns over a single time period as R^X, R^Y and R^0. If we want to maximise our expected utility we can define the function we wish to maximise as follows [49], [42]:

$$maxE[U((x_XR^X + x_YR^Y + (1 - x_X - x_Y)R^0)W_0)]. \tag{4.1.4}$$

The risk neutral probabilities can then be calculated using the following formula:

$$q_{ij} = \frac{p_{ij}U'ij}{\Sigma_{k,l=1}^{2} p_{kl}U'_{kl}}. \tag{4.1.5}$$

Having derived the formula for determining the risk neutral probabilities, we can now discuss the conditions when this will introduce the additional relationship to make the probabilities unique.

Theorem 1 [42]
Suppose the risk neutral probabilities q_{ij} are determined using Eq. (4.1.5). Then the relation

$$\frac{q_{11}q_{22}}{q_{12}q_{21}} = \frac{p_{11}p_{22}}{p_{12}p_{21}}, \tag{4.1.6}$$

can be used if the following properties are satisfied:

1. The utility function is part of the exponential family of functions, i.e. $U(f(x)) = e^{(cf(x))}$.
2. The time step (Δt) is small.
3. Either x^X or x^Y is 0.

Eq. (4.1.6) is called the invariance equality.

Proof:
In order to prove the theorem, we fist need to define optimal independence. Marginal Utilities are optimally independent if:

$$U'_{11}U'_{22} = U'_{12}U'_{21}. \tag{4.1.7}$$

Using this as our outcome, we can now prove the theorem.

1. Suppose U is an exponential function, and we know that our Wealth W_{ij} after the first time period is $(R^0 + R_i^X + R_j^Y)W_0$. If we substitute this into our Utility function we find:

   $$\begin{aligned} U(W_{ij}) &= e^{-aW_{ij}} \\ &= e^{-a(R^0+R_1^X+R_2^X+R_1^Y+R_2^Y)W_0}. \end{aligned}$$

   Thus if we differentiate we find:

   $$\begin{aligned} U'_{11}U'_{22} &= (-ae^{(R^0+R_1^X+R_1^Y)})(-ae^{(R^0+R_2^X+R_2^Y)})U'_{11}U'_{22} \\ &= a^2e^{(R^0+R_1^X+R_1^Y+R_2^X+R_2^Y)}. \end{aligned}$$

   This is equal to $U'_{12}U'_{21}$.

2. For small Δt the return over one period must be close to one [42]. Thus we can write:

   $$R_i^X + R_j^Y + R^0 = 1 + r_i^X + r_j^Y + r^0,$$

   where r^X, r^Y and r^0 are small.

   This is approximately equal to:

   $$1 + r^X + r^Y + r^0 \approx (1 + r^X)(1 + r^Y)(1 + r^0). \tag{4.1.8}$$

   We can now write our Utility function using the approximation in Eq. (4.1.8):

   $$\begin{aligned} U'(W_{ij}) &= U'[(1 + r_i^X + r_j^Y + r^0)W(0)] \\ &\approx U'[W_{ij}] + U''[W_{ij}](r_i^X + r_j^Y + r^0)W(0) \\ &\approx U'[W_{ij}](1 + \frac{U''[W_{ij}]}{U'[W_{ij}]}W(0)(r_i^X + r_j^Y + r^0). \end{aligned} \tag{4.1.9}$$

   It is clear from the Eq. (4.1.9) that $U'_{11}U'_{22} = U'_{12}U'_{21}$.

3. If either x^X is 0 Eq. (4.1.4) becomes:

$$maxE[U((x_Y R^Y + (1 - x_Y)R^0)W_0)]. \tag{4.1.10}$$

Thus any changes in X do not affect the utility function, and thus $U'_{11} = U'_{21}$ and $U'_{22} = U'_{12}$. From this it is clear that $U'_{11}U'_{22} = U'_{12}U'_{21}$. Similarly if x^Y is 0, we have the same result.

We now have the additional equation we need to define the risk neutral probabilities uniquely. The next section will derive the product of trees from the standard representation of an Ito process.

4.1.3 Ito's Process

Deriving the product of trees using Ito's Process as a starting point follows a process very similar to that we used when matching moments in Chapter 2. Again suppose we have two assets that have log transformed processes X and Y. Using the share price process presented in Section 2.2 we can write:

$$\begin{aligned} dX &= \alpha_X dt + \sigma_X dW_X(t) \\ dY &= \alpha_Y dt + \sigma_Y dW_Y(t), \end{aligned}$$

where $W_i(t)$ are the standard Wiener processes and $\alpha_X = (\mu_X - 0.5\sigma_X^2)$ and $\alpha_Y = (\mu_Y - 0.5\sigma_Y^2)$.

We can discretise the process over a time period T, with n intervals of length $\Delta t = T/n$. We will do this for the asset X, as the process is analogous for both X and Y. The distribution of the log returns is log normal, with an expected value of $(\mu_X - 0.5\sigma_X^2)\Delta t$ and a variance of $\sigma_X^2 \Delta t$.

To build the two binomial trees for X and Y, we need to find the probabilities of up and down movements. We match the expected values to the continuous distribution's mean:

$$p_1^X U^X + p_2^X D^X = \alpha_X \Delta t. \tag{4.1.11}$$

Since $u = e^{\sigma_X \sqrt{\Delta t}}$, we know that $U^X = lnu = \sigma_X \sqrt{\Delta t} = -D^X$. If we substitute this into Eq. (4.1.11), we find the following:

$$\begin{aligned} p_1^X U^X - p_2^X U^X &= \alpha_X \Delta t \\ p_1^X \sigma_X \sqrt{\Delta t} - p_2^X \sigma_X \sqrt{\Delta t} &= \alpha_X \Delta t \\ p_1^X \sigma_X - p_2^X \sigma_X &= \alpha_X \sqrt{\Delta t}. \end{aligned}$$

Since $p_2^X = 1 - p_1^X$ we find that:

$$2p_1^X \sigma_X - \sigma_X = \alpha_X \sqrt{\Delta t} \tag{4.1.12}$$

$$p_1^X = \frac{\alpha_X \sqrt{\Delta t}}{2\sigma_X} + 0.5. \tag{4.1.13}$$

We now look at the covariance of the two assets.

$$\sigma_{XY} = E[XY] - E[X]E[Y]. \tag{4.1.14}$$

We know that $E[X] = \alpha_X \Delta t$ and $E[Y] = \alpha_Y \Delta t$. Thus $E[X]E[Y]$ is of the order Δt^2 and we can approximate the covariance using the following equation:

$$\sigma_{XY} = E[XY].$$

We now only need to calculate $E[XY]$:

$$E[XY] = (p_{11}U^X U^Y + p_{12}U^X D^Y + p_{21}D^X U^Y + p_{22}D^X D^Y).$$

Writing this in terms of up movements only we find:

$$\begin{aligned} E[XY] &= (p_{11}U^X U^Y - p_{12}U^X U^Y - p_{21}U^X U^Y + p_{22}U^X U^Y) \\ &= (p_{11}\sigma_X\sigma_Y - p_{12}\sigma_X\sigma_Y - p_{21}\sigma_X\sigma_Y + p_{22}\sigma_X\sigma_Y)\Delta t \\ &= (p_{11} - p_{12} - p_{21} + p_{22})\sigma_X\sigma_Y\Delta t. \end{aligned}$$

Thus we have

$$\sigma_{XY} = (p_{11} - p_{12} - p_{21} + p_{22})\sigma_X\sigma_Y\Delta t.$$

Since we know from [27] that

$$\rho = \frac{\sigma_{XY}}{\sqrt{\sigma_X^2 \sigma_Y^2}}, \tag{4.1.15}$$

and using the relationships between the probabilities in the underlying trees and the probabilities in the product tree, we find the equations we require:

$$\begin{aligned} p_{11}p_{12} - p_{21} + p_{22} &= \rho \\ p_{11} + p_{12} &= \frac{\alpha_X \sqrt{\Delta t}}{2\sigma_X} + 0.5 \\ p_{11} + p_{21} &= \frac{\alpha_Y \sqrt{\Delta t}}{2\sigma_Y} + 0.5 \\ p_{11} + p_{12} + p_{21} + p_{22} &= 1. \end{aligned}$$

We can now write the set of equations in terms of the probabilities we require for our product tree:

$$\begin{aligned}
p_{11} &= \frac{1+\rho}{4} + 0.25(\frac{\alpha_X}{\sigma_X} + \frac{\alpha_Y}{\sigma_Y})\sqrt{\Delta t} \\
p_{12} &= \frac{1-\rho}{4} + 0.25(\frac{\alpha_X}{\sigma_X} - \frac{\alpha_Y}{\sigma_Y})\sqrt{\Delta t} \\
p_{21} &= \frac{1-\rho}{4} + 0.25(\frac{\alpha_Y}{\sigma_Y} + \frac{\alpha_X}{\sigma_X})\sqrt{\Delta t} \\
p_{22} &= \frac{1+\rho}{4} - 0.25(\frac{\alpha_X}{\sigma_X} + \frac{\alpha_Y}{\sigma_Y})\sqrt{\Delta t}.
\end{aligned}$$

In order to determine the risk neutral probabilities, we can either use the four equations derived but replacing the expected value with the risk free expected value, i.e:

$$r - 0.5\sigma_X^2.$$

This results in the following set of equations that can be solved:

$$\begin{aligned}
q_{11} - q_{12} - q_{21} + q_{22} &= \rho \\
q_{11} + q_{12} &= \frac{(r-0.5\sigma_X^2)\sqrt{\Delta t}}{2\sigma_X} + 0.5 \\
q_{11} + q_{21} &= \frac{(r-0.5\sigma_Y^2)\sqrt{\Delta t}}{2\sigma_Y} + 0.5 \\
q_{11} + q_{12} + q_{21} + q_{22} &= 1.
\end{aligned}$$

After solving for this set of equations, we can calculate the invariance ratio using Eq. (4.1.6) to calculate the additional equation required:

$$\frac{q_{11}q_{22}}{q_{12}q_{21}} = \frac{(\frac{1+\rho}{4})^2 - \frac{1}{16}(\frac{r-0.5\sigma_X^2}{\sigma_X} + \frac{r-0.5\sigma_Y^2}{\sigma_Y})^2\Delta t}{(\frac{1-\rho}{4})^2 - \frac{1}{16}(\frac{r-0.5\sigma_X^2}{\sigma_X} - \frac{r-0.5\sigma_Y^2}{\sigma_Y})^2\Delta t}. \tag{4.1.16}$$

4.2 Increasing the number of states

The natural extension of the product of trees is to investigate whether one can increase the number of states in the underlying trees. Luenberger notes in his paper that the results derived using binomial underlying trees, can be applied to higher state trees. We will still focus on only two assets, X and Y. If each asset's underlying tree has two successor nodes as with the binomial product tree, the required number of risk neutral probabilities is 2 × 2 = 4. Thus we need 4 equations to be able to specify unique risk neutral probabilities. Since we have two underlying assets, we have two equations, and the sum of probabilities equal one relationship provides another.

In general, if asset X and asset Y have k and l successor nodes respectively, we require $k \times l$ probabilities and we have 3 so we need an additional $k \times l - 3$ equations. If we can show that the invariance ratio holds for higher order states, and that the invariance ratio provides $k \times l - 3$ equations, the additional equations can be found. The invariance Eq. (4.1.6) can be written as:

$$\frac{q_{ii}q_{jj}}{q_{ij}q_{ji}} = \frac{p_{ii}p_{jj}}{p_{ij}p_{ji}} \tag{4.2.1}$$

for all $i \neq j$.

These equations are not independent, but they have rank of $k \times l - 3$, and thus will define the necessary equations to extend the Product of trees approach to higher state trees.

4.3 Implementation

To illustrate how the product of trees approach works, we will implement the model on a derivative with two underlying assets, and a derivative with three underlying sources of risk. A convertible bond will also be valued using the Hung and Wang approach. For ease of illustration, the implementation will only be done on trees with three steps. The two asset example is a max option on two underlying shares. For this example, we assume that the shares are not independent. For the option on three sources of risk, we have a max option on three shares. Once again the log returns of the shares are correlated. Fig. 4.3 and Fig. 4.4 depict the underlying binomial trees and the backward induction process for the two derivatives.

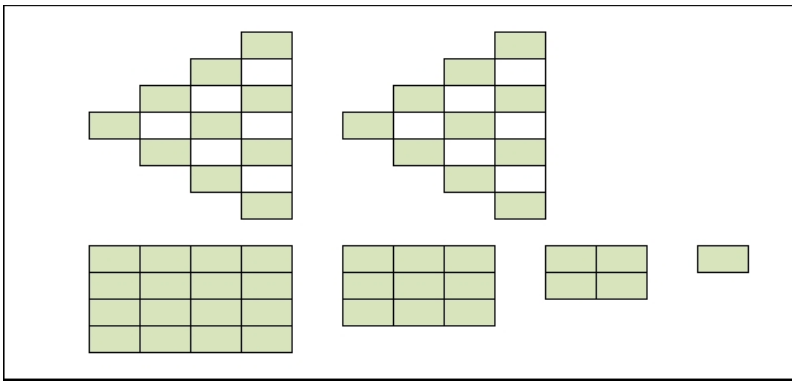


Figure 4.3: 2 State Product of Trees Layout for two underlying assets

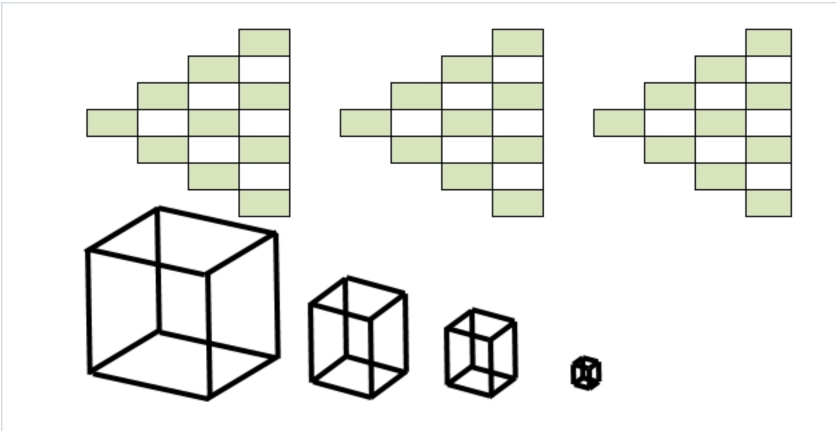


Figure 4.4: 2 State Product of Trees Layout for three underlying assets

4.3.1 Two Dimensional Max Option

Table 4.3 depicts the specifications of the option to be considered. This problem comes from the paper by Boyle [9]. This assists with comparison purposes. Fig. 4.5 depicts the binomial trees for each asset. The max option payoff was given by Eq. 3.1.1 in Section 3.1.

Using the probability equations from the previous section, we can calculate q_{11}, q_{22}, q_{12} and q_{21} as in Table 4.4.

Fig. 4.6 shows that the Invariance quantity converges to its actual value. Using the probabilities defined, we can calculate backwards to get the final option value. Fig. 4.7 depicts the process. If we define the first matrix as A where $A(1,2)$ indicates matrix A, row 1 column 2. The numbers on the outside of the matrix correspond to Fig. 4.5. Matrix A is calculated by using the final nodes of the Binomial trees and the payoff. In other words, $A(1,1) = max(S_1(T) - X, S_2(T) - X) = max(19.49, 12.12) = 19.49$.

S_1	40
S_2	40
X	40
σ_1	0.2
σ_2	0.3
ρ	0.5
r	0.04879
T	0.58333

Table 4.3: Two Dimensional Max Option Specifications

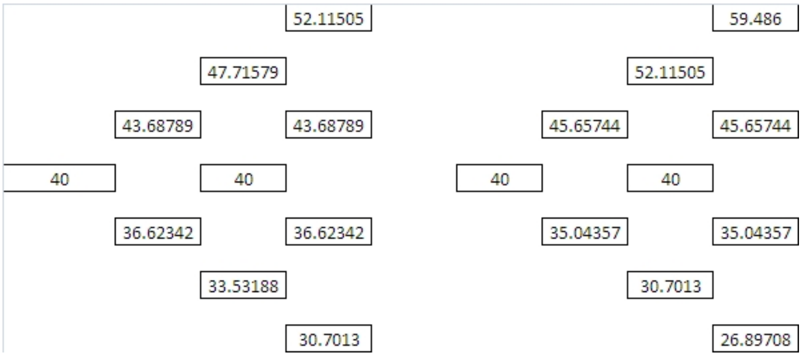


Figure 4.5: Underlying Binomial Trees for 2D Max Option

Probability	Movement	Value
q_{11}	uu	0.392261641
q_{12}	ud	0.13947626
q_{21}	du	0.357738359
q_{22}	dd	0.11052374

Table 4.4: 2D Risk neutral probabilities

Once matrix A is populated, the others can be defined using backward induction. For example $B(1,1) = e^{(-rt)}(q_{11}A(1,1) + q_{12}A(1,2) + q_{21}A(2,1) + q_{22}A(2,2))$. The final value of 9.06. This approach will converge to the true price of 5.488 and this will be shown in the next chapter.

4.3.2 Three Dimensional Max Option

Table 4.5 depicts the specifications of the option to be considered. This problem comes from the paper by Boyle et al [11]. Fig. 4.8 depicts the binomial trees for each asset. The max option payoff was given by Eq. (3.1.1) in Section 3.1.

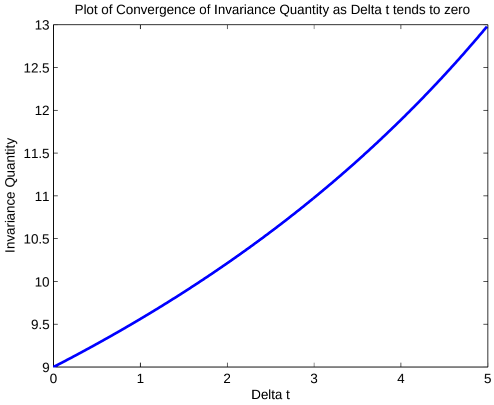
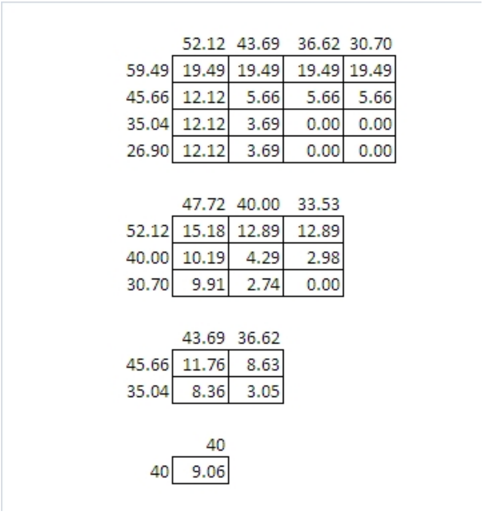


Figure 4.6: Convergence to limiting value of invariance quantity



	52.12	43.69	36.62	30.70
59.49	19.49	19.49	19.49	19.49
45.66	12.12	5.66	5.66	5.66
35.04	12.12	3.69	0.00	0.00
26.90	12.12	3.69	0.00	0.00

	47.72	40.00	33.53
52.12	15.18	12.89	12.89
40.00	10.19	4.29	2.98
30.70	9.91	2.74	0.00

	43.69	36.62
45.66	11.76	8.63
35.04	8.36	3.05

	40
40	9.06

Figure 4.7: Backward Induction using product of trees approach

Using the approach from Section 4.2, we can calculate q_1, q_2, q_3, q_4, q_5, q_6, q_7 and q_8.

S_1	100
S_2	100
S_2	100
X	100
σ_1	0.2
σ_2	0.2
σ_3	0.2
ρ_{12}	0.5
ρ_{13}	0.5
ρ_{23}	0.5
r	0.1
T	1

Table 4.5: Three Dimensional Max Option Specifications

Probability	Movement	Value
q_1	uuu	0.39910254
q_2	uud	0.091367513
q_3	udu	0.091367513
q_4	duu	0.091367513
q_5	udd	0.033632487
q_6	dud	0.033632487
q_7	ddu	0.033632487
q_8	ddd	0.22589746

Table 4.6: 3D Risk neutral probabilities

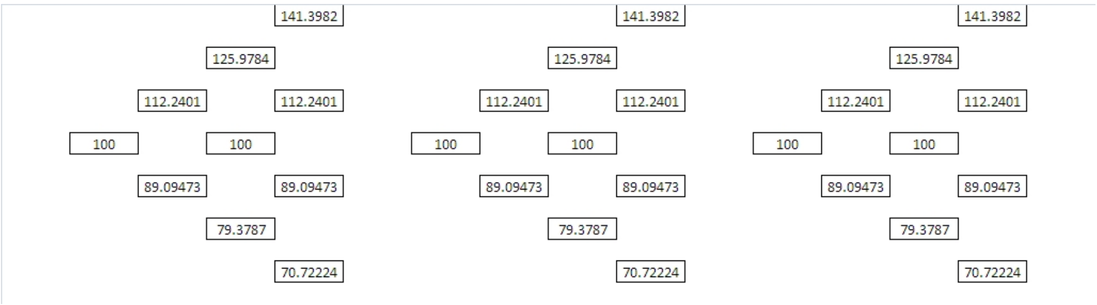


Figure 4.8: Underlying Binomial Trees for 3D Max Option

Fig. 4.9 shows that the Invariance quantity converges to its actual value.

Once again,as in the case of the 2 dimensional problem, once we have the terminal nodes we can calculate backwards to get the final option value. Fig. 4.10 depicts the process. Instead of having matrices that decrease in size, we have cubes. These cubes can be seen as n × n × n matrices as in Fig. 4.10. The

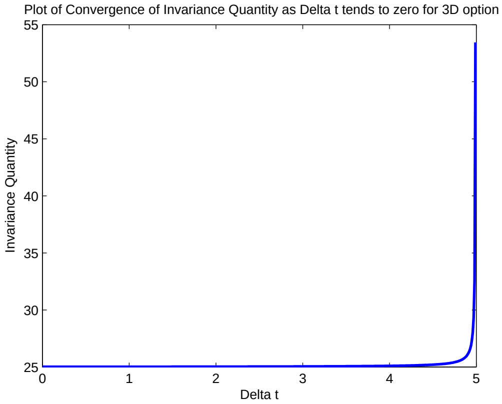


Figure 4.9: Convergence to limiting value of invariance quantity for 3D option

first 4 matrices should be seen as a cube. If we define the first cube as A where $A(1,2,1)$ indicates cube A, matrix 1, row 2 column 1. Cube A is calculated by using the final nodes of the binomial trees and the payoff. Thus $A(1,2,3) = max(S_1(T)-X, S_2(T)-X, S_3(T)-X) = max(141.40, 112.24, 89.09) = 141.40$.

Once cube A is populated, the others can be defined using backward induction. For example $B(1,1,1) = e^{(-rt)}(q_1A(1,1,1) + q_2A(1,1,2) + q_3A(1,2,1) + q_4A(1,2,2) + q_5A(2,1,1) + q_6A(2,1,2) + q_7A(2,2,1) + q_8A(2,2,2))$

The final price is then given as 19.84. This approach will converge to the true price of 22.672 and this will be shown in the next chapter.

4.3.3 Three Dimensional Convertible Bond

The table below depicts the specifications of the convertible bond to be considered. The problem comes from the paper by Hung and Wang [29].

S_0	30
$Call$	105
F	100
δ	0.32
σ_S	0.23
σ_2	0.2
σ_2	0.2
ρ_{12}	0.5
ρ_{13}	0.5
ρ_{23}	0.5
r_0	0.1
T	3
w	3

Table 4.7: Convertible Bond Specifications

As noted in the previous chapter, Convertible Bonds are dependent on Equity, Interest rate and Credit risk. The Chambers and Lu model was explained in the previous chapter. Fig. 4.11 depicts the share price and the riskless interest rate trees.

Using Eq. (3.2.6) we can calculate the default probabilities for each of the three time periods:

δ_1	0.0717
δ_2	0.0774
δ_3	0.0844

Table 4.8: CB Default Probabilities

Since the convertible bond is dependent on the interest rate, probability measures will be different for each period. The probability is calculated using the following equation:

$$\hat{p} = \frac{e^{r\Delta t}/(1-\delta_t) - d}{u - d}, \tag{4.3.1}$$

$\hat{p}$ can then be used to calculate the joint probabilities using the equations from Table 3.3. This equation was first given in Eq. (3.2.4).

Fig. 4.12 depicts the process of valuing the option. The first two matrices depicts the value of the option at maturity. The left-hand side matrix if no default occurs, and the right-hand side matrix if default occurs. Each cell in the left-hand side matrix is the following function:

$$max(cS_3, F), \tag{4.3.2}$$

where the c is the conversion factor. The right-hand side matrix is the recovery rate δ times the face value F.

In order to do backward induction, the probabilities for each time period need to be calculated. These are given in Fig. 4.13. First of all the $\hat{p}$ are calculated for each time period. Then the individual probabilities can be calculated using the Table 3.3. As an example, the first cell (p_{11}) for Time 2 can be calculated using the equation:

$$0.5(\hat{p} + \sqrt{\hat{p}(1-\hat{p})}\rho) \tag{4.3.3}$$

Once the probabilities are calculated the backward induction can be done. We can calculate the first cell of the second time period matrix as an example. The value of the cell is as follows:

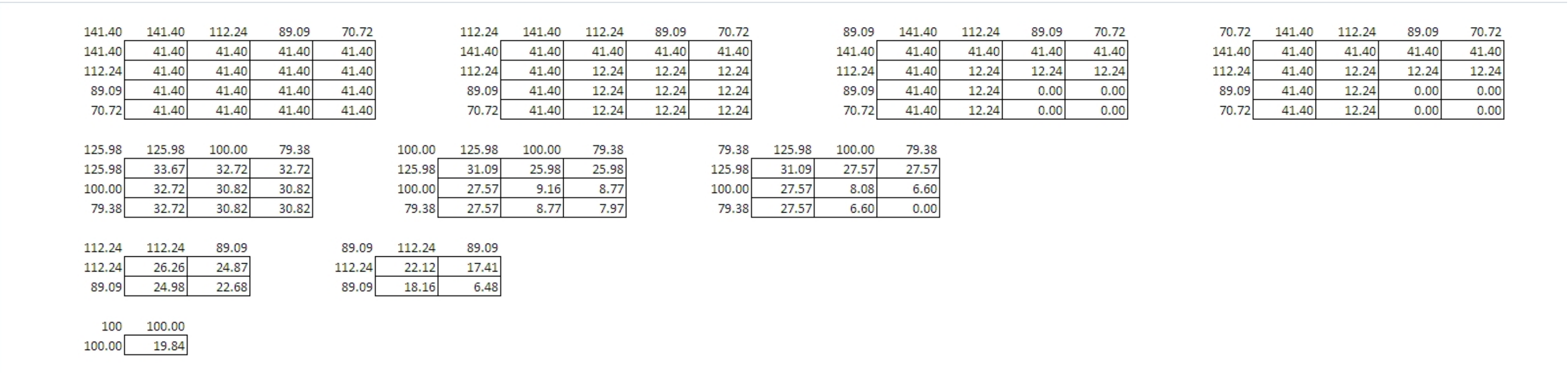
$$Max[Min(RB, Call), ConversionValue, Put], \tag{4.3.4}$$

where the RB (Roll Back Value) can be calculated as follows:

$$RB = (1-\lambda_3)[p_{11}*A(1,1)+p_{12}*A(1,2)+p_{21}*A(2,1)+p_{22}*A(2,2)]+\lambda_3 B(1,1),$$

where A is the first-right hand side matrix, and B is the first left hand side matrix. In this case $p_{11} - p_{22}$ are the first row of probabilities from Fig. 4.13. Once the backward induction process is complete, the final value of the Bond is 93.15. This corresponds to the value found by Chambers and Lu [14].

We can thus conclude that the product of trees approach can be seen as a way to visualise problems on multiple sources of underlying risk.



141.40	141.40	112.24	89.09	70.72
141.40	41.40	41.40	41.40	41.40
112.24	41.40	41.40	41.40	41.40
89.09	41.40	41.40	41.40	41.40
70.72	41.40	41.40	41.40	41.40

112.24	141.40	112.24	89.09	70.72
141.40	41.40	41.40	41.40	41.40
112.24	41.40	12.24	12.24	12.24
89.09	41.40	12.24	12.24	12.24
70.72	41.40	12.24	12.24	12.24

89.09	141.40	112.24	89.09	70.72
141.40	41.40	41.40	41.40	41.40
112.24	41.40	12.24	12.24	12.24
89.09	41.40	12.24	0.00	0.00
70.72	41.40	12.24	0.00	0.00

70.72	141.40	112.24	89.09	70.72
141.40	41.40	41.40	41.40	41.40
112.24	41.40	12.24	12.24	12.24
89.09	41.40	12.24	0.00	0.00
70.72	41.40	12.24	0.00	0.00

125.98	125.98	100.00	79.38
125.98	33.67	32.72	32.72
100.00	32.72	30.82	30.82
79.38	32.72	30.82	30.82

100.00	125.98	100.00	79.38
125.98	31.09	25.98	25.98
100.00	27.57	9.16	8.77
79.38	27.57	8.77	7.97

79.38	125.98	100.00	79.38
125.98	31.09	27.57	27.57
100.00	27.57	8.08	6.60
79.38	27.57	6.60	0.00

112.24	112.24	89.09
112.24	26.26	24.87
89.09	24.98	22.68

89.09	112.24	89.09
112.24	22.12	17.41
89.09	18.16	6.48

100	100.00
100.00	19.84

Figure 4.10: Backward Induction using product of trees approach for 3 dimensions

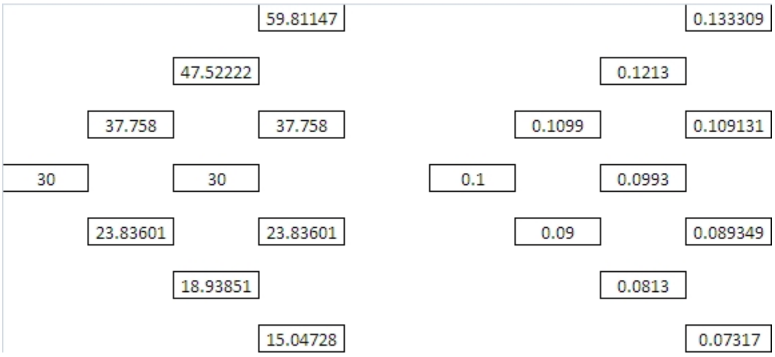
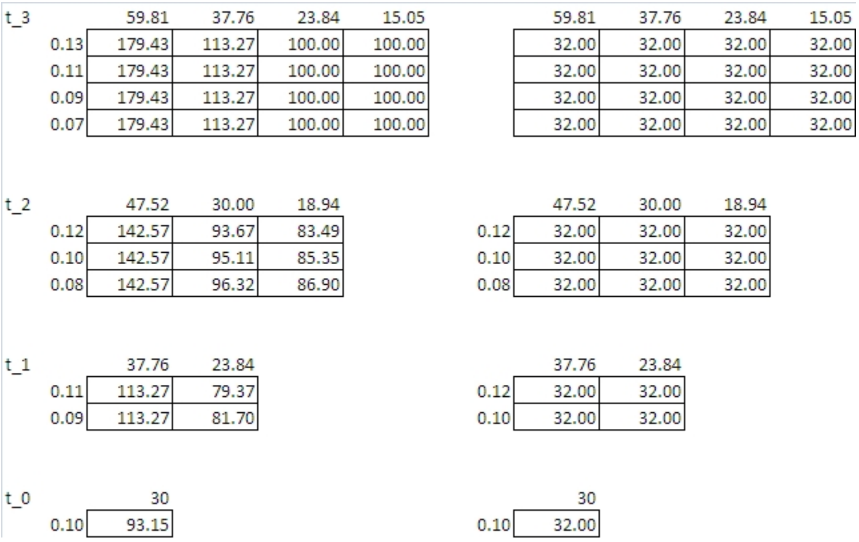


Figure 4.11: Underlying Binomial Trees for 3D Max Convertible Bond



t_3	59.81	37.76	23.84	15.05
0.13	179.43	113.27	100.00	100.00
0.11	179.43	113.27	100.00	100.00
0.09	179.43	113.27	100.00	100.00
0.07	179.43	113.27	100.00	100.00

	59.81	37.76	23.84	15.05
	32.00	32.00	32.00	32.00
	32.00	32.00	32.00	32.00
	32.00	32.00	32.00	32.00
	32.00	32.00	32.00	32.00

t_2	47.52	30.00	18.94
0.12	142.57	93.67	83.49
0.10	142.57	95.11	85.35
0.08	142.57	96.32	86.90

	47.52	30.00	18.94
0.12	32.00	32.00	32.00
0.10	32.00	32.00	32.00
0.08	32.00	32.00	32.00

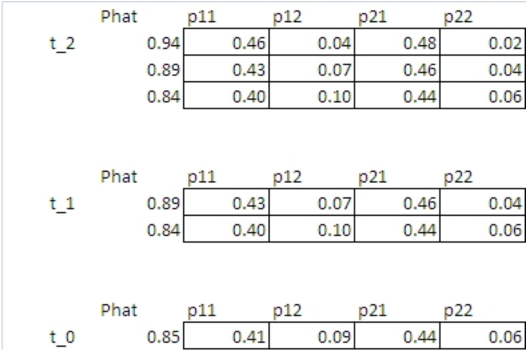
t_1	37.76	23.84
0.11	113.27	79.37
0.09	113.27	81.70

	37.76	23.84
0.12	32.00	32.00
0.10	32.00	32.00

t_0	30
0.10	93.15

	30
0.10	32.00

Figure 4.12: Backward Induction using product of trees approach for Convertible Bond



	Phat	p11	p12	p21	p22
t_2	0.94	0.46	0.04	0.48	0.02
	0.89	0.43	0.07	0.46	0.04
	0.84	0.40	0.10	0.44	0.06

	Phat	p11	p12	p21	p22
t_1	0.89	0.43	0.07	0.46	0.04
	0.84	0.40	0.10	0.44	0.06

	Phat	p11	p12	p21	p22
t_0	0.85	0.41	0.09	0.44	0.06

Figure 4.13: Probabilities for Convertible Bond

Chapter 5

Increasing the Sources of Underlying Risk

Chapter 2 focused on expanding the CRR binomial lattice into the multinomial space, but looking at only one underlying source of risk. Chapter 4 looked at expanding the number of underlying risk sources with the use of the product of trees approach. In this chapter, we will look at increasing the sources of underlying risk using the methods of Boyle. An investigation will also be done on simultaneously increasing the number of nodes in the lattice. It is not difficult to see that there will be a certain level of computational intensity associated with such an experiment. This section will investigate both the convergence and intensity of three models: The Boyle, Evnine and Gibbs (BEG) model [11]; the Kamrad and Ritchken (KR) model [37] and finally the Decoupled approach [40].

5.1 The BEG Model

This section will look at the BEG model [11]. We will see that the BEG model in two dimensions gives results analogous to those derived for the product of tree approach. This is then extended into a lattice model for n sources of underlying risk.

5.1.1 Derivation

We will derive the probabilities for the n-sources of risk model, by starting off with three sources and then using the pattern that emerges to infer the n-source equations. The method used by BEG differs slightly from the approach we used for the product of trees derivation, in that the moment generating function is expanded using a Taylor expansion, and then through matching of coefficients, the probability equations are found. The derivation will be shown here for completeness:

We start by showing the general valuation equation for a security f, which is dependent on n assets S_i, for $1 \leq i \leq n$. We can represent each of the shares S_i by the share price process discussed in Section 2.2.2:

$$dS_i = \mu_i S dt + \sigma_i dW_i. \tag{5.1.1}$$

where W_i is the standard Wiener process. From the Section 2.4 know that the standard representation of a derivative that depends on these n underlying shares can be written as:

$$\frac{\partial f}{\partial t} + \sum_{i=1}^{n} \frac{\partial f}{\partial S_i}(\mu_i - \lambda_i \sigma_i) + 0.5 \sum_{i=1}^{n} \sum_{j=1}^{n} \rho_{ij} \sigma_i \sigma_j S_i S_j \frac{\partial^2 F}{\partial S_i \partial S_j} = rf, \tag{5.1.2}$$

where $\mu_i - \lambda_i \sigma_i = r$

If we look at a derivative with two underlying shares, and again we let the magnitude of the up and down movements be equal to $e^{\sigma_i \sqrt{\Delta t}}$ and $e^{-\sigma_i \sqrt{\Delta t}}$ respectively we can use moment matching to find the probabilities of the joint process.

We define the characteristic function of the discrete distribution as:

$$\phi(x_1, x_2) = E[e^{ix_1 S_1 + ix_2 S_2}]. \tag{5.1.3}$$

Using the probabilities as defined in Section 4.1, this can be expanded to:

$$\begin{aligned}\phi(x_1, x_2) = p_{11} & e^{i\sqrt{\Delta t}(x_1\sigma_1 + x_2\sigma_2)} \\ + p_{12} & e^{i\sqrt{\Delta t}(x_1\sigma_1 - x_2\sigma_2)} \\ + p_{21} & e^{i\sqrt{\Delta t}(-x_1\sigma_1 + x_2\sigma_2)} \\ + p_{22} & e^{i\sqrt{\Delta t}(-x_1\sigma_1 - x_2\sigma_2)}.\end{aligned}$$

This can be expanded using a Taylor series as follows:

$$\begin{aligned}\phi(x_1, x_2) = p_{11}(1 + i\sqrt{\Delta t}(x_1\sigma_1 + x_2\sigma_2) - \frac{\Delta t}{2}(x_1\sigma_1 + x_2\sigma_2))+ \\ p_{12}(1 + i\sqrt{\Delta t}(x_1\sigma_1 - x_2\sigma_2) - \frac{\Delta t}{2}(x_1\sigma_1 - x_2\sigma_2))+ \\ p_{21}(1 + i\sqrt{\Delta t}(-x_1\sigma_1 + x_2\sigma_2) - \frac{\Delta t}{2}(-x_1\sigma_1 + x_2\sigma_2))+ \\ p_{22}(1 + i\sqrt{\Delta t}(-x_1\sigma_1 - x_2\sigma_2) - \frac{\Delta t}{2}(-x_1\sigma_1 - x_2\sigma_2)) + \dots\end{aligned} \tag{5.1.4}$$

We can now equate the discrete distribution to the continuous bivariate normal distribution. The characteristic function for the bivariate normal is given by:

$$\phi_2(x_1,x_2)=1+i\sqrt{\Delta t}(x_1\mu_1+x_2\mu_2)-\frac{\Delta t}{2}(x_1^2\sigma_1^2+x_1x_2 2\sigma_1\sigma_2\rho+x_2^2\sigma_2^2)+... \quad (5.1.5)$$

If we shuffle around the terms from Eq. (5.1.4) in order of match those of Eq. (5.1.5), Eq. (5.1.4) becomes:

$$\begin{aligned}\phi(x_1,x_2)=(p_{11}+p_{12}+p_{21}+p_{22})+&\\ &i\sqrt{\Delta t}[x_1\mu_1(p_{11}+p_{12}-p_{21}-p_{22})+\\ &x_2\mu_2(p_{11}+p_{12}-p_{21}-p_{22})]-\frac{\Delta t}{2}(x_1^2\sigma_1^2+\\ &x_1x_2 2\sigma_1\sigma_2(p_{11}-p_{12}-p_{21}+p_{22})+x_2^2\sigma_2^2)+...\end{aligned}$$

If we match the coefficients, we find the following set of equations:

$$(p_{11}+p_{12}+p_{21}+p_{22})=1$$

$$(p_{11}-p_{12}+p_{21}-p_{22})=\rho$$

$$(p_{11}+p_{12}-p_{21}-p_{22})=\frac{\mu_1\sqrt{\Delta t}}{\sigma_1}$$

$$(p_{11}-p_{12}+p_{21}-p_{22})=\frac{\mu_2\sqrt{\Delta t}}{\sigma_2}.$$

This can be solved to obtain the result:

$$\begin{aligned}p_{11}&=\frac{1+\rho}{4\lambda^2}+0.25(\frac{\mu_1-0.5\sigma_1^2}{\sigma_1}+\frac{\mu_2-0.5\sigma_2^2}{\sigma_2})\frac{\sqrt{\Delta t}}{\lambda}\\ p_{12}&=\frac{1-\rho}{4\lambda^2}+0.25(\frac{\mu_1-0.5\sigma_1^2}{\sigma_1}-\frac{\mu_2-0.5\sigma_2^2}{\sigma_2})\frac{\sqrt{\Delta t}}{\lambda}\\ p_{21}&=\frac{1-\rho}{4\lambda^2}+0.25(\frac{\mu_1-0.5\sigma_1^2}{\sigma_1}+\frac{\mu_2-0.5\sigma_2^2}{\sigma_2})\frac{\sqrt{\Delta t}}{\lambda}\\ p_{22}&=\frac{1+\rho}{4\lambda^2}-0.25(\frac{\mu_1-0.5\sigma_1^2}{\sigma_1}+\frac{\mu_2-0.5\sigma_2^2}{\sigma_2})\frac{\sqrt{\Delta t}}{\lambda}.\end{aligned}$$

From the pattern BEG derives the n underlying assets probability equations as follows:

$$p_m=\frac{1}{2^n}[\sqrt{\Delta t}\sum_{i=1}^{n}x_{im}(\frac{\mu_i}{\sigma_i})+\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}x_{ij}^m(\rho_{ij})] \quad (5.1.6)$$

for $m = 1, 2, ..., 2^n$, $n \geq 2$, and

$$x_{im} = \begin{cases} 1 & \text{if asset } i \text{ has an up movement in state } m \\ -1 & \text{if asset } i \text{ has a down movement in state } m \end{cases}$$

$$x_{ij}^m = \begin{cases} 1 & \text{if assets } i \text{ and } j \text{ move in same direction in state } m \\ -1 & \text{if assets } i \text{ and } j \text{ move in opposite direction in state } m \end{cases}$$

5.2 The Kamrad and Ritchken Extension

Kamrad and Ritchken [37] use the same principles as BEG, but by adding a horizontal jump, they increase the convergence rate. The additional horizontal jump transforms the underlying binomial model into a trinomial model. The additional horizontal jump is obtained by adding a constant in the jump size formula. Thus, $u_i = e^{\sigma_i\sqrt{\Delta t}}$ becomes $u_i = e^{\lambda_i\sigma_i\sqrt{\Delta t}}$. This adds an additional probability to the probability sets derived in the previous section, i.e. the probability of no movement at all. If $\lambda_i = 1$ then the model reduces to the BEG model. We will look at the log return process, and if we define $U^i = lnu_i = \lambda_i\sigma_i\sqrt{\Delta t}$ as the natural logarithms of the respective movements. Table 5.1 depicts the probabilities used.

p_{11}	X moves up and Y moves up
p_{12}	X moves up and Y moves down
p_{21}	X moves down and Y moves up
p_{22}	X moves down and Y moves down
p_0	no movement

Table 5.1: KR Probabilities for two underlying shares

If we follow the same process of matching the expected values, variance and covariances of the underlying assets as in Chapter 3, we come to the following set of equations:

Expected Value:

$$U^X(p_{11} + p_{12} - p_{21} - p_{22}) = \mu_X\Delta t$$
$$U^Y(p_{11} - p_{12} + p_{21} - p_{22}) = \mu_Y\Delta t.$$

Variance:

$$
\begin{aligned}
{U^X}^2(p_{11}+p_{12}+p_{21}+p_{22})-(u_X\Delta t)^2 &= \sigma_X^2\Delta t\\
{U^Y}^2(p_{11}+p_{12}+p_{21}+p_{22})-(u_Y\Delta t)^2 &= \sigma_Y^2\Delta t.
\end{aligned}
$$

Covariance:

$$U^X U^Y(p_{11}-p_{12}-p_{21}+p_{22})-\mu_X\mu_Y\Delta t^2=\sigma_X\sigma_Y\rho\Delta t.$$

If we substitute U^i with $\lambda_i\sigma_i\sqrt{\Delta t}$ we have the following set of equations:

$$
\begin{aligned}
(p_{11}+p_{12}-p_{21}-p_{22}) &= \frac{\mu_X\sqrt{\Delta t}}{\lambda_X\sigma_X^2}\\
(p_{11}-p_{12}+p_{21}-p_{22}) &= \frac{\mu_Y\sqrt{\Delta t}}{\lambda_Y\sigma_Y^2}\\
(p_{11}+p_{12}+p_{21}+p_{22}) &= \frac{1}{\lambda_X^2} \qquad (5.2.1)\\
(p_{11}+p_{12}+p_{21}+p_{22}) &= \frac{1}{\lambda_Y^2} \qquad (5.2.2)\\
(p_{11}-p_{12}-p_{21}+p_{22}) &= \frac{\rho}{\lambda_X\lambda_Y}.
\end{aligned}
$$

These equations are a result of ignoring terms of Δt and higher. Eq. (5.2.1) and Eq. (5.2.2) also results in $\lambda_X=\lambda_Y=\lambda$. We can thus specify the final set of equations to obtain the probabilities as:

$$
\begin{aligned}
p_{11} &= \frac{1+\rho}{4\lambda^2}+0.25(\frac{\mu_X-0.5\sigma_X^2}{\sigma_X}+\frac{\mu_Y-0.5\sigma_Y^2}{\sigma_Y})\frac{\sqrt{\Delta t}}{\lambda}\\
p_{12} &= \frac{1-\rho}{4\lambda^2}+0.25(\frac{\mu_X-0.5\sigma_X^2}{\sigma_X}-\frac{\mu_Y-0.5\sigma_Y^2}{\sigma_Y})\frac{\sqrt{\Delta t}}{\lambda}\\
p_{21} &= \frac{1-\rho}{4\lambda^2}+0.25(\frac{\mu_X-0.5\sigma_X^2}{\sigma_X}+\frac{\mu_Y-0.5\sigma_Y^2}{\sigma_Y})\frac{\sqrt{\Delta t}}{\lambda}\\
p_{22} &= \frac{1+\rho}{4\lambda^2}-0.25(\frac{\mu_X-0.5\sigma_X^2}{\sigma_X}+\frac{\mu_Y-0.5\sigma_Y^2}{\sigma_Y})\frac{\sqrt{\Delta t}}{\lambda}\\
p_0 &= 1-\frac{1}{\lambda^2}. \qquad (5.2.3)
\end{aligned}
$$

The n source set of probability equations is thus derived as:

$$p_m=\frac{1}{2^n}[\frac{1}{\lambda^2}+\frac{\sqrt{\Delta t}}{\lambda}\sum_{i=1}^{n}x_{im}(\frac{\mu_i}{\sigma_i})+\frac{1}{\lambda^2}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}x_{ij}^m(\rho_{ij})] \qquad (5.2.4)$$

for $m = 1, 2, ..., 2^n$, $n \geq 2$, $\lambda \geq 1$ and

$$x_{im} = \begin{cases} 1 & \text{if asset } i \text{ has an up movement in state } m \\ -1 & \text{if asset } i \text{ has a down movement in state } m \end{cases}$$

$$x_{ij}^{m} = \begin{cases} 1 & \text{if assets } i \text{ and } j \text{ move in same direction in state } m \\ -1 & \text{if assets } i \text{ and } j \text{ move in opposite direction in state } m \end{cases}$$

An important aspect of the KR model is the value of the λ parameter. This additional degree of freedom needs to be specified, and may vary from problem to problem. Fig. 5.1 depicts the sensitivity when computing the price of a max option to the choice of λ. It is clear that for any λ above 1.5, the value of the option differs significantly form the analytical value. This is a definite drawback to the KR approach.

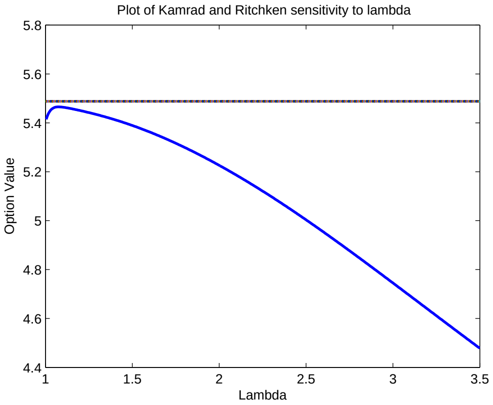


Figure 5.1: Kamrad and Ritchken approach sensitivity to lambda

The next section deals in more detail with the choice of λ.

5.2.1 The Modified Kamrad and Ritchken Extension

In the Kamrad and Ritchken model, the terms of order Δt and higher are omitted form Eq. (5.1.5). Herath and Kumar [22] investigate the implications of this omission. Kamrad and Ritchken also mention that their model yields feasible probabilities for any value of λ where $\lambda > 1$, this is proved to be false by Herath and Kumar, who provide examples where negative probabilities are obtained. By developing bounds for the stretch parameter, as well as bounding the correlation coefficient, they are able to provide intervals where the probabilities produced will always be feasible.

It is relatively simple to derive the level of error introduced into the Kamrad and Ritchken model, by leaving the higher order Δt terms and comparing the probabilities to the probabilities derived when the terms are dropped. By ignoring this term for the two sources of underlying risk problem, only one λ is required ($\lambda_X = \lambda_Y$). If the Δt terms were not ignored, following equations would replace Eq. (5.2.1) and Eq. (5.2.2):

$$(p_{11} + p_{12} + p_{21} + p_{22}) = \frac{1 + \gamma_X^2 \Delta t}{\lambda_X^2} \tag{5.2.5}$$

$$(p_{11} + p_{12} + p_{21} + p_{22}) = \frac{1 + \gamma_Y^2 \Delta t}{\lambda_Y^2}, \tag{5.2.6}$$

where we define $\gamma_i = \frac{u_i}{\sigma_i}$.

We thus have that:

$$\frac{\lambda_X}{1 + \gamma_X^2 \Delta t} = \frac{\lambda_Y}{1 + \gamma_Y^2 \Delta t}.$$

Thus $\lambda_X = \lambda_Y$ if $\gamma_X = \gamma_Y$. It is clear that the error introduced into the KR model will be small if γ_X and γ_Y are close.

Furthermore, Herath and Kumar propose that both λ and the correlation between the assets should be bounded to ensure positive probabilities. These bounds are as follows for the two dimensional case:

$$\lambda_X \geq \sqrt{1 + \gamma_X^2 \Delta t}$$

$$\lambda_Y = \frac{\sqrt{1 + \gamma_Y^2 \Delta t}}{\sqrt{1 + \gamma_X^2 \Delta t}},$$

and the correlation should be bounded as follows:

$$max[L_1, L_2] \leq \rho \leq min[U_1, U_2],$$

where

$$L_1 = [-\lambda_X(\tilde{\lambda}\gamma_X + \gamma_Y) - \tilde{\lambda}(1 + \gamma_X \Delta t) - \gamma_X \gamma_Y \Delta t]$$
$$L_2 = [\lambda_X(-\tilde{\lambda}\gamma_X + \gamma_Y) + \tilde{\lambda}(1 + \gamma_X \Delta t) - \gamma_X \gamma_Y \Delta t]$$
$$U_1 = [\lambda_X(\tilde{\lambda}\gamma_X - \gamma_Y) + \tilde{\lambda}(1 + \gamma_X \Delta t) - \gamma_X \gamma_Y \Delta t]$$
$$U_2 = [\lambda_X(\tilde{\lambda}\gamma_X + \gamma_Y) - \tilde{\lambda}(1 + \gamma_X \Delta t) - \gamma_X \gamma_Y \Delta t],$$

and where $\tilde{\lambda} = \frac{\lambda_X}{\lambda_Y}$

We can finally conclude that the Herath and Kumar approach will provide results that are analogous to Kamrad and Ritchken when the time step is very small. The work done on providing bounds for the stretch parameter λ will be important when dealing with instances where negative probabilities occur.

5.3 The Decoupling Approach

Korn and Muller [40] proposed an alternative to the methods used by BEG and Kamrad and Ritchken. Korn and Muller note a number of drawbacks to the BEG and Kamrad and Ritchken models, these include the difficulty in constructing the m-dimensional Markov chain to model the m-dimensional share price process and the possibility of negative probabilities due to the correlation parameter being present in the transition probabilities. The decoupling approach hopes to provide a viable alternative to the moment matching approach by:

1. Separating the correlation from the tree, and thus ensuring positive probabilities.
2. Simplifying the construction of the tree.
3. Improving numerical performance.

This section will investigate the claims made by Korn and Muller. The decoupling method will be derived, and the spectral and Cholesky decomposition methods will be studied.

Suppose that once again we have two correlated assets, namely Y_1 and Y_2. If we look at the assets' log return process i.e. at X_1 and X_2, we find:

$$dX_i = \mu_i dt + \sigma_i dW^i(t) \tag{5.3.1}$$

for $i = 1, 2$, and where W^i are two correlated Brownian motions and $\mu_i = r - 0.5\sigma_i$.

The idea behind decoupling would be two transform the two correlated processes into independent processes, by decoupling the Variance-Covariance matrix. For the two asset case presented above, Hull and White [28] suggest the following decoupling procedure:

$$\mathbf{L} = \begin{pmatrix} \sigma_2 & \sigma_1 \\ \sigma_2 & -\sigma_1 \end{pmatrix} \mathbf{X}$$

In the next section we will look at the general decoupling method for m sources of underlying risk as described by Korn and Muller.

5.3.1 General Decoupling Approach

Suppose we have m correlated assets Y_i. If we look at the assets' log return process:

$$dX_i = \mu_i dt + \sigma_i dW^i(t) \tag{5.3.2}$$

for $i = 1, ..., m$ and where $\mu_i = r - 0.5\sigma_i$.

We can write the Variance Covariance matrix as:

$$\mathbf{\Sigma} = \begin{pmatrix} \sigma_1^2 & \dots & \rho_{1m}\sigma_1\sigma m \\ \vdots & \ddots & \vdots \\ \rho_{1m}\sigma_1\sigma m & \dots & \sigma_m^2 \end{pmatrix} = GDG^T. \tag{5.3.3}$$

Where G and D are $m \times m$ and D diagonal. Since we assume that Σ is positive-definite, both G and D are invertible. We can now write the decomposition elementwise as:

$$\rho_{ij}\sigma_i\sigma j = \sum_{k=1}^{m}(g_{ij}d_{kk}g_{jk}) \tag{5.3.4}$$

for $i, j = 1, ...m$

Korn and Muller proceed to show that the transformation $Z = G^{-1}X$ results in m independent Brownian motions. The process followed by Z can be given by:

$$dZ_i(t) = \alpha_i dt + \sqrt{d_{ii}}dW^i(t), \tag{5.3.5}$$

where $Z(0) = G^{-1}X(0)$ and i goes from 1 to m, and $\underline{\alpha} = G^{-1}\mu_i = G^{-1}(\underline{r} - 0.5\underline{\sigma}^2)$

The next step is to decide on the decomposition approach. We will discuss two different types: Spectral and Cholesky.

5.3.2 Decoupling with Spectral Decomposition

The Spectral theorem states that there is an orthogonal matrix G such that when

$$\mathbf{\Sigma} = GDG^T, \tag{5.3.6}$$

D is Diagonal and each element is an eigenvalue of Σ [40]. This means that we can replace the d_{ii} diagonal elements in Eq. (5.3.5) with the eigenvalues.

5.3.3 Decoupling with Cholesky Decomposition

With Cholesky decomposition we can decompose the Variance-Covariance matrix as follows:

$$\mathbf{\Sigma} = GG^T, \tag{5.3.7}$$

where G is a lower triangular matrix. When using the Cholesky decomposition, all the diffusion elements (d_{ii}) are equal to 1.

5.3.4 Discretisation Process

In order to use the decoupling procedure, we need to discretise the process of $Z = G^{-1}X$. Korn and Muller propose the following setup:

$$\mathbf{Z_k^N} = \begin{pmatrix} Z^N_{k-1,1} + \alpha_1 \Delta t + q_{k,1}\sqrt{d_{11}}\sqrt{\Delta t} \\ \vdots \\ Z^N_{k-1,m} + \alpha_m \Delta t + q_{k,m}\sqrt{d_{mm}}\sqrt{\Delta t} \end{pmatrix}, \tag{5.3.8}$$

where $Z_(0)^N = Z_0 = G^{-1}X_0$ and

$$q_{k,i} = \begin{cases} 1 \text{ With probability } 0.5 \\ -1 \text{ With probability } 0.5 \end{cases}$$

The one-step transition then has a probability of 0.5^m.

In order to price a derivative, where the payments are traditionally given in terms of the asset price, we need to transform the discrete process back so that it is in terms of the initial asset. This transformation is given by:

$$Y_N^N = (e^{G_1 \underline{Y}}, ..., e^{G_m \underline{Y}})^T. \tag{5.3.9}$$

The following Algorithm proposed by Korn and Muller is used to price a derivative using the decoupling method introduced in this section:

Algorithm 1:

1. Decompose the Variance Covariance matrix into GDG^T using either spectral or Cholesky decomposition.
2. Transform the underlying asset X into the process Z such that the new process has independent Brownian motions.
3. Discretise the process using Eq. (5.3.8).
4. Transform the discrete process using Eq. (5.3.9) so that it is in terms of the underlying asset once more.
5. Calculate the payoff across all nodes using the transformation, and work back through the tree using backward induction.

5.4 Numerical Results

This sections compares the convergence results of the BEG, Kamrad and Ritchken and Decoupled approaches. These approaches are used to value three different derivatives:

1. European Max Option
2. European Product Option
3. European Digital Option.

Both two dimensional and three dimensional derivatives are valued. The main point of discussion will be the convergence of the different models to their analytical values. Computational intensity is also discussed.

Before looking at the convergence results, we will investigate the computational intensity. Both Kamrad and Ritchken and Herath and Kumar consider the number of nodes generated by each model, as well as the number of additions and multiplications required for each node. We will consider the number of nodes required for one asset, two asset, three asset and n asset models, comparing the BEG model, the Kamrad and Ritchken model and the decoupling approach. Table 5.2 shows the number of nodes required for each option:

Number of Assets	# Nodes: BEG	# Nodes: Kamrad and Ritchken	# Nodes: Decoupled Approach
1	$\sum_{i=0}^{j}(i+1)$	$\sum_{i=0}^{j}(2i+1)$	$\sum_{i=1}^{j} i$
2	$\sum_{i=0}^{j}(i+1)^2$	$\sum_{i=0}^{j}[(i+1)^2+i^2]$	$\sum_{i=1}^{j} i^2$
3	$\sum_{i=0}^{j}(i+1)^3$	$\sum_{i=0}^{j}[(i+1)^3+i^3]$	$\sum_{i=1}^{j} i^3$
n	$\sum_{i=0}^{j}(i+1)^n$	$\sum_{p=0}^{j}\sum_{i=0}^{p}[(i+1)^n]$	$\sum_{i=1}^{j} i^n$

Table 5.2: Number of nodes generated after j iterations

If we look at a simple example where we have 100 iterations, the following table shows the number of nodes required for each model:

Number of Assets	# Nodes: BEG	# Nodes: Kamrad and Ritchken	# Nodes: Decoupled Approach
1	5151	10201	5050
2	348,551	686,901	338,350
3	26,532,801	55,195,1971	25,502,500

Table 5.3: Number of nodes generated after 100 iterations

The following three graphs show the nature of the increase in computational intensity of the three models.

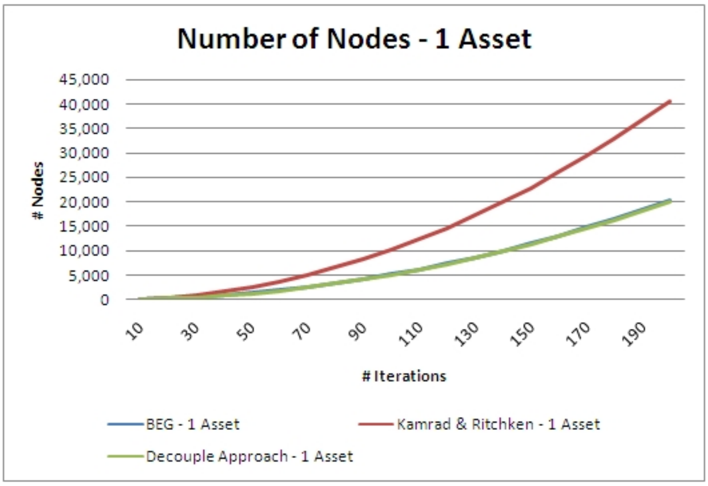


Figure 5.2: Number of Nodes Required for 1 Share

Fig. 5.2 shows that for one underlying share, the BEG and Decoupled approaches require a similar number of computations, whereas the KR approach is more intensive. After around 200 timesteps, the KR approach requires double the amount of computations as the others.

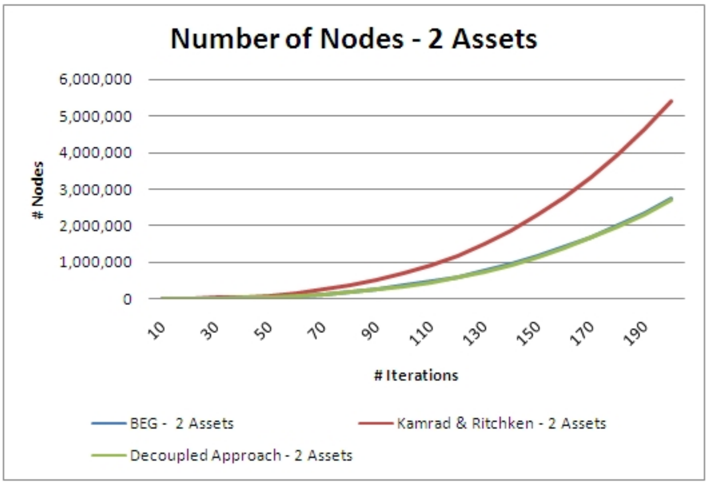


Figure 5.3: Number of Nodes Required for 2 Shares

Fig. 5.3 paints a similar picture, but it is clear that the KR approach is showing some computational strain when compared to the other two. After 200 timesteps, the KR approach requires double the amount of computations as the others.

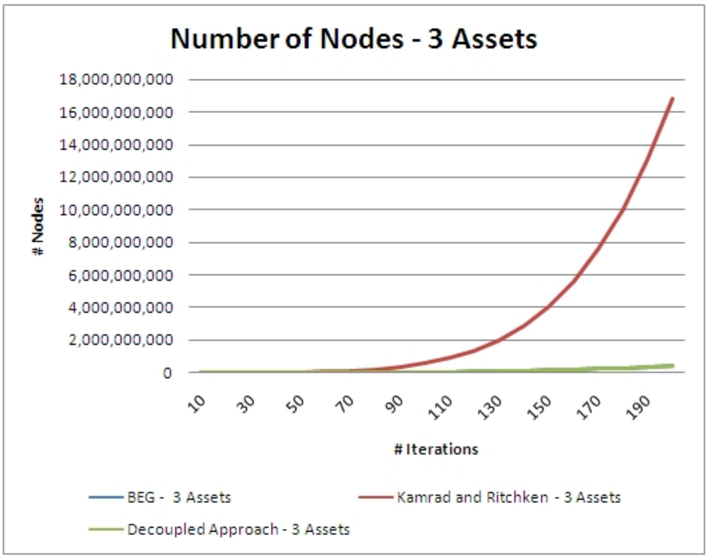


Figure 5.4: Number of Nodes Required for 3 Shares

When a third share is introduced, the KR approach explodes as can be seen in Fig. 5.4. It is not completely clear from the graph, but looking at Table 5.3, the Decoupled approach starts outperforming the BEG approach significantly when the number of timesteps increases beyond 100.

The graphs and the table clearly show that in terms of computational intensity, the decoupled approach is only slightly less intensive than the BEG model when there are one or two underlying shares, but once the option consists of three shares, there is a drastic difference. The Kamrad and Ritchken approach "blows up" completely for more than two shares.

We know now that the Kamrad and Ritchken model is considerably more computationally intensive than the BEG or the Decoupled approach. It thus makes sense that it needs to converge to the analytical solution much faster than the other two. We will now investigate the order of convergence of the three models. We know that traditionally binomial models converge at a rate of $O(\frac{1}{n})$ from [23]. We also proved the rates for the multinomial case in Section 2.3, and saw that the actual rate of convergence is closer to $O(\frac{1}{\sqrt{n}})$, and it was thus clear that the actual convergence did not really live up to the theoretical convergence. As in Section 2.3, we investigate the error ratio as we double the number of iterations. If the convergence is of order $O(\frac{1}{n})$ we expect to see error ratios of 2.

The following three options are valued using the approaches outlined above. The Matlab code used in each case can be found in Appendix C.2

1. European Max Option:

   These problems come from the papers by Boyle [9] and Boyle et al [11]. This assists with comparison purposes.

S_1	40
S_2	40
X	40
σ_1	0.2
σ_2	0.3
ρ	0.5
r	0.04879
T	0.58333
Analytical Price	5.488

Table 5.4: Two Dimensional Max Option Specifications

S_1	100
S_2	100
S_3	100
X	100
σ_1	0.2
σ_2	0.2
σ_3	0.2
ρ_{12}	0.5
ρ_{13}	0.5
ρ_{23}	0.5
r	0.1
T	1
Analytical Price	22.672

Table 5.5: Three Dimensional Max Option Specifications

2. European Product Option:

   These problems come from the paper by Korn and Muller [40]. This assists with comparison purposes.

S_1	20
S_2	22
X	10
σ_1	0.2
σ_2	0.25
ρ	0.5
r	0.1
T	1
Analytical Price	3.26214

Table 5.6: Two Dimensional Product Option Specifications

S_1	20
S_2	20
X	10
σ_1	0.2
σ_2	0.25
ρ	0.5
r	0.1
T	1
Analytical Price	10.8208

Table 5.7: Two Dimensional ITM Product Option Specifications

S_1	20
S_2	20
X	20
σ_1	0.2
σ_2	0.25
ρ	0.5
r	0.1
T	1
Analytical Price	2.5294

Table 5.8: Two Dimensional ATM Product Option Specifications

3. European Digital Option:

   These problems come from the paper by Korn and Muller [40]. This assists with comparison purposes.

S_1	20
S_2	20
X	30
σ_1	0.2
σ_2	0.25
ρ	0.5
r	0.1
T	1
Analytical Price	0.1042

Table 5.9: Two Dimensional OTM Product Option Specifications

S_1	22
S_2	20
S_3	25
X	20
σ_1	0.2
σ_2	0.25
σ_3	0.15
ρ_{12}	0.5
ρ_{13}	-0.2
ρ_{23}	-0.4
r	0.1
T	1
Analytical Price	3.90427

Table 5.10: Three Dimensional Product Option Specifications

5.4.1 Convergence Results

Comparing Tables 5.12 and 5.13 one can see that the computational intensity of the KR approach results in it taking much longer than the BEG approach. The error ratio does seem to be around the 2 mark, indicating that both these approaches have a order of convergence of approximately $O(1/N)$. From Fig. 5.5 and Fig. 5.6 the convergence of both these approaches to the analytical results is clear.

Both Fig. 5.7 and Fig. 5.8 show that the BEG and KR models seem to have similar convergence patterns.

S_1	12
S_2	12
X_1	17
X_2	20
σ_1	0.2
σ_2	0.25
Cash	100
ρ	0.5
r	0.1
T	1
Analytical Price	1.35

Table 5.11: Two Dimensional Digital Option Specifications

N	BEG	Absolute Error	Error Ratio	Time(seconds)
5	5.623	0.135		0.014
10	5.401	0.087	1.557	0.006
20	5.444	0.044	1.969	0.041
40	5.470	0.018	2.471	0.316
80	5.479	0.009	1.979	2.356
160	5.482	0.006	1.486	18.400
250	5.484	0.004	1.638	78.456

Table 5.12: Two Dimensional Max Option Convergence - BEG

N	KR	Absolute Error	Error Ratio	Time(seconds)
5	5.444	0.044		0.970
10	5.462	0.026	1.721	0.101
20	5.475	0.013	1.961	0.569
40	5.483	0.005	2.423	4.701
80	5.485	0.003	1.937	42.980
160	5.486	0.002	1.458	320.521

Table 5.13: Two Dimensional Max Option Convergence - KR

If we compare Tables 5.15, 5.15 and 5.16 one can see that when the number of time steps reaches more than 160, the Decoupled approach is less computationally intensive, which proves our postulation in Table 5.3.

The error ratio also seems to be around the 2 mark, indicating all these approaches have a order of convergence of approximately $O(1/N)$ for the Product option. From Fig. 5.9, Fig. 5.10 and Fig. 5.11 we can see the convergence.

The impact of the strike level of the option is investigated for the Decou-

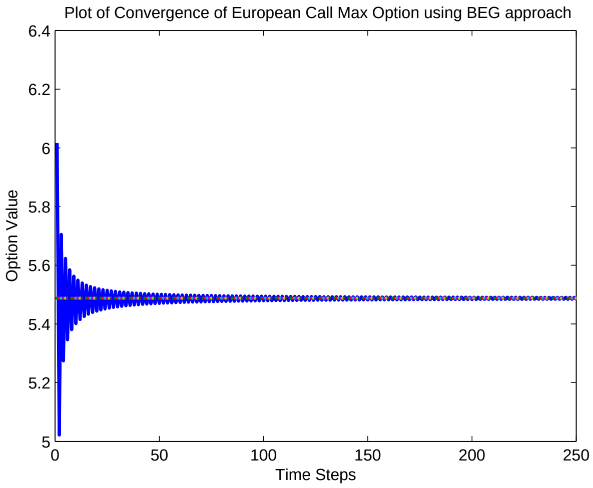


Figure 5.5: 2D Max Option - BEG

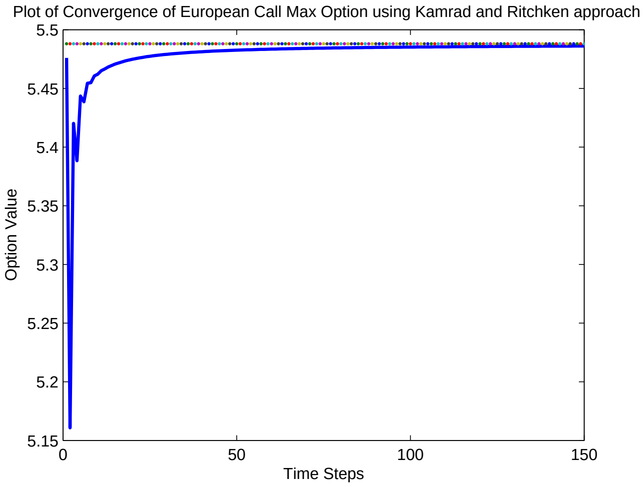


Figure 5.6: 2D Max Option - Kamrad and Ritchken

pled approach in Fig. 5.12, Fig. 5.13 and Fig. 5.14. The strike level does not seem to impact the convergence of the Decoupled approach. Looking at the error ratios, all three methods provide a rate of convergence that is quite similar.

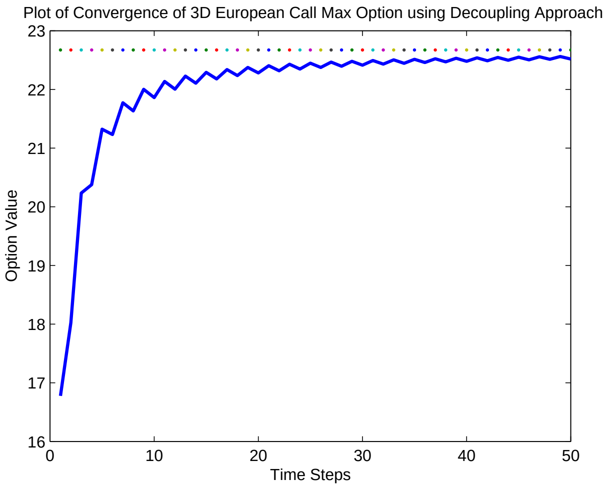


Figure 5.7: 3D Max Option - BEG

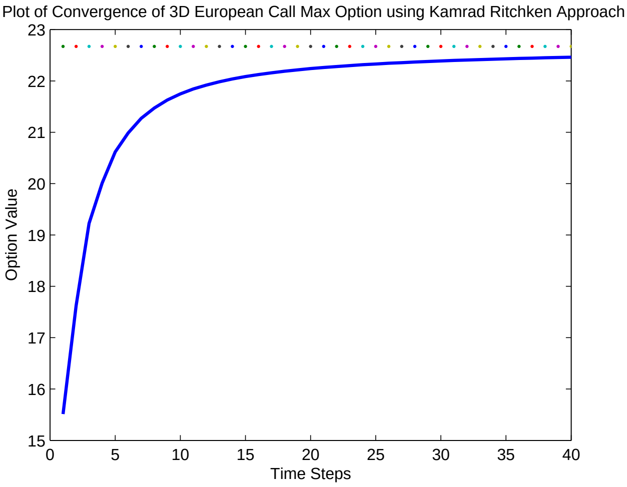


Figure 5.8: 3D Max Option - Kamrad and Ritchken

N	BEG	Absolute Error	Error Ratio	Time(seconds)
5	3.265	0.003		0.004
10	3.261	0.001	3.768	0.006
20	3.260	0.003	0.275	0.037
40	3.261	0.001	2.586	0.616
80	3.262	0.000	2.407	4.774
160	3.262	0.000	1.633	16.728
250	3.262	0.000	1.573	69.976

Table 5.14: Two Dimensional Product Option Convergence - BEG

N	KR	Absolute Error	Error Ratio	Time(seconds)
5	3.262	0.000		0.016
10	3.262	0.000	2.555	0.089
20	3.259	0.003	0.047	0.723
40	3.261	0.002	2.035	4.597
80	3.261	0.001	2.397	39.444
160	3.262	0.000	1.963	261.982

Table 5.15: Two Dimensional Prod Option Convergence - KR

N	Decoupled	Absolute Error	Error Ratio	Time(seconds)
5	3.260	0.002		0.008
10	3.267	0.005	0.319	0.010
20	3.263	0.001	7.379	0.052
40	3.262	0.000	2.676	0.628
80	3.263	0.000	0.640	4.431
160	3.262	0.000	1.747	17.824
250	3.262	0.000	2.623	66.595

Table 5.16: Two Dimensional Prod Option Convergence - Decoupled

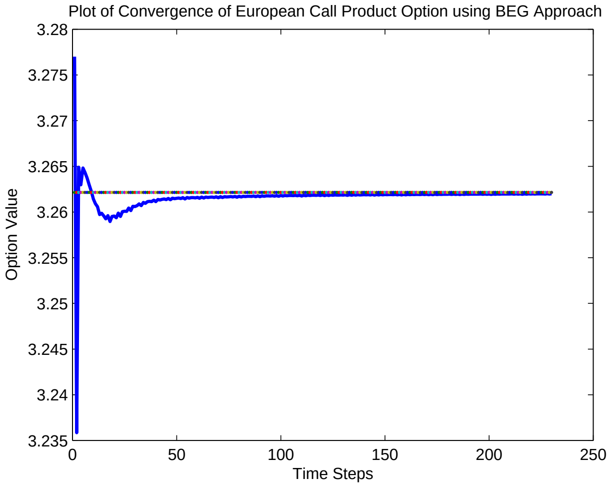


Figure 5.9: 2D Product Option - BEG

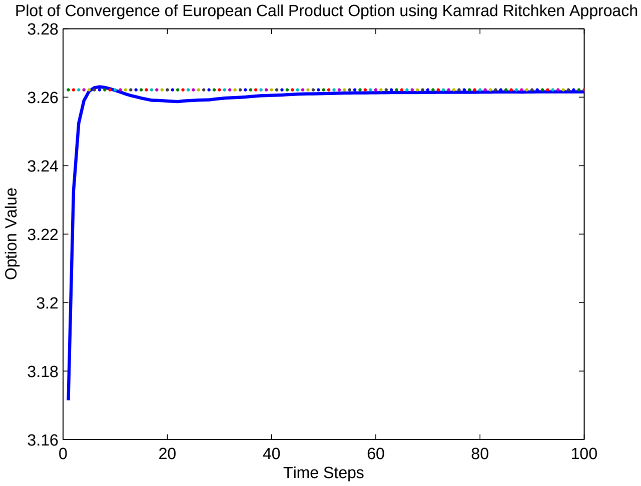


Figure 5.10: 2D Product Option - KR

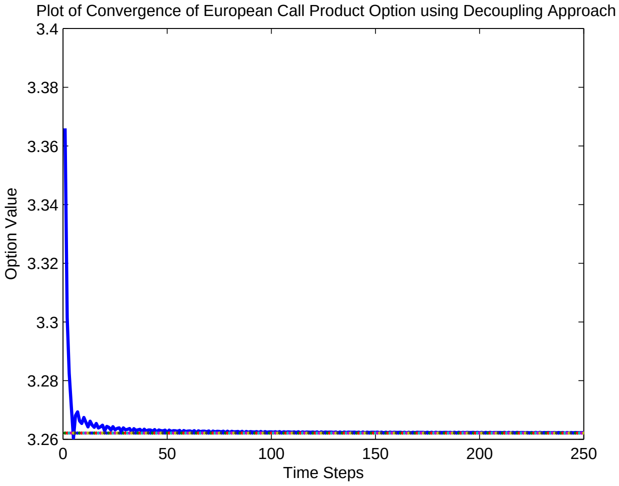


Figure 5.11: 2D Product Option - Decoupled

N	BEG	Absolute Error	Error Ratio	Time(seconds)
5	1.678	0.338		0.004
10	1.078	0.262	1.294	0.008
20	1.743	0.403	0.650	0.059
40	1.348	0.008	52.434	0.396
80	1.323	0.017	0.463	3.355
160	1.395	0.055	0.301	25.622
250	1.493	0.153	0.360	191.237

Table 5.17: Two Dimensional Digital Option Convergence - BEG

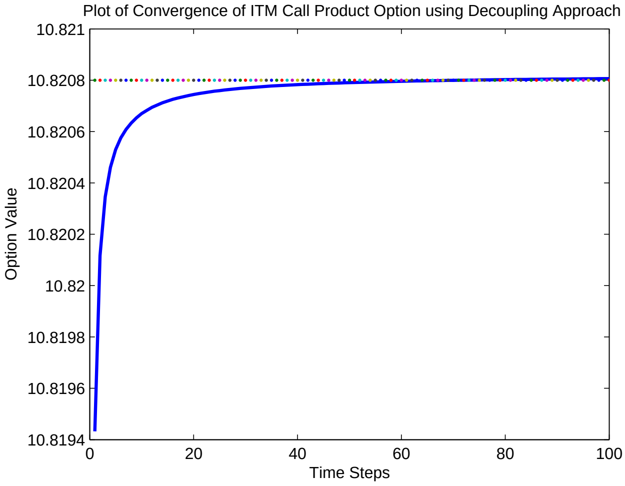


Figure 5.12: 2D ITM Product Option - Decoupled

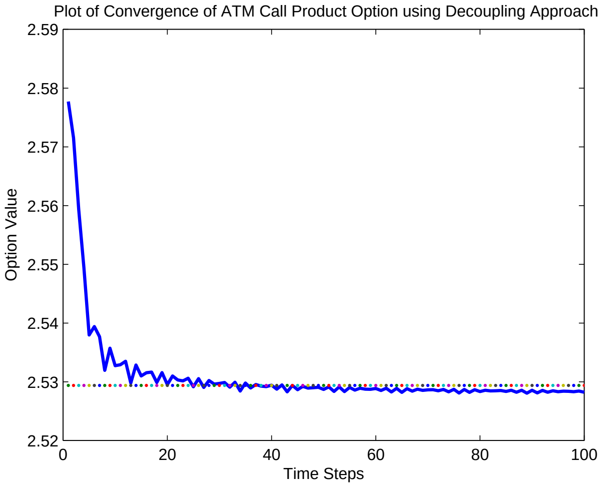


Figure 5.13: 2D ATM Product Option - Decoupled

N	KR	Absolute Error	Error Ratio	Time(seconds)
5	0.607	0.733		0.011
10	1.674	0.334	2.198	0.0812
20	1.318	0.022	15.107	0.896
40	1.469	0.129	0.171	6.563
80	1.356	0.016	8.091	70.234
160	1.293	0.047	0.338	639.868

Table 5.18: Two Dimensional Digital Option Convergence - KR

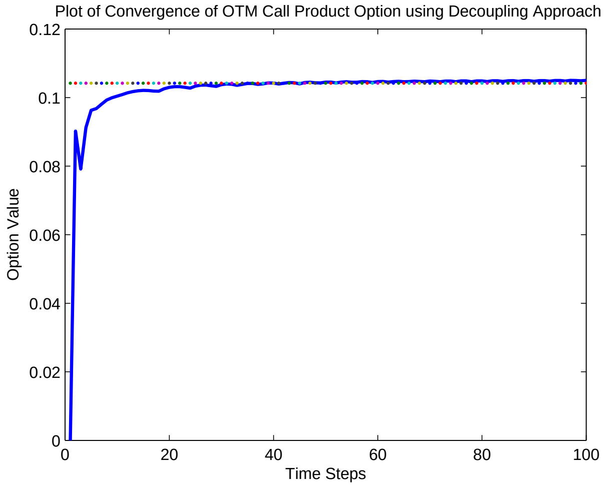


Figure 5.14: 2D OTM Product Option - Decoupled

Comparing Tables 5.17 and 5.18 one can see that the computational intensity of the KR approach results in it taking much longer than the BEG approach. The error ratio seems to be very volatile, and no clear pattern emerges. From Fig. 5.15 and Fig. 5.16 the convergence of both these approaches to the analytical results is clear.

It is interesting to note that the convergence pattern for the Digital option differs significantly form the previous options. This is due to the non-linearity of the payoff function.

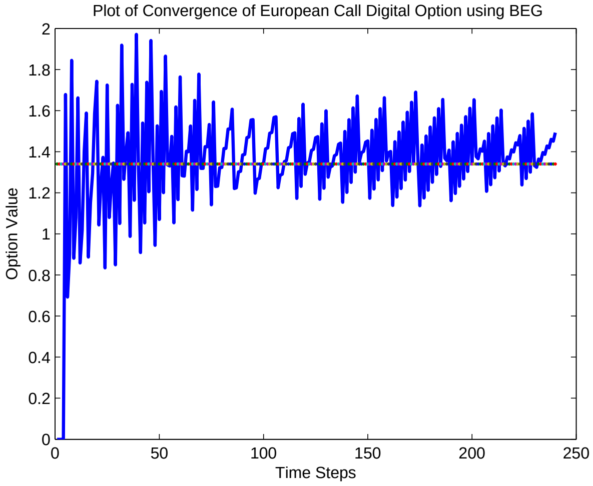


Figure 5.15: 2D Digital Option - BEG

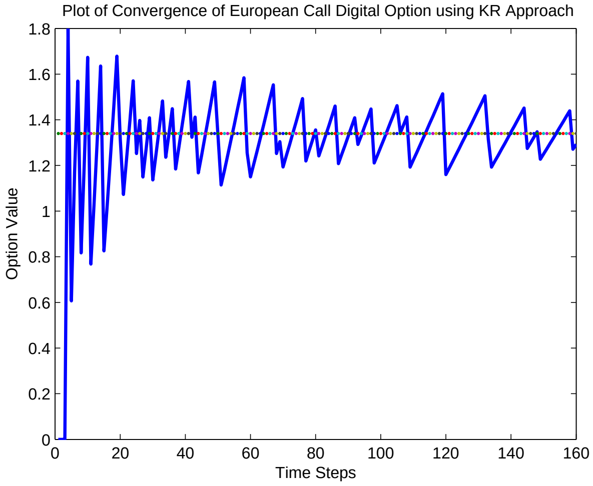


Figure 5.16: 2D Digital Option - KR

5.4.2 Sensitivity Analysis

This next section investigates the sensitivity of options with two underlying assets to changes in their parameters. Correlation, Strike Values and Volatility sensitivity are tested. For each instance a European Product Call Option is used, with specifications as in Table 5.6.

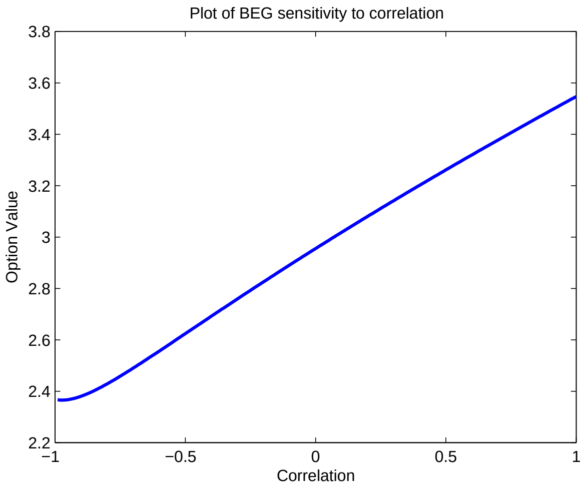


Figure 5.17: BEG - Sensitivity to Correlation

Fig. 5.17 depicts the sensitivity of the option value to changes in the correlation between the two assets. If the correlation between the assets increases, the value of the option also increases. The option value does seem to be relatively sensitive to movements in the correlation as the range of the option value is from 2.4 to 3.6.

Fig. 5.18 depicts the sensitivity of the option value to changes in the strike of the option. The greater the strike, the lower the value of the option. This makes sense as it is a call option that is being valued. The sensitivity is quite pronounced when the strike moves lower into the in the money range.

Fig. 5.19 depicts the sensitivity of the option value to changes in the volatility of each of the shares. The Vega surface can be seen. The sensitivity is not as pronounced as with the strike, but greater than with the correlation. The greater the volatility of both shares, the greater the value of the option.

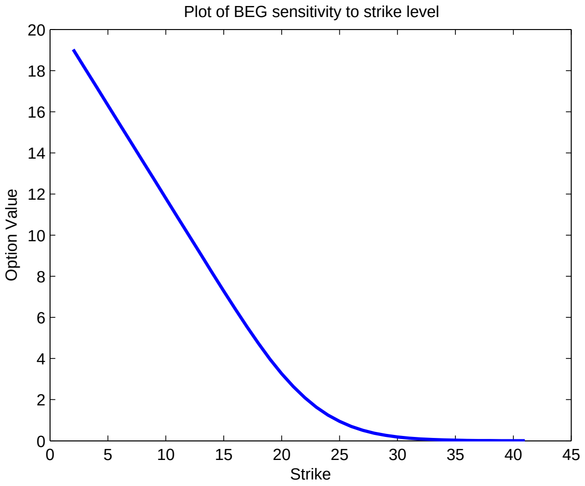


Figure 5.18: BEG - Sensitivity to Strike

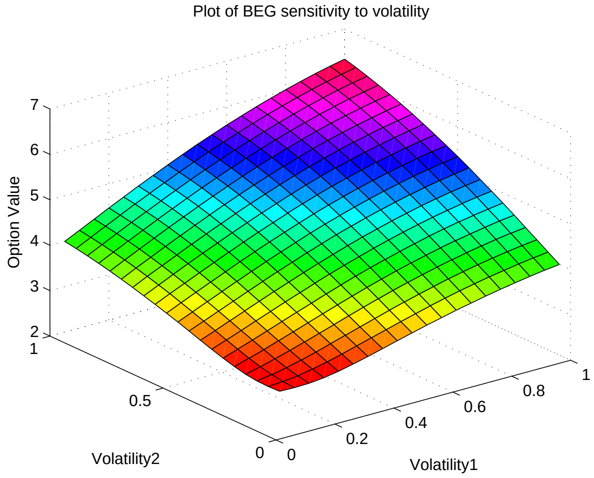


Figure 5.19: BEG - Sensitivity to Volatility

5.4.3 Decoupling Approach in higher dimensions

The previous section investigated the Decoupling approach up to two dimensions. Korn and Muller mention that their approach will continue to be an improvement over alternate methods as the dimension of the problem increases. This section will look at the efficiency of the Decoupling approach for up to five dimensions. Convergence and efficiency will be examined and compared against a Monte Carlo approach.

Product options with the specifications as in Table 5.19 are valued using the approach. The figures clearly show that the decoupled approach is much less intensive than the Monte Carlo. The convergence to the exact price takes less and less steps as the dimension increases.

Parameter	2D	3D	4D	5D
X	20	20	20	20
T	1	1	1	1
S_1	22	22	22	100
S_2	20	20	20	20
S_3		25	25	25
S_4			25	25
S_5				22
σ_1	0.2	0.2	0.2	0.2
σ_2	0.25	0.25	0.25	0.25
σ_3		0.15	0.15	0.15
σ_4			0.15	0.15
σ_5				0.1
ρ_{12}	0.5	0.5	0.2	0.5
ρ_{13}		-0.2	-0.2	-0.2
ρ_{23}		-0.4	-0.4	-0.4
ρ_{14}			0.4	0.4
ρ_{24}			0.3	0.3
ρ_{34}			0.3	0.3
ρ_{15}				0.4
ρ_{25}				-0.3
ρ_{35}				-0.3
ρ_{45}				0.1
r	0.1	0.1	0.1	0.1

Table 5.19: Higher Dimensional Product Option Specifications

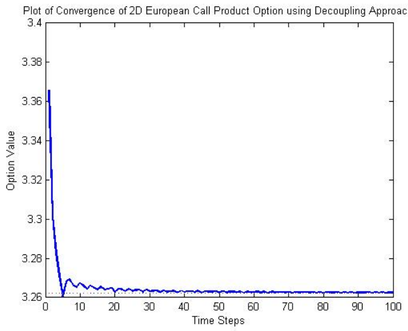


Figure 5.20: Convergence of 2D Decoupled - In this case time steps refer to number of discretisations

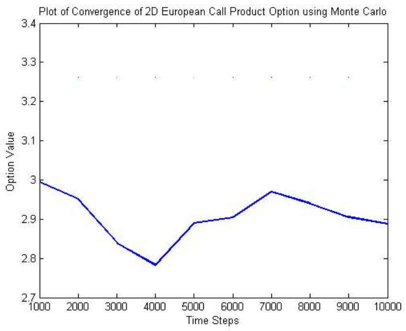


Figure 5.21: Convergence of 2D Monte Carlo - In this case time steps refer to number of iterations

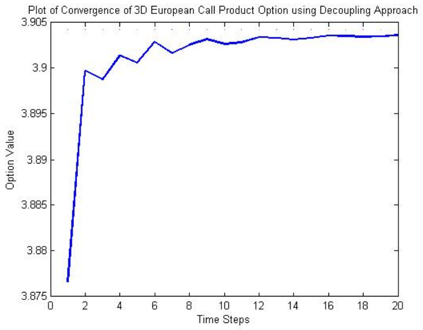


Figure 5.22: Convergence of 3D Decoupled - In this case time steps refer to number of discretisations

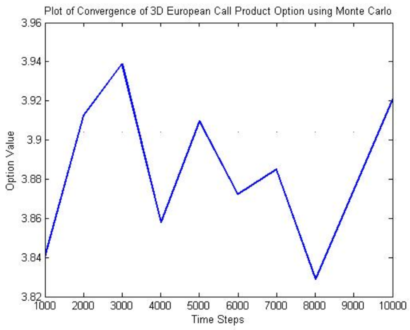


Figure 5.23: Convergence of 3D Monte Carlo - In this case time steps refer to number of iterations

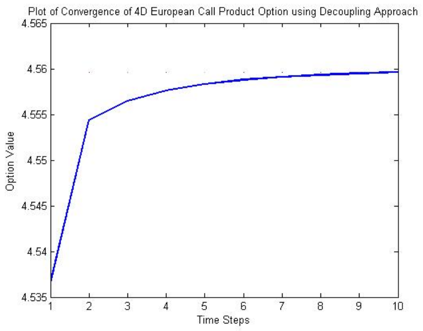


Figure 5.24: Convergence of 4D Decoupled - In this case time steps refer to number of discretisations

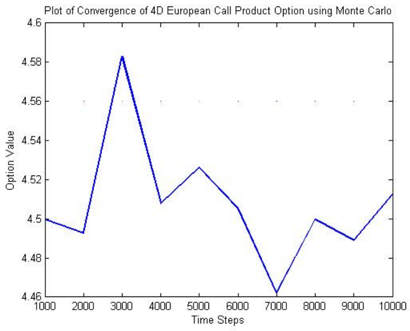


Figure 5.25: Convergence of 4D Monte Carlo - In this case time steps refer to number of iterations

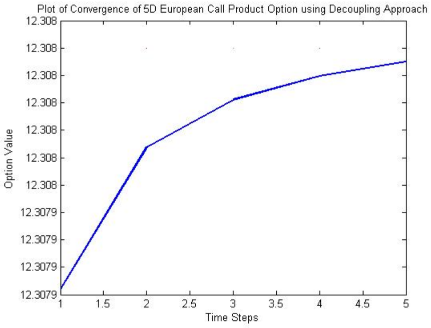


Figure 5.26: Convergence of 5D Decoupled - In this case time steps refer to number of discretisations

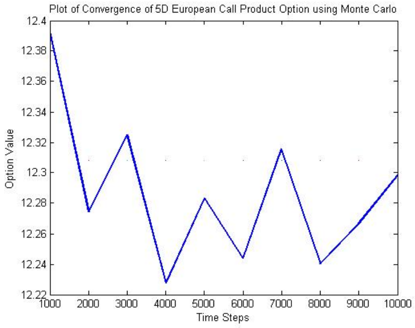


Figure 5.27: Convergence of 5D Monte Carlo - In this case time steps refer to number of iterations

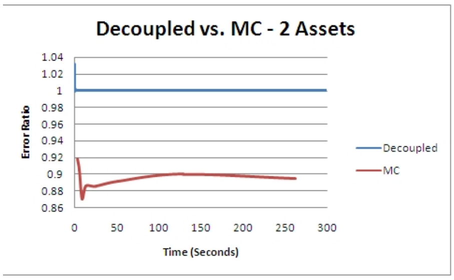


Figure 5.28: Efficiency of Decoupling vs MC in 2 Dimensions

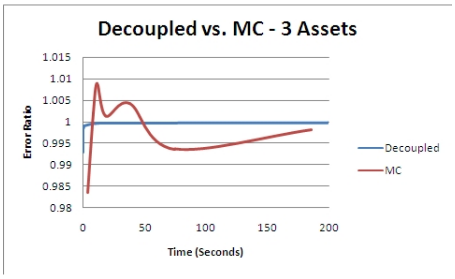


Figure 5.29: Efficiency of Decoupling vs MC in 3 Dimensions

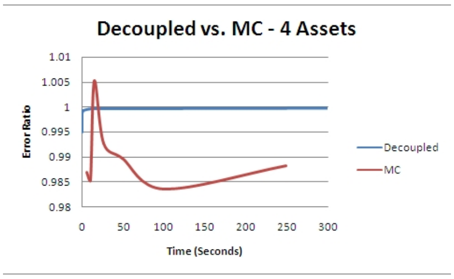


Figure 5.30: Efficiency of Decoupling vs MC in 4 Dimensions

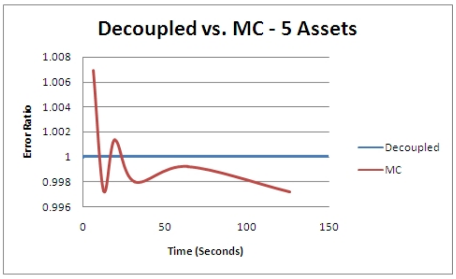


Figure 5.31: Efficiency of Decoupling vs MC in 5 Dimensions

Chapter 6

Conclusion

Tree-based models are pedagogically speaking incredibly important in modern finance [35]. The ability to examine the option structure visually provides a deeper understanding of both the payoff and the backward induction process.

There is a large amount of literature discussing the valuation of options using tree-based models. Most of these articles investigate simple options on one underlying share using a binomial approach. A few authors such as Jabbour et al. [30], Kamrad and Ritchken [37], Heston and Zhou [23] expand the binomial model into higher-state trees, and fewer still investigate the impact of introducing multiple source of underlying risk. This type of expansion is mainly covered by Boyle [11] and Kamrad and Ritchken. This dissertation aimed to do all of the above.

The main objective of the dissertation was to investigate tree-based models and their applications in modern finance. A comprehensive exploration of lattice models, starting with the binomial lattice on one underlying share was desired.

Chapter 2 investigated the CRR model, and expanded the binomial lattice to higher order trees. By increasing the number of possible nodes after each time step, the multinomial model was introduced. The theoretical convergence as well as the actual convergence of the models was depicted and compared. From the analysis it can be seen that theoretically the increase of the number of possible states should lead to an increase in the rate of convergence, but this is not always the case. This might be due to the nature of the put option considered. Further research can be done on investigating the convergence rates for exotic options with payoffs that are not smooth or continuous.

The chapter ended with a discussion surrounding the properties of multinomial models and the limiting case of the multinomial model. It is shown that the multinomial model converges to either a Gaussian Normal or Poisson distribution depending on the assumptions surrounding skewness and kurtosis.

It is interesting to note the relationship between the tree model and the finite difference methods. These are compared for a binomial lattice. It is well documented that the trinomial model is equivalent to the Explicit Finite Difference method [13]. This analysis can be expanded by investigating the higher order lattice models and their relationships to finite difference nets.

The one way of expanding the binomial method is to increase the number of possible states after each time step. This was investigated in Chapter 2. The second means of expansion, is to increase the number of sources of underlying risk. Chapter 3 discusses some of the popular options that are dependent on multiple sources of risk.

Convertible Bonds, Basket Options and Rainbow Options are discussed in detail. Basket and Rainbow options are dependent on more than one underlying share. These include Min and Max options, Product options and Cash or Nothing options. The analytical solutions to these options are discussed, as they are used for comparison purposes in later chapters.

Convertible Bonds are very interesting as they are dependent on equity, interest rate and credit risk. This combination makes them particularly difficult to price. It makes sense that due to these complexities, lattice methods are favoured when determining the prices of these options.

Chapter 4 introduces the Luenberger's method of pricing options that are dependent on multiple source of risk. The product of trees approach investigates whether an option on multiple sources of risk can be reduced to the trees of the underlying risky assets, and then combined. The problem arises when the underlying risk sources are correlated. The resulting joint return process is then used to price the option.

The first hurdle that Luenberger attempts to overcome is the fact that due to the product representation, the number of underlying risky assets is less than the number of nodes at each step. Replication arguments can thus not apply, and the risk neutral probabilities are not uniquely defined. Using utility functions, Luenberger is able to overcome this shortcoming, as the additional arguments are derived from the so-called invariance inequality.

The chapter concludes by looking at the implementation of the product approach on a Max option on two and three underlying shares. An interesting way of viewing an option on three assets as a cube decreasing in size as backward induction occurs is explored.

A convertible bond is also priced using the product of trees approach. The Hung method for allowing for credit risk is used. The graphical representation of the implementation shows how a complex exotic option such as the convertible bond can be valued using the Luenberger approach.

Chapter 5 expands on the idea of increasing the number of underlying sources of risk by investigating alternative approaches to that of Luenberger. The BEG, KR and Decoupled approaches are investigated.

The method of Boyle uses binomial moment matching approach that is shown to be analogous to the product of trees approach. Kamrad and Ritchken expand on the BEG approach by using a trinomial tree. The addition of the so-called stretch parameter λ is shown to be an improvement over the BEG approach, but comes with additional complexity. The parameter needs to be estimated for every problem, and this requires knowledge of the problem at hand. It is shown that the method is quite sensitive to the level of the stretch parameter.

KR mention in their paper that the addition of the stretch parameter overcomes a weakness of the BEG model where negative probabilities can occur. This claim is refuted by Herath and Kumar. They introduce the modified KR approach by including the $O(\Delta t)$ terms that KR discard in their probability approximations. Bands for both the stretch parameter as well as the correlation coefficient are given to ensure positive probabilities

The impact of both the negative probabilities and the exclusion of the $O(\Delta t)$ terms by KR is shown to be minimal when Δt is small. The work by Herath and Kumar [22] does, however, emphasise the importance of investigating the probabilities for each option.

The third model investigated is the so-called decoupled approach by Korn and Muller. This approach looks at decoupling an option on dependent sources of risk into their independent components. An understanding of the variance-covariance matrix is paramount to this method. There are two ways of decoupling the matrix, either by Spectral or Cholesky decomposition. The Korn and Muller approach results in an algorithm that can be used to price options on a number of correlated assets.

The chapter concludes by investigating the convergence and computational intensity of the different methods. It is important to consider the computational intensity of the approaches, as it is logical that the simultaneous addition of underlying risk sources and possible states can lead to a problem blowing up. This explosion is evident in the KR approach when the number of nodes required to be calculated is determined for each method.

The convergence patterns are shown, and it is clear that all the methods' convergence to the analytical prices for the three different options is considered. This is important to note, as this means that the methods work for options when the payoff is continuous and discontinuous. The rate of convergence is determined using the error ratio as with the multinomial approach.

Finally some sensitivity analysis is compiled on the options considered. The Strike level, correlation, and volatility of the underlying shares are varied, and the impact on the option value is discussed.

This dissertation aimed to investigate tree-based models and their applications. The investigation included the derivation of a number of models, as well as the convergence rates and computation intensity of these models. These models are compared using a number of exotic options that are dependent on multiple sources of risk. A novel way of looking at a three dimensional option as a cube is proposed.

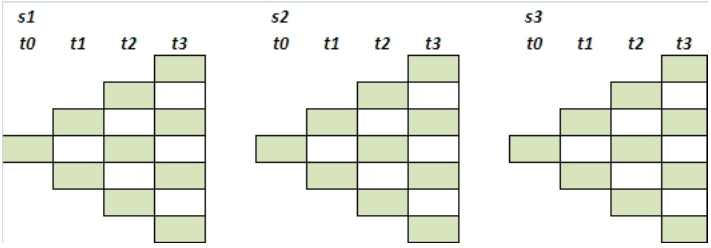


Figure 6.1: Underlying Binomial Trees for 3D option

This expansion of the three trees seen in Fig. 6.1 of a three dimensional option can be seen in Fig. 6.2 below.

The first cube of dimension 4 × 4 × 4 depicts the final nodes for all three underlying shares. The next cube is 3 × 3 × 3 and so on. If the option is European, only the first cube will be dependent on the three shares, i.e. each block "X" contains a function on the three shares. The other cubes will then be calculated using backward induction and the eight probabilities described in Chapter 4 and 5. If the option is American each block in each cube will be dependent on the share values at that time.

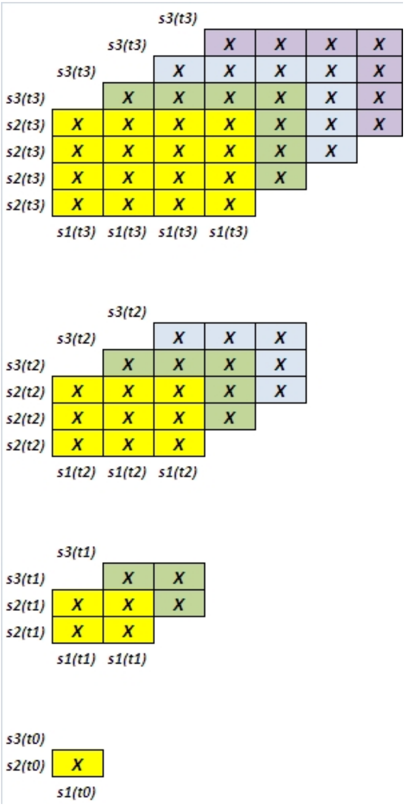


Figure 6.2: Cube View of Backward Induction Process for 3D Option

A number of specific objectives were defined in Chapter 1. These were completed as follows:

1. Survey existing literature surrounding the derivation of Binomial trees.

   A number of articles are referenced throughout the dissertation.

2. Explore the expansion of the Binomial tree, both in terms of expanding the number of states and the number of sources of risk.

   The Binomial model is investigated in Chapter 2. The methods of expanding are discussed in detail. The multinomial expansion is derived and the relationship between the Black-Scholes model and the Binomial tree is highlighted.

3. Investigate some options that are dependent on multiple sources of risk.

A number of options that depend on more than one source of risk are introduced. These include Basket and Rainbow options, as well as Convertible bonds.

4. Provide an in depth analysis on the Product of Trees approach

   Luenberger's Product of trees approach [43] is introduced and explained. The dissertation goes beyond the objective of introducing the model, by expanding the approach to examples where options are dependent on three sources of risk. A visual method of seeing these types of valuations as cubes is introduced.

5. Introduce three other methods that can value options on multiple sources of risk.

   The three main methods of valuing options mentioned by the previous objective are described. These are the BEG model [11], the Kamrad and Ritchken approach [37] and the Decoupled approach [40]. They are deduced and explained in detail.

6. Compare the accuracy and efficiency of the models when implemented on the options discussed using Matlab.

   The methods are tested on the options described by Objective 3. The results are compared to analytical solutions and the accuracy is discussed. The Matlab code used is provided in Appendix C.

7. Investigate the convergence patterns and rates of these models.

   The methods required in the objectives are used to value options, and the convergence patterns are discussed in detail. The rate of convergence for each of the models is measured via the error ratio, and using this ratio, the actual and theoretical convergence is compared.

8. Provide a number of alternative ideas for increasing the amount of literature available on the topic of expanding the Binomial Tree.

   The dissertation highlighted the lack of literature available on the topic of expanding the Binomial tree into multidimensional models. A number of topics that come up in the dissertation can be used as a basis for future research. These include:

a) Relationships between higher order Finite Difference methods and tree models

b) Convergence of multinomial models on exotic options with discontinuous payoffs

c) Application of the decoupling approach on higher order derivatives

d) Computational improvements of the models described

Appendices

Appendix A

Wiener Processes and Ito's Lemma

A.1 Stochastic Processes

A stochastic process can be defined as any process where the variable's value changes over time in an uncertain way [27]. The processes can be divided into continuous or discrete processes, with continuous or discrete variables. A stochastic process is said to have the Markov property if the probability distribution of the price in the future is not dependent on the path followed by the price in the past.

A variable is said to follow a Wiener process or alternatively called Brownian motion if the following two properties hold [27]:

- $\Delta Q = \epsilon\sqrt{\Delta t}$ where ϵ has a standard normal distribution $\sim N(0,1)$ and
- The values for ΔQ in any two time intervals Δt are independent.

We can write a generalised Wiener process with drift as:

$$dx = adt + bdQ. \tag{A.1.1}$$

If a and b are functions of x and t, then we have an Ito process:

$$dx = a(x,t)dt + b(x,t)dQ. \tag{A.1.2}$$

A.2 Share Price Processes

Using the definitions of a Brownian motion from the previous section, we can investigate the process followed by a share price. If the return shown by a share over a period of time is defined as μS, the value after one time period Δt is $\mu S \Delta t$. If we assume that the volatility is zero, and we let Δt tend to 0, we find:

$$dS = \mu S dt, \tag{A.2.1}$$

or

$$\frac{dS}{S} = \mu dt. \tag{A.2.2}$$

If we integrate over time 0 to T, we find:

$$S_T = S_0 e^{\mu T}. \tag{A.2.3}$$

We know that shares have volatility, so if we add it to our equation we find:

$$\frac{dS}{S} = \mu dt + \sigma dQ. \tag{A.2.4}$$

A.3 Ito's Lemma

In order to price derivatives, a very important result from Ito is required. It is called Ito's Lemma. We know that a derivative is a function of the share price. Ito's lemma states that if a variable follows an Ito process [27]:

$$dS = \mu(S,t)dt + \sigma(S,t)dQt, \tag{A.3.1}$$

then a function f that is dependent on x and t follows the following process:

$$df = (\frac{\partial f}{\partial S}\mu + \frac{\partial f}{\partial t} + 0.5\frac{\partial^2 f}{\partial S^2}\sigma^2)dt + \frac{\partial f}{\partial S}\sigma dQ. \tag{A.3.2}$$

We can prove this equation using Taylor expansions:

Proof:

If f is a continuous and differentiable function of S then we know from Taylor that:

$$\Delta f = \frac{df}{dS}\Delta S + 0.5\frac{d^2 f}{dS^2}\Delta S^2 + \frac{1}{6}\frac{d^3 f}{dS^3}\Delta S^3 + ..., \tag{A.3.3}$$

provides the equation when a small change in S results in a small change in f.

If the function f is dependent on two variables S and t then we have:

$$\Delta f = \frac{\partial f}{\partial S}\Delta S + \frac{\partial f}{\partial t}\Delta t + 0.5\frac{\partial^2 f}{\partial S^2}\Delta S^2 + \frac{\partial^2 f}{\partial S^2}\Delta S\Delta t + 0.5\frac{\partial^2 f}{\partial t^2}\Delta t^2 + ... \tag{A.3.4}$$

If function f follows an Ito process, we have:

$$dS = \mu(S,t)dt + \sigma(S,t)dQ. \tag{A.3.5}$$

If this is discretised we find:

$$\Delta S = \mu(S,t)\Delta t + \sigma(S,t)N\sqrt{\Delta t}, \tag{A.3.6}$$

where N is a standard normal variable. If we square Eq. (A.3.7) we have that:

$$\Delta S^2 = \sigma(S,t)^2 N^2 \Delta t. \tag{A.3.7}$$

We can ignore the higher terms of Δt. Since ΔS^2 has a term of order Δt, it cannot be ignored.

Since N has an expected value of 0 and a variance of 1, we know that $E[N^2] = 1$ and thus if Δt tend to zero, $N^2\Delta t$ will be equal to Δt. Thus as Δt tend to zero, Eq. (A.3.7) becomes:

$$\Delta S^2 = \sigma(S,t)^2 \Delta t \tag{A.3.8}$$

If we let Δt and ΔS tend to zero in Eq. (A.3.4), and we substitute the result from Eq. (A.3.8) we find that:

$$df = \frac{\partial f}{\partial S} dS + \frac{\partial f}{\partial t} dt + 0.5 \frac{\partial^2 f}{\partial S^2} \sigma(S,t)^2 dt \tag{A.3.9}$$

We can now substitute Eq. (A.3.7) into Eq. (A.3.9) which results in :

$$df = (\frac{\partial f}{\partial S} dS \mu(S,t) + \frac{\partial f}{\partial t} + 0.5 \frac{\partial^2 f}{\partial S^2} \sigma(S,t)^2) dt + \frac{df}{dS} \sigma(S,t) dQt. \tag{A.3.10}$$

This result can be used to prove the Lognormal property used throughout the dissertation:

If we let $f = lnS$ we can calculate:

$$\frac{\partial f}{\partial S} = \frac{1}{S} \tag{A.3.11}$$

$$\frac{\partial f}{\partial t} = 0 \tag{A.3.12}$$

$$\frac{\partial^2 f}{\partial S^2} = -\frac{1}{S^2}, \tag{A.3.13}$$

and thus we find that from Eq. (A.3.10):

$$df = (\mu(S,t) - 0.5\sigma(S,t)^2) dt + \sigma(S,t) dQ. \tag{A.3.14}$$

Appendix B

Moment Generating Functions

B.1 Moment Generating Functions

If we let Z be a random variable, then the Moment Generating Function (MGF) of Z can be define as:

$$M(t) = E(e^{(}tZ)). \tag{B.1.1}$$

From the definition of Expected value, we have that:

$$M(t) = \int_{-\infty}^{\infty} e^{tz} f(z) dz. \tag{B.1.2}$$

The moments can be generated from Eq. (B.1.1) as follows:

$$M(t) = E(e^{(tZ)}) = E[\sum_{n=0}^{\infty} \frac{Z^n}{n!} t^n] = \sum_{n=0}^{\infty} \frac{E[Z^n]}{n!} t^n. \tag{B.1.3}$$

The derivatives of the MGF at 0 determine all of the moments of the variable, i.e.:

$$M^{(n)}(0) = E(Z^n). \tag{B.1.4}$$

For the standard normal distribution, we have the probability density function:

$$\phi(z) = \frac{1}{\sqrt{2\pi}} e^{-0.5z^2}. \tag{B.1.5}$$

The MGF thus becomes:

$$M(t) = e^{0.5t^2}, \tag{B.1.6}$$

and

$$E[Z^2 n] = \frac{(2n)!}{2^n n!} \tag{B.1.7}$$

$$E[Z^{2n+1}] = 0. \tag{B.1.8}$$

This can be written as:

$$E[Z^n] = \begin{cases} \frac{(n)!}{2^{\frac{n}{2}}\frac{n}{2}!} & \text{for all even n} \\ 0 & \text{for all odd n.} \end{cases}$$

Appendix C

Code

This appendix provides the Matlab code for the results shown in the dissertation. The following figure depicts the flow of the code:

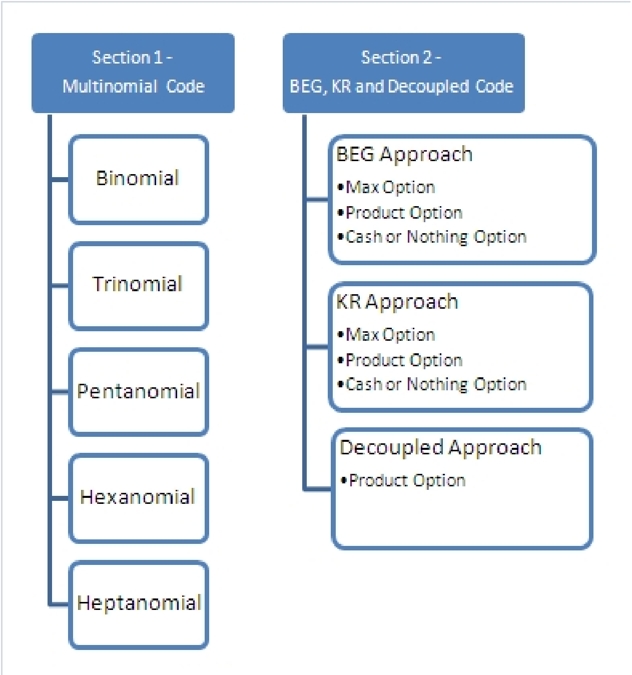


Figure C.1: Flow of Matlab Code

C.1 Matlab Code - Multinomial Models

C.1.1 Binomial Lattice

```
for k =1:100
%Initialise Parameters
 BS        =        5.3081;
 S         =        100
 X         =        90
 r         =        0.05
 Div       =        0
 T         =        1
 vol       =        0.3
 n         =        k
 dt = T / n;
 alpha = r - (vol^2)/2

RR = exp(r * dt);

w1 = -1;
w2 =  1;

P1 = 0.5;     % Initialise probabilities
P2 = 0.5;

Df = exp(-r * dt);

u = [exp(alpha*dt + w1*vol*sqrt(dt)) %initialise Jump Parameter Matrix
 exp(alpha*dt + w2*vol*sqrt(dt))]


for i = 0:n
    State = i + 1;
    Value(State) = max(0,  X - S * u(1,1)^i *  u(2,1)^(n - i));
end


% Work backwards recursively to determine the price of the option
for tt = (n - 1):-1:0
    for i = 0:tt
        State = i + 1;
        Value(State) = (P1 * Value(State+1) + P2 * Value(State)) * Df;
    end
end
Binomial(n) = [Value(1)]
error(n) = Binomial(n) - BS
end
```

C.1.2 Trinomial Lattice

```
for k =1:100
%Initialise Parameters
 BS        =        5.3081;
 S         =        100
 X         =        90
 r         =        0.05
 Div      =        0
 T        =        1
 vol      =        0.3
 n        =        k
 dt = T / n;
 alpha = r - (vol^2)/2

RR = exp(r * dt);

w1 =  -1.732051;
w2 =   0.000000;
w3 =   1.732051;



P1 = 0.166667;     % Initialise probabilities
P2 = 0.666667;
P3 = 0.166667;



Df = exp(-r * dt);

u = [exp(alpha*dt + w1*vol*sqrt(dt)) %initialise Jump Parameter Matrix
 exp(alpha*dt + w2*vol*sqrt(dt))
 exp(alpha*dt + w3*vol*sqrt(dt))]


for i = 0:(2 * n)
    State = i + 1;
    Value(State) = max(0, X - S * u(1,1)^max(i - n, 0) *
    u(3,1)^ max(n - i, 0));
end



% Works backwards recursively to determine the price of the option
for tt = (n - 1):-1:0
    for i = 0:(tt * 2)
        State = i + 1;
        Value(State) = (P1 * Value(State+2)
        + P2 * Value(State+1) + P3 * Value(State)) * Df;
    end
end
Trinomial(n) = [Value(1)]
error(n) = Trinomial(n) - BS
end
```

C.1.3 Pentanomial Lattice

```
for k = 1:100
%Initialise Parameters
 BS       =        5.3081;
 S      =      100
 X      =      90
 r      =      0.05
 Div    =      0
 T      =      1
 vol    =      0.3
 n      =      k
 dt = T / n;
 alpha = r - (vol^2)/2

RR = exp(r * dt);
w1 = -2.738608;
w2 = -1.369304;
w3 =  0.000000;
w4 =  1.369304;
w5 =  2.738608;


P1 = 0.013333;    % Initialise probabilities
P2 = 0.213334;
P3 = 0.546666;
P4 = 0.213334;
P5 = 0.013333;


Df = exp(-r * dt);

u = [exp(alpha*dt + w1*vol*sqrt(dt)) %initialise Jump Parameter Matrix
 exp(alpha*dt + w2*vol*sqrt(dt))
 exp(alpha*dt + w3*vol*sqrt(dt))
 exp(alpha*dt + w4*vol*sqrt(dt))
 exp(alpha*dt + w5*vol*sqrt(dt))]


for i = 0:n                         %Finding values at the final time
     State = i + 1;
     Value(State) = max(0,X-S*u(4,1)^i*u(5,1)^(n-i));
end;
for i = n+1:2*n
     State = i + 1;
     Value(State) = max(0,X-S*u(3,1)^(i-n)*u(4,1)^(2*n-i));
end;
for i = 2*n+1:3*n
     State = i + 1;
     Value(State) = max(0,X-S*u(2,1)^(i-2*n)*u(3,1)^(3*n-i));
end;
for i = 3*n+1:4*n
     State = i + 1;
     Value(State) = max(0,X-S*u(1,1)^(i-3*n)*u(2,1)^(4*n-i));
end;
```

```
% Works backwards recursively to determine the price of the option
for tt = (n - 1):-1:0
     for i = 0:(tt * 4)
          State = i + 1;
          Value(State) = (P1 * Value(State + 4) +
           P2 * Value(State + 3) + P3 * Value(State+2) +
           P4*Value(State+1) + P5*Value(State)) * Df;
     end
end
Pentanomial(n) = [Value(1)]
error(n) = Pentanomial(n) - BS
end
```

C.1.4 Hexanomial Lattice

```
for k =1:100
%Initialise Parameters
 BS        =         5.3081;
 S         =         100
 X         =         90
 r         =         0.05
 Div       =         0
 T         =         1
 vol       =         0.3
 n         =         k
 dt = T / n;
 alpha = r - (vol^2)/2

RR = exp(r * dt);
w1 = -3.189031;
w2 = -1.913419;
w3 = -0.637806;
w4 =  0.637806;
w5 =  1.913419;
w6 =  3.189031;


P1 = 0.003316;    % Initialise probabilities
P2 = 0.081193;
P3 = 0.415492;
P5 = 0.081193;
P6 = 0.003316;


Df = exp(-r * dt);

u = [exp(alpha*dt + w1*vol*sqrt(dt)) %initialise Jump Parameter Matrix
 exp(alpha*dt + w2*vol*sqrt(dt))
 exp(alpha*dt + w3*vol*sqrt(dt))
 exp(alpha*dt + w4*vol*sqrt(dt))
 exp(alpha*dt + w5*vol*sqrt(dt))
 exp(alpha*dt + w6*vol*sqrt(dt))]
```

```
for i = 0:n                              %Finding values at the final time
    State = i + 1;
    Value(State) = max(0,X-S*u(5,1)^i*u(6,1)^(n-i));
end;
for i = n+1:2*n
    State = i + 1;
    Value(State) = max(0,X-S*u(4,1)^(i-n)*u(5,1)^(2*n-i));
end;
for i = 2*n+1:3*n
    State = i + 1;
    Value(State) = max(0,X-S*u(3,1)^(i-2*n)*u(4,1)^(3*n-i));
end;
for i = 3*n+1:4*n
    State = i + 1;
    Value(State) = max(0,X-S*u(2,1)^(i-3*n)*u(3,1)^(4*n-i));
end;
for i = 4*n+1:5*n
    State = i + 1;
    Value(State) = max(0,X-S*u(1,1)^(i-4*n)*u(2,1)^(5*n-i));
end;\\

 % Works backwards recursively to determine the price of the option
for tt = (n - 1):-1:0
    for i = 0:(tt * 5)
        State = i + 1;
        Value(State) = (P1 * Value(State + 5) +
        P2 * Value(State + 4) + P3 * Value(State+3)
        + P4*Value(State+2) + P5*Value(State+1)
        + P6*Value(State)) * Df;
    end
end
Hexanomial(n) = [Value(1)]
error(n) = Hexanomial(n) - BS
end}
```

C.1.5 Heptanomial Lattice

```
for k =1:100
%Initialise Parameters
 BS      =       5.3081;
 S       =       100
 X       =       90
 r       =       0.05
 Div     =       0
 T       =       1
 vol     =       0.3
 n       =       k
 dt = T / n;
 alpha = r - (vol^2)/2

RR = exp(r * dt);
```

```
w1 =  -3.594559;
w2 =  -2.396373;
w3 =  -1.198186;
w4 =   0.000000;
w5 =   1.198186;
w6 =   2.396373;
w7 =   3.594559


P1 =   0.000802        % Initialise probabilities
P2 =   0.026810;
P3 =   0.233813;
P4 =   0.477150;
P5 =   0.233813;
P6 =   0.026810;
P7 =   0.000802;


Df = exp(-r * dt);

u = [exp(alpha*dt + w1*vol*sqrt(dt)) %initialise Jump Parameter Matrix
 exp(alpha*dt + w2*vol*sqrt(dt))
 exp(alpha*dt + w3*vol*sqrt(dt))
 exp(alpha*dt + w4*vol*sqrt(dt))
 exp(alpha*dt + w5*vol*sqrt(dt))
 exp(alpha*dt + w6*vol*sqrt(dt))
 exp(alpha*dt + w7*vol*sqrt(dt))]


for i = 0:n                          %Finding values at the final time
    State = i + 1;
    Value(State) = max(0,X-S*u(6,1)^i*u(7,1)^(n-i));
end;
for i = n+1:2*n
    State = i + 1;
    Value(State) = max(0,X-S*u(5,1)^(i-n)*u(6,1)^(2*n-i));
end;
for i = 2*n+1:3*n
    State = i + 1;
    Value(State) = max(0,X-S*u(4,1)^(i-2*n)*u(5,1)^(3*n-i));
end;
for i = 3*n+1:4*n
    State = i + 1;
    Value(State) = max(0,X-S*u(3,1)^(i-3*n)*u(4,1)^(4*n-i));
end;
for i = 4*n+1:5*n
    State = i + 1;
    Value(State) = max(0,X-S*u(2,1)^(i-4*n)*u(3,1)^(5*n-i));
end;
for i = 5*n+1:6*n
    State = i + 1;
    Value(State) = max(0,X-S*u(1,1)^(i-5*n)*u(2,1)^(6*n-i));
```

```
end;
 % Works backwards recursively to determine the price of the option
for tt = (n - 1):-1:0
    for i = 0:(tt * 5)
        State = i + 1;
        Value(State) = (P1 * Value(State + 6) +
        P2 * Value(State + 5) + P3 * Value(State+4) +
        P4*Value(State+3) + P5*Value(State+2) +
        P6*Value(State+1)+P7*Value(State)) * Df;
    end
end
Heptanomial(n) = [Value(1)]
error(n) = Heptanomial(n) - BS
end
```

C.2 Matlab Code - Higher Order Models

C.2.1 BEG Model Code - Max Option

```
for k = 1:10
    tic
%Initialise Parameters
 Boyle = 5.488;
 S_1       =       40;
 S_2       =       40;
 X      =      40;
 r      =      0.04879;
 T      =      0.58333;
 vol_1     =       0.2;
 vol_2     =       0.3;
 corr = 0.5;
 n      =      k;
 dt = T / n;
mu_1 = r - (vol_1^2)/2;
mu_2 = r - (vol_2^2)/2;

RR = exp(r * dt);

p1 = 0.25*(1+corr + sqrt(dt)*((mu_1/vol_1)+(mu_2/vol_2)));
p2 = 0.25*(1-corr + sqrt(dt)*((mu_1/vol_1)-(mu_2/vol_2)));
p3 = 0.25*(1-corr + sqrt(dt)*(-(mu_1/vol_1)+(mu_2/vol_2)));
p4 = 0.25*(1+corr + sqrt(dt)*(-(mu_1/vol_1)-(mu_2/vol_2)));
Df = exp(-r * dt);

u_1 = exp(vol_1*sqrt(dt));
u_2 = exp(vol_2*sqrt(dt));
d_1 = 1/u_1;
d_2 = 1/u_2;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Share values at time T
for i = 1:n+1;
    for j = i:n+1;
        S1(i,j) = max(S_1*u_1^(j-i)*d_1^(i-1)-X,0);
        S2(i,j) = max(S_2*u_2^(j-i)*d_2^(i-1)-X,0);
    end
end

T = zeros(n+1,n+1,n+1);
U = zeros(n+1,n+1,n+1);
P = zeros(n+1,n+1,n+1);
for j = 1:1:n+1
        T(:,:,j) = repmat(S1(:,j),1,n+1);
        U(:,:,j) = repmat(S2(:,j),1,n+1);
        P(:,:,j) = max(T(:,:,j),U(:,:,j)');
end

E = zeros(n+1,n+1,n+1);
E(:,:,n+1) = P(:,:,n+1);
for j = n:-1:1
    for i = 1:1:j
        for k = 1:1:j
        E(i,k,j) = (p1*E(i,k,j+1)+p2*E(i,k+1,j+1)
+p3*E(i+1,k,j+1)+p4*E(i+1,k+1,j+1))*Df;
        end
    end
end
European(n) = [E(1,1,1)]
toc
end
```

C.2.2 BEG Model Code - Product Option

```
for k = 1:250
    tic
%Initialise Parameters
 BS = 3.26214;
 S_1         =        22;
 S_2         =        20;
 X        =        20;
 r        =        0.1;
 T        =        1;
 vol_1      =        0.2;
 vol_2      =        0.25;
 corr = 0.5;
 n        =        k;
 dt = T / n;
mu_1 = r - (vol_1^2)/2;
mu_2 = r - (vol_2^2)/2;

RR = exp(r * dt);

p1 = 0.25*(1+corr + sqrt(dt)*((mu_1/vol_1)+(mu_2/vol_2)));
```

```
p2 = 0.25*(1-corr + sqrt(dt)*((mu_1/vol_1)-(mu_2/vol_2)));
p3 = 0.25*(1-corr + sqrt(dt)*(-(mu_1/vol_1)+(mu_2/vol_2)));
p4 = 0.25*(1+corr + sqrt(dt)*(-(mu_1/vol_1)-(mu_2/vol_2)));
Df = exp(-r * dt);

u_1 = exp(vol_1*sqrt(dt));
u_2 = exp(vol_2*sqrt(dt));
d_1 = 1/u_1;
d_2 = 1/u_2;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Share values at time T
for i = 1:n+1;
    for j = i:n+1;
        S1(i,j) = S_1*u_1^(j-i)*d_1^(i-1);
        S2(i,j) = S_2*u_2^(j-i)*d_2^(i-1);
    end
end

T = zeros(n+1,n+1,n+1);
U = zeros(n+1,n+1,n+1);
P = zeros(n+1,n+1,n+1);
for j = 1:1:n+1
        T(:,:,j) = repmat(S1(:,j),1,n+1);
        U(:,:,j) = repmat(S2(:,j),1,n+1);
        P(:,:,j) = max(sqrt(T(:,:,j).*U(:,:,j)') - X,0);
end
A = zeros(n+1,n+1,n+1);
A(:,:,n+1) = P(:,:,n+1);
E = zeros(n+1,n+1,n+1);
E(:,:,n+1) = P(:,:,n+1);
for j = n:-1:1
    for i = 1:1:j
        for k = 1:1:j
        E(i,k,j) = (p1*E(i,k,j+1)+p2*E(i,k+1,j+1)
+p3*E(i+1,k,j+1)+p4*E(i+1,k+1,j+1))*Df;
        end
    end
end
European(n) = [E(1,1,1)]
toc
end
```

C.2.3 BEG Model Code - Digital Option

```
for k = 1:240
    tic
%Initialise Parameters
 S_1     =     12;
 S_2     =     12;
 r     =     0.1;
 T       =        1;
 vol_1     =        0.2;
 vol_2     =        0.25;
 corr = 0.5;
 n       =        k;
 dt = T / n;
mu_1 = r - (vol_1^2)/2;
mu_2 = r - (vol_2^2)/2;

RR = exp(r * dt);

p1 = 0.25*(1+corr + sqrt(dt)*((mu_1/vol_1)+(mu_2/vol_2)));
p2 = 0.25*(1-corr + sqrt(dt)*((mu_1/vol_1)-(mu_2/vol_2)));
p3 = 0.25*(1-corr + sqrt(dt)*(-(mu_1/vol_1)+(mu_2/vol_2)));
p4 = 0.25*(1+corr + sqrt(dt)*(-(mu_1/vol_1)-(mu_2/vol_2)));
Df = exp(-r * dt);

u_1 = exp(vol_1*sqrt(dt));
u_2 = exp(vol_2*sqrt(dt));
d_1 = 1/u_1;
d_2 = 1/u_2;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Share values at time T
for i = 1:n+1;
    for j = i:n+1;
        S1(i,j) = S_1*u_1^(j-i)*d_1^(i-1) - 17;
        S2(i,j) = S_2*u_2^(j-i)*d_2^(i-1) - 20;
    end
end



T = zeros(n+1,n+1,n+1);
U = zeros(n+1,n+1,n+1);
P = zeros(n+1,n+1,n+1);
for j = 1:1:n+1
        T(:,:,j) = repmat(S1(:,j),1,n+1);
        U(:,:,j) = repmat(S2(:,j),1,n+1);

end


for j = 1:1:n+1
    for i = 1:1:j
        for k = 1:1:j
           P(i,k,j) = 0;
       if T(i,k,j) >= 0 & U(k,i,j) >= 0;
           P(i,k,j) = 100;
       end
        end
    end
end

E = zeros(n+1,n+1,n+1);
E(:,:,n+1) = P(:,:,n+1)';
for j = n:-1:1
    for i = 1:1:j
        for k = 1:1:j
         E(i,k,j) = (p1*E(i,k,j+1)+p2*E(i,k+1,j+1)
+p3*E(i+1,k,j+1)+p4*E(i+1,k+1,j+1))*Df;
        end
    end
end
European(n) = [E(1,1,1)]
toc
end
```

C.2.4 KR Model Code - Max Option

```
for k = 1:160
    tic
%Initialise Parameters
 Boyle = 5.488;
 S_1        =          40;
 S_2        =          40;
 X        =          40;
 r        =          0.04879;
 T        =          0.58333;
 vol_1      =          0.2;
 vol_2      =          0.3;
 corr  = 0.5;
 n        =          k;
 dt = T / n;
mu_1 = r - (vol_1^2)/2;
mu_2 = r - (vol_2^2)/2;
lambda = 1.11803;

RR = exp(r * dt);

p1 = 0.25*((1+corr)/(lambda^2) + (sqrt(dt)/lambda)*((mu_1/vol_1)+(mu_2/vol_2)));
p2 = 0.25*((1-corr)/(lambda^2) + (sqrt(dt)/lambda)*((mu_1/vol_1)-(mu_2/vol_2)));
p3 = 0.25*((1+corr)/(lambda^2) + (sqrt(dt)/lambda)*(-(mu_1/vol_1)-(mu_2/vol_2)));
p4 = 0.25*((1-corr)/(lambda^2) + (sqrt(dt)/lambda)*(-(mu_1/vol_1)+(mu_2/vol_2)));
p5 = 1 - (1/(lambda^2));
Df = exp(-r * dt);

u_1 = exp(lambda*vol_1*sqrt(dt));
u_2 = exp(lambda*vol_2*sqrt(dt));
d_1 = 1/u_1;
d_2 = 1/u_2;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Share values at time T
for j=1:n+1
    S1(1,j) = S_1*u_1^(j-1)
    S2(1,j) = S_2*u_2^(j-1)
for i=1:2*n+1
if i==j
S1(i,j) = S_1
S2(i,j) = S_2
end
if i == 2*j-1
        S1(i,j) = S_1*d_1^(j-1)
        S2(i,j) = S_2*d_2^(j-1)
end
end
end

for j = 3:n+1
    for i = 2: 2*(j-1)
S1(i,j) = S1(i-1,j-1);
S2(i,j) = S2(i-1,j-1);
    end
end

T = zeros(2*n+1,2*n+1,n+1);
U = zeros(2*n+1,2*n+1,n+1);
P = zeros(2*n+1,2*n+1,n+1);
for j = 1:1:n+1
        T(:,:,j) = repmat(S1(:,j),1,2*n+1);
        U(:,:,j) = repmat(S2(:,j),1,2*n+1);
        P(:,:,j) = max(max(T(:,:,j),U(:,:,j)')-X,0);
end

E = zeros(2*n+1,2*n+1,n+1);
E(:,:,n+1) = P(:,:,n+1);
for j = n:-1:1
    for i = 1:1:j*2-1
        for k = 1:1:j*2-1
        E(i,k,j) = (p1*E(i,k,j+1)+p2*E(i,k+2,j+1)
+p4*E(i+2,k,j+1)+p3*E(i+2,k+2,j+1)+ p5*E(i+1,k+1,j+1))*Df;
        end
    end
end
European(n) = [E(1,1,1)]
toc
end
```

C.2.5 KR Model Code - Product Option

```
for k = 1:100
    tic
%Initialise Parameters
 BS = 3.26214;
 S_1        =        22;
 S_2        =        20;
 X        =        20;
 r        =        0.1;
 T        =        1;
 vol_1        =        0.2;
 vol_2        =        0.25;
 corr = 0.5;
 n        =        k;
 dt = T / n;
mu_1 = r - (vol_1^2)/2;
mu_2 = r - (vol_2^2)/2;
lambda = 1.11803;

RR = exp(r * dt);

p1 = 0.25*((1+corr)/(lambda^2) + (sqrt(dt)/lambda)*((mu_1/vol_1)+(mu_2/vol_2)));
p2 = 0.25*((1-corr)/(lambda^2) + (sqrt(dt)/lambda)*((mu_1/vol_1)-(mu_2/vol_2)));
p3 = 0.25*((1+corr)/(lambda^2) + (sqrt(dt)/lambda)*(-(mu_1/vol_1)-(mu_2/vol_2)));
p4 = 0.25*((1-corr)/(lambda^2) + (sqrt(dt)/lambda)*(-(mu_1/vol_1)+(mu_2/vol_2)));
p5 = 1 - (1/(lambda^2));
Df = exp(-r * dt);

u_1 = exp(lambda*vol_1*sqrt(dt));
u_2 = exp(lambda*vol_2*sqrt(dt));
d_1 = 1/u_1;
d_2 = 1/u_2;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Share values at time T
for j=1:n+1
    S1(1,j) = S_1*u_1^(j-1)
    S2(1,j) = S_2*u_2^(j-1)
for i=1:2*n+1
if i==j
S1(i,j) = S_1
S2(i,j) = S_2
end
if i == 2*j-1
        S1(i,j) = S_1*d_1^(j-1)
        S2(i,j) = S_2*d_2^(j-1)
end
end
end

for j = 3:n+1
    for i = 2: 2*(j-1)
S1(i,j) = S1(i-1,j-1);
S2(i,j) = S2(i-1,j-1);
    end
end
```

```
T = zeros(2*n+1,2*n+1,n+1);
U = zeros(2*n+1,2*n+1,n+1);
P = zeros(2*n+1,2*n+1,n+1);
for j = 1:1:n+1
        T(:,:,j) = repmat(S1(:,j),1,2*n+1);
        U(:,:,j) = repmat(S2(:,j),1,2*n+1);
        P(:,:,j) = max(sqrt(T(:,:,j).*U(:,:,j)') - X,0);
end

E = zeros(2*n+1,2*n+1,n+1);
E(:,:,n+1) = P(:,:,n+1);
for j = n:-1:1
    for i = 1:1:j*2-1
        for k = 1:1:j*2-1
        E(i,k,j) = (p1*E(i,k,j+1)+p2*E(i,k+2,j+1)
+p4*E(i+2,k,j+1)+p3*E(i+2,k+2,j+1)+ p5*E(i+1,k+1,j+1))*Df;
        end
    end
end
European(n) = [E(1,1,1)]
toc
end
```

C.2.6 KR Model Code - Digital Option

```
for k = 1:160
    tic
%Initialise Parameters
 S_1       =       12;
 S_2       =       12;
 r       =       0.1;
 T       =       1;
 vol_1     =       0.2;
 vol_2     =       0.25;
 corr = 0.5;
 n       =       k;
 dt = T / n;
 mu_1 = r - (vol_1^2)/2;
mu_2 = r - (vol_2^2)/2;
lambda = 1.11803;

RR = exp(r * dt);

p1 = 0.25*((1+corr)/(lambda^2) + (sqrt(dt)/lambda)*((mu_1/vol_1)+(mu_2/vol_2)));
p2 = 0.25*((1-corr)/(lambda^2) + (sqrt(dt)/lambda)*((mu_1/vol_1)-(mu_2/vol_2)));
p3 = 0.25*((1+corr)/(lambda^2) + (sqrt(dt)/lambda)*(-(mu_1/vol_1)-(mu_2/vol_2)));
p4 = 0.25*((1-corr)/(lambda^2) + (sqrt(dt)/lambda)*(-(mu_1/vol_1)+(mu_2/vol_2)));
p5 = 1 - (1/(lambda^2));
Df = exp(-r * dt);

u_1 = exp(lambda*vol_1*sqrt(dt));
u_2 = exp(lambda*vol_2*sqrt(dt));
```

```
d_1 = 1/u_1;
d_2 = 1/u_2;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Share values at time T
for j=1:n+1
    S1(1,j) = S_1*u_1^(j-1)
    S2(1,j) = S_2*u_2^(j-1)
for i=1:2*n+1
if i==j
S1(i,j) = S_1
S2(i,j) = S_2
end
if i == 2*j-1
        S1(i,j) = S_1*d_1^(j-1)
        S2(i,j) = S_2*d_2^(j-1)
end
end
end

for j = 3:n+1
    for i = 2: 2*(j-1)
S1(i,j) = S1(i-1,j-1);
S2(i,j) = S2(i-1,j-1);
    end
end

T = zeros(2*n+1,2*n+1,n+1);
U = zeros(2*n+1,2*n+1,n+1);

for j = 1:1:n+1
        T(:,:,j) = repmat(S1(:,j),1,2*n+1);
        U(:,:,j) = repmat(S2(:,j),1,2*n+1);
end


P = zeros(2*n+1,2*n+1,n+1);
for j = 1:1:n+1
    for i = 1:1:j
        for k = 1:1:j
           P(i,k,j) = 0;
       if T(i,k,j)-17 >= 0 & U(k,i,j)-20 >= 0;
           P(i,k,j) = 100;
       end
        end
    end
end


E = zeros(2*n+1,2*n+1,n+1);
E(:,:,n+1) = P(:,:,n+1);
for j = n:-1:1
```

```
     for i = 1:1:j*2-1
          for k = 1:1:j*2-1
          E(i,k,j) = (p1*E(i,k,j+1)+p2*E(i,k+2,j+1)
+p4*E(i+2,k,j+1)+p3*E(i+2,k+2,j+1)+ p5*E(i+1,k+1,j+1))*Df;
          end
     end
end
European(n) = [E(1,1,1)]
toc
end
```

C.2.7 Decoupled Model Code - Product Option

```
for k = 1:10
     tic
%Initialise Parameters
 BS = 3.26214;
 S_1        =         22;
 S_2        =         20;
 X       =         20;
 r       =         0.1;
 T       =         1;
 vol_1      =         0.2;
 vol_2      =         0.25;
 corr = 0.5;
 n       =         k;
 dt = T / n;
Df = exp(-r * dt);

M = [vol_1^2 corr*vol_1*vol_2
     corr*vol_1*vol_2 vol_2^2];

R = [r
      r];
Sigma2 = [vol_1^2
           vol_2^2]

G = chol(M);
Ginv = inv(G);
alpha = Ginv*(R - 0.5*Sigma2);


Y = Ginv*[log(S_1);
log(S_2)];
Y1 = zeros(n+1,n+1);
Y2 = zeros(n+1,n+1);
Y1(1,1) = Y(1);
Y2(1,1) = Y(2);
for j = 2:n+1;
 Y1(1,j) = Y1(1,j-1) + alpha(1)*dt + sqrt(dt);
 Y2(1,j) = Y2(1,j-1) + alpha(2)*dt + sqrt(dt);
end
```

```
for i = 2:n+1;
    for j = i:n+1;
Y1(i,j) =  Y1(i-1,j-1) + alpha(1)*dt - sqrt(dt);
Y2(i,j) =  Y2(i-1,j-1) + alpha(2)*dt - sqrt(dt);
    end
end



for i = 1:n+1;
for j = 1:n+1;
H1(i,j) = exp(G(1,:)*[Y1(i,j) Y2(i,j)]')';
H2(i,j) = exp(G(2,:)*[Y1(i,j) Y2(i,j)]')';
end
end

T = zeros(n+1,n+1,n+1);
U = zeros(n+1,n+1,n+1);
P = zeros(n+1,n+1,n+1);
for j = 1:1:n+1
        T(:,:,j) = repmat(H1(:,j),1,n+1);
        U(:,:,j) = repmat(H2(:,j),1,n+1);
        P(:,:,j) = max(sqrt(T(:,:,j).*U(:,:,j)') - X,0);
end
p=0.25
E = zeros(n+1,n+1,n+1);
E(:,:,n+1) = P(:,:,n+1);
for j = n:-1:1
    for i = 1:1:j
        for k = 1:1:j
        E(i,k,j) = (p*E(i,k,j+1)+p*E(i,k+1,j+1)
+p*E(i+1,k,j+1)+p*E(i+1,k+1,j+1))*Df;
        end
    end
end
Decoupled(n) = [E(1,1,1)]
toc
end
```

Bibliography

[1] D. Ahn, S. Figlewski and B. Gao, *Pricing discrete barrier options with an adaptive mesh model*, The Journal of Derivatives (1999).

[2] J. Alford and N. Webber, *Very high order lattice methods for one factor models*, (2001).

[3] A. Allison, H. Beattie and L. Olsen, *Convertibles and the k-factor, implications of the credit-equity relationship in the current market*, Convertible Bonds research - Barclays Capital (2007).

[4] A. Andricopoulos, P. Duck, D. Newton and M. Widdicks, *On the enhanced convergence of standard lattice methods for option pricing*, Journal of Futures Markets **22** (2002), 315–338.

[5] E. Ayache, P. Forsyth and K. Vetzal, *Valuation of convertible bonds with credit risk*, Journal of Derivatives **11** (2003), 9–29.

[6] K. Arrow and G. Debreu, *Existence of an equilibrium for a competitive economy*, Econometrica **22** (1954), no. 3, 265–290.

[7] S. Benninga and Z. Wiener, *Binomial Option Pricing, the Black-Scholes Option Pricing Formula and Exotic Options*,Mathematica in eduction and research **6** (1997).

[8] F. Black and M. Scholes, *The pricing of options and corporate liabilities*, Journal of Political Economy **81** (1973).

[9] P. Boyle, *Valuation of derivative securities involving several assets using discrete time methods*, The Review of Financial Studies **2** (1990), 241–250.

[10] P. Boyle and Y. Tse, *An algortihm for computing values of options on the maximum or minimum of several assets*, Journal of Financial and Quantitative Analysis **25** (1990), 215–227.

[11] P. Boyle, J. Evnine and S. Gibbs, *Numerical evaluation of multivariate contingent claims*, Journal of Economic Dynamics and Control **9** (1989), 131–139.

[12] M. Brennan and E. Schwartz, *The valuation of american put options*, Journal of Finance **32** (1977).

[13] ______, *Finite difference methods and jump processes arising in the pricing of contingent claims: A synthesis*, Journal of Financial and Quantitative Analysis **12** (1978), 461–474.

[14] D. Chambers and Q. Lu, *A tree model for pricing convertible bonds with equity, interest and default risk*, Journal of Derivatives (2007), 25–46.

[15] H. Chen, D. Chen and S. Chung, *The accuracy and efficiency of alternative option pricing approaches relative to a log-transformed trinomial model*, Journal of Futures Markets **22** (2002), 557–577.

[16] J. Cox and S. Ross, *The valuation of options for alternative stochastic process*, Journal of Financial Economics **3** (1976).

[17] J. Cox, S. Ross and M. Rubinstein, *Option pricing: A simplified approach*, Journal of Financial Economics **7** (1979), 229–263.

[18] E. Deutsch, *Implementing the adaptive mesh model*, 2008, Carnegie Mellon University.

[19] S. Easton, *A note on modified lattice approaches to option pricing*, Journal of Futures Markets **16** (1996), 585–594.

[20] S. Hariparsad, *The valuation and calibration of convertible bonds*, Master's thesis, The University of Pretoria, 2009.

[21] D. Heath and S. Herzel, *Efficient option valuation using trees*, Applied Mathematics of Finance **9** (2002), 163–178.

[22] H. Herath and P. Kumar, *Multinomial approximating models for options*, Master's thesis, University of Northern British Columbia,, 2004.

[23] S. Heston and G. Zhou, *On the rate of convergence of discrete-time contingent claims*, Journal of Mathematical Finance **10** (2000), 53–75.

[24] R. Heynen and H. Kat, *Discrete partial barrier options with a moving barrier*, Journal of Financial Engineering **5** (1996).

[25] T. Ho and S. Lee, *Term structure movements and pricing interest rate contingent claims*, Journal of Finance **41** (1986).

[26] T. Ho, R. Stapleton and M. Subrahmanyam, *Multivariate binomial approximations for asset prices with nonstationary variance and covariance characteristics*, The Review of Financial Studies **8** (1995), 1125–1152.

[27] J. Hull, *Options, futures and other derivatives*, seventh ed., Pearson Prentice Hall, 2009.

[28] J. Hull and A. White, *Valuing derivative securities using the explicit finite difference method*, Journal of Financial and Quantitative Analysis **25** (1990), 87–100.

[29] M. Hung and J. Wang, *Pricing convertible bonds subject to default risk*, Journal of Derivatives **10** (2002), 75–87.

[30] G. Jabbour, M Kramin, T Kramin and S. Young, *Multinomial lattices and derivatives pricing*, vol. 8, Rutgers University, 2005.

[31] R. Jarrow and A. Rudd, *Option pricing*, Homewood, IL: Dow Jones-Irwin Publishing, 1983.

[32] L. Jiang and M. Dai, *Convergence of binomial tree methods for european/american path-dependent options*, SIAM Journal on Numerical Analysis **42** (2005), 1094–1109.

[33] H. Johnson, *Options on the maximum or the minimum of several assets*, Journal of Financial and Quantitative Analysis **22** (1987), 277–283.

[34] M. Joshi, *Achieving smooth asymptotics for the prices of european options in binomial trees*, Journal of Quantitative Finance **9** (2009), 171–176.

[35] ______, *Achieving higher order convergence for the prices of european options in binomial trees*, Journal of Mathematical Finance **20** (2010), 89–103.

[36] M. Joshi and M. Staunton, *On the analytical/numerical pricing of american put options against binomial tree prices*, Journal of Quantitative Finance **12** (2012), 17–20.

[37] B. Kamrad and P. Ritchken, *Multinomial approximating models for options with k-state variables*, Management Science **37** (1991), 1640–1652.

[38] N. Kan, *Generalised multinomial crr option pricing model and its black scholes type limit*, Ph.D. thesis, Gotingen University, 2005.

[39] W. Kim, K. Moon and H. Kim, *Adaptive binomial tree methods for two-asset options*, Master's thesis, Korea University, 2004.

[40] R. Korn and S. Muller, *The decoupling approach to binomial pricing of multi-asset options*, Journal of Computational Finance **12** (2009), 1–30.

[41] Y. Kwok and K. Lau, *Accuracy and reliability considerations of option pricing algorithms*, The Journal of Futures Markets **21** (2001), 875–903.

[42] D. Luenberger, *Investment science*, Oxford University Press, 1998.

[43] ______, *Products of trees for investment analysis*, Journal of Mathematics and Economics **22** (1998), 1403–1417.

[44] D. Madan, F. Milne and H. Shefrin, *The multinomial option pricing model and its brownian and poisson limits*, The Review of Financial Studies **2** (1989), no. 2, 251–265.

[45] W. Margrabe, *The value of an option to exchange one asset for another*, Journal of Finance **23** (1978), 177–186.

[46] R. Merton, *Theory of rational option pricing*, Bell Journal of Economics and Management Science **4** (1973), 141–183.

[47] A. Mijatovic, *Spectral properties of trinomial trees*, Proceedings: Mathematical, Physical and Engineering Sciences **463** (2007), 1681–1696.

[48] K. Natcheva, *On numerical pricing methods of innovative financial products*, Ph.D. thesis, TU Kaiserslautern, 2006.

[49] J. Norstad, *An introduction to utility theory*, Master's thesis, The North Western University, 2011.

[50] A. Omberg, *A note on the convergence of binomial pricing and compound option models*, Journal of Finance **42** (1987), 463–469.

[51] P. Ouwehand and G. West, *Pricing rainbow options*, vol. 8, University of the Witwatersrand, 2006.

[52] J. Primbs and Y. Yamada, *Properties of multinomial lattices with cumulants for option pricing and hedging*, American Control Center (2006).

[53] R. Rebib, *Barrier options pricing*, Master's thesis, National University of Singapore, 2008.

[54] R. Rendleman and B. Bartter, *Two-state option pricing*, The Journal of Finance **34** (1979), 1093–1110.

[55] C. Shea, *Numerical valuation of discrete barrier options with the adaptive mesh model and other competing techniques*, Master's thesis, National Taiwan University, 2008.

[56] J. Stampfli and V. Goodmanl, *The mathematics of finance: Modelling and hedging*, Brooks/Cole, 2001.

[57] R. Stulz, *Options on the minimum or the maximum of two risky assets*, Journal of Financial Economics **10** (1982), 161–185.

[58] T. Tian, *A modifified approach to option pricing*, Journal of Futures Markets **13** (1993), 563–577.

[59] K. Tsiverotis and C. Fernandes, *Valuing convertible bonds with credit risk*, Journal of Fixed Income **8** (1998), 95–102.

[60] A. Yigitbasioglu, *Pricing convertible bonds with interest rate, equity, credit, fx and volatility risk*, Tech. report, Universiy of Reading, 2002.

[61] A. Zadikov, *Methods of pricing convertible bonds*, Master's thesis, The University of Cape Town, 2010.