UNIVERSITY OF PIRAEUS

Department of Banking and Financial Management

M.Sc. in Banking and Finance

Thesis Theme

Adaptive Mesh Model (AMM) Structures

Charalampos Dimitrakopoulos
(mxrh 1102)

COMMITTEE

Lecturer Nikolaos Englezos (supervisor)

Professor Nikolaos Apergis

Professor Nikitas Pittis

Piraeus

February 2013

ACKNOWLEDGEMENTS

I would like to express my gratitude to my supervisor, Dr. N. Englezos for his instant guidance throughout this difficult period. I must note he has also been a wake-up call for me every time I was falling behind schedule.

Special thanks must be given to my family, friends and co-students who supported me with their positive thinking throughout this period.

ABSTRACT

Derivative securities can be priced by using theoretical valuation models or by some numerical approximation techniques. In the context of this paper the technique called Adaptive Mesh Model (AMM), introduced in Figlewski and Gao (1999), is analyzed. The main advantage of this model is its capability of reducing nonlinearity error by grafting one or more small sections of fine-high resolution lattice onto a tree with coarser time and price steps. Four AMM structures are presented, one for pricing ordinary options, one for barrier options, one for double barrier options and one for computing delta and gamma efficiently. It must also be noted that lattice models are more intuitive and flexible than theoretical models which require knowledge of more complex mathematics. Accuracy is increased and execution time of the pricing algorithm is decreased through AMM structures.

Keywords: Adaptive Mesh Model (AMM), Binomial & Trinomial Pricing Models, Ordinary Option Pricing, Barrier Option Pricing, Double Barrier Option Pricing, Greeks Pricing

Table of Contents

1. Introduction ... 1

  1.1 Historical Overview ... 1

  1.2 Basic Concepts ... 3

    1.2.1 Ordinary Option ... 3

    1.2.2 Barrier Option ... 4

    1.2.3 Greek Letters ... 4

  1.3 Option Pricing ... 5

    1.3.1 Closed-form Valuation ... 5

    1.3.2 Lattice-based Valuation ... 6

    1.3.3 Approximation Error ... 7

2. Option valuation ... 8

  2.1 Theoretical Valuation ... 8

    2.1.1 Black-Scholes Valuation Model ... 9

  2.2 Finite Difference Methods ... 11

    2.2.1 Explicit Finite Difference Method ... 11

    2.2.2 Implicit Finite Difference Method ... 14

  2.3 AMM Valuation ... 15

    2.3.1 Binomial Lattice Model ... 15

    2.3.2 Trinomial Lattice Model ... 18

    2.3.3 AMM (Trinomial Lattice-Based) ... 21

3. Barrier Option Pricing ... 26

  3.1 Single Barrier Option ... 27

    3.1.1 Trinomial Lattice Pricing Procedure ... 27

    3.1.2 Simple AMM Lattice ... 33

    3.1.3 General AMM Lattice ... 36

  3.2 Double Barrier Option ... 38

4. Gamma and Delta Calculation ... 41

  4.1 Trinomial Lattice ... 42

  4.2 AMM (Quadrinomial Lattice-Based) ... 44

    4.2.1 Simple AMM Lattice ... 45

    4.2.2 General AMM Lattice ... 47

5. Conclusion ... 51

REFERENCES 53
DIAGRAMS 56
TABLES 58
MATLAB ALGORITHMS 62

1. Introduction

1.1 Historical Overview

Since the introduction of listed options exchange in 1973 many researchers have created models for option pricing. We must consider that virtually all corporate securities can be interpreted as portfolios of puts and calls on the assets of the firm. If we consider an elementary case of a firm with a single liability of a homogeneous class of pure discount bonds, then the stockholders have a call option on the assets of the firm which they can choose to exercise at the maturity date of the debt by paying its principal to the bondholders. Consequently, the bonds can be interpreted as a portfolio containing a default-free loan with the same face value as the bonds and a short position in a put option on the assets of the firm. Therefore, methods for pricing options actually are also methods for pricing a firm. Consecutively, having the ability to accurately price the elements of a financial system we are able to make the best investment decisions at a specific moment.

Within the concept of the above idea many researchers have created pricing methods over the time. Two main approaches of option valuation, closed-form solutions and, when derivation of such solutions wasn't possible, numeric solutions were derived.

Fischer Black and Myron Scholes (1973) presented the first completely satisfactory equilibrium option pricing model. In the same year, Robert Merton (1973) extended their model in several important ways. Unfortunately, the mathematical tools employed in the Black-Scholes and Merton articles are quite advanced and have tended to obscure the underlying economics. So if the difficulty in understanding rises, we can imagine how difficult it may become to evolve or use such pricing models. Nevertheless, according to William Sharpe (1978) it is possible to derive the same results using only elementary mathematics.

Lattice models are used widely for option pricing. The binomial option pricing model first proposed by Cox, Ross and Rubinstein (1979) provides a generalized numerical method for the valuation of options. This lattice approach has been extended by Rendleman and Bartter (1979), Boyle (1986), (1988) and Hull and White (1988). Moreover, by its very construction, it gives rise to a simple and efficient numerical procedure for valuing options for which early exercise may be optimal.

Further generalizations of the binomial approach include the multinomial methods of Omberg (1988), Boyle, Envine and Gibbs (1989) , Kamrad and Ritchken (1991) and Parkinson (1977). An accelerated binomial method was proposed in Breen (1991) where a simple recursion was used to avoid redundant computations.

Another lattice model is the trinomial option pricing model. The trinomial option pricing model differs from the binomial option pricing model in one key aspect, which is incorporating another possible value in one period's time. Under the binomial option pricing model, it is assumed that the value of the underlying asset will either be greater than or less than, its current value. Therefore, the trinomial model incorporates a third possible value, a zero change in value over a time period, considering as possible for the value to be constant over a time step.

Peter Ritchken (1996) offers an approach in the context of a trinomial model by introducing a stretch parameter into the lattice, which changes the price step just enough to place nodes in the desired location. Terry H.F. Cheuk and Ton C.F. Vorst (1996) also introduce a deformation of the trinomial tree where the extra degrees of freedom in a trinomial lattice allow price nodes to be placed more or less where the analyst chooses.

In this direction of research Figlewski and Gao (1999) presented the adaptive mesh model (AMM), a model that reduces the computational effort increasing the accuracy of pricing in specific critical regions. In the same year, Ahn, and Gao (1999) presented an especially effective AMM structure for pricing options with discrete barriers. In order to understand the magnitude of the computational effort, a basic example in that paper, according to which an AMM with 60 time steps is not only ten times more accurate than a 5000 step trinomial one, but runs more than 1000 times faster.

1.2 Basic Concepts

In order to set a basis for our future understanding, we must define a financial derivative called option.

Definition 1.1: A financial derivative that represents a contract sold by one party (option writer) to another party (option holder), subject to certain conditions. The contract offers the buyer the right, but not the obligation, to buy (call) or sell (put) a security or other financial asset at an agreed-upon price (the strike price) during a certain period of time or on a specific date (exercise date).

During the 17^{TH} century option contracts were traded both in Amsterdam and in London. Ever since more complex option contracts have arisen. In this thesis we will encounter notions as ordinary option, barrier option and some of the Greek letters.

1.2.1 Ordinary Option

With the notion ordinary option we mean American or European call and put option. Their main difference is that an American option can be exercised at any given time during its life, while a European option can be exercised only at the maturity date of the option. When you are a call owner you benefit from an upward stock move and when you are a put owner you benefit from a downward stock move, as long as the underlying asset of the option is a stock.

1.2.2 Barrier Option

Barrier options are more complex than plain vanilla options. In fact, a typical barrier option pays off at expiration like an ordinary call or put, except that the payoff is contingent upon whether the underlying asset price has reached a prespecified barrier price at some earlier point during the option's lifetime.

Barrier options come in two types: in options and out options. An in barrier option, or knock-in option, pays off only if the stock finishes in the money and the barrier is crossed at some time before expiration. When the stock crosses the barrier option is knocked in and becomes a standard option of the same type (call or put) with the same strike and expiration. If the stock never crosses the barrier, the option expires worthless.

An out-barrier option, or knockout option, pays off only if the stock finishes in the money and the barrier is never crossed before expiration. As long as the stock never crosses the barrier, the out-barrier option remains a standard option of the same type (call or put) with the same strike and expiration. If the stock crosses the barrier, the option is knocked out and expires worthless. Therefore, barrier options can be up-and-out, up-and-in, down-and-out, down-and-in.

1.2.3 Greek Letters

Definition 1.2: The Greek letters are defined as the sensitivity of the option price to a single-unit change in the value of either a state variable or a parameter.

Such sensitivities can represent the different dimensions to the risk in an option. Financial institutions who sell options to their clients can manage their risk by Greek letters analysis. As a result it is crucial for these institutions to have the optimal estimation of their risk through delta and gamma calculation.

The Delta of an option, Δ, is defined as the rate of change of the option price with respect to the rate of change of the underlying asset price:

$$\Delta = \frac{\partial \Pi}{\partial S}, \qquad (1.1)$$

where Π is the option price and S is underlying asset price. The Gamma of an option, Γ, is defined as the rate of change of Delta with respect to the rate of change of the underlying asset price:

$$\Gamma = \frac{\partial \Delta}{\partial S} = \frac{\partial^2 \Pi}{\partial S^2}, \qquad (1.2)$$

where Π is the option price and S the underlying asset price.

1.3 Option Pricing

1.3.1 Closed-form Valuation

In order for an investor to take the right decisions about his portfolio, he must know the price of each option he holds. Most of derivative securities need to be priced by numerical techniques, since only a small subset of these securities can be priced through closed-form valuation equations.

Definition 1.3: In mathematics, an expression is called a closed-form expression if it can be expressed analytically in terms of a finite number of well-known functions.

One of the most common closed-form valuation equations are derived in the Black-Scholes (1973) (BS) model. For example the (BS) closed-form valuation equation of the European call option for a non-dividend paying underlying stock is given below:

$$C(S,t) = N(d_1)S - N(d_2)Ke^{-r(T-t)}, \qquad (1.3)$$

$$d_1 = \frac{\ln\left(\frac{S}{K}\right) + \left(r + \frac{\sigma^2}{2}\right)(T-t)}{\sigma\sqrt{T-t}}, \qquad (1.4)$$

$$d_2 = \frac{\ln\left(\frac{S}{K}\right) + \left(r - \frac{\sigma^2}{2}\right)(T-t)}{\sigma\sqrt{T-t}} = d_1 - \sigma\sqrt{T-t}\,; \quad (1.5)$$

here S is the price of the stock, C(S,t) the price of a European call option, K the strike price of the option, r the annualized risk-free interest rate, continuously compounded, σ the volatility of the stock's returns, t the time in years and N(x) the standard normal cumulative distribution function given by the following formula:

$$N(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{z^2}{2}} dz. \quad (1.6)$$

1.3.2 Lattice-based Valuation

However, the securities in their majority are priced using appropriate numerical approximation techniques, such as Binomial and Trinomial lattice models. These models are widely used because they are intuitive and very flexible. We must take into consideration that as the lattice becomes finer, these methods converge to the theoretical option values that would be produced by a continuous-time, continuous-state model such as Black-Scholes. Convergence of the binomial method for pricing American options is proved in Amin and Khanna (1994).

A serious disadvantage of these models is that they require a very large number of calculations to achieve acceptable accuracy. Particularly in some critical regions where the option is highly non-linear, i.e. around the strike price at expiration or near the knock-out price for a barrier option, the computational effort required for a specific level of accuracy increases very rapidly if we attempt to reduce the step size. In addition to that, most of this computational effort is wasted on unimportant regions.

A solution to the above problem is the Adaptive Mesh Model (AMM) presented by Figlewski and Gao (1999), which is a very flexible approach that increases considerably the efficiency in trinomial lattices. Coarse time and price steps are used in most of the tree, but small sections of finer mesh are constructed to improve resolution in specific critical areas. A relatively coarse grid that is fast to calculate is used for most of the lattice, but a small section of fine mesh is constructed where greater accuracy really matters.

1.3.3 Approximation Error

Approximation error in lattice models can be thought of as arising from two different sources, distribution error and nonlinearity error. Distribution error occurs because, throughout the tree, the model attempts to approximate a continuous lognormal distribution with a discrete binomial or trinomial distribution. As a result no matter how fine we make the lattice we will not have enough price or time steps to cover the initial distribution.

Nonlinearity error arises because the option value is nonlinear in the underlying asset price, especially in critical regions, in a way that cannot be captured accurately by the discrete lattice. The Adaptive Mesh Model (AMM) sharply reduces nonlinearity error by grafting one or more small sections of fine high-resolution lattice onto a tree with coarser time and price steps.

2. Option valuation

2.1 Theoretical Valuation

The Black-Scholes (1973) (BS) option valuation model was one of the first that provided closed-form equations for pricing European call and put options based on observable parameters. Of equal importance, the no-arbitrage principle used to obtain the BS equation pointed the way towards theoretical valuation models for all types of contingent claims.

Prior to (BS) several researchers have approached the valuation of options expressed in terms of warrants. Sprenkle (1961), Ayres (1963), Boness (1964), Samuelson (1965), Baumol, Malkiel and Quandt (1966), and Chen (1970) have produced such incomplete valuation formulas due to the use of one or more arbitrary parameters. The use of the no-arbitrage principle in the valuation procedure used by BS is the key difference between their predecessors' models and BS. However, American options and other contracts with early exercise present a serious problem because, although the no-arbitrage principle still holds, usable closed-form valuation formulas seldom exist.

2.1.1 Black-Scholes Valuation Model

Definition 2.1: According to Black-Scholes (1973) the following assumptions are made in order to value an option in terms of the price of the stock.

(1) The short-term interest rate is known and is constant through time.
(2) The stock price follows a random walk in continuous time with a variance rate proportional to the square of the stock price. Thus the distribution of possible stock prices at the end of any finite interval is lognormal. The variance rate of the return on the stock is constant.
(3) The stock pays no dividends.
(4) The option is called European if it can only be exercised at maturity.
(5) There are no transaction costs in buying or selling the stock or the option.
(6) It is possible to borrow any fraction of the price of a security in order to buy it or hold it, at the short-term interest rate.
(7) There are no penalties to short selling. A seller who does not own a security will simply accept the price of the security from a buyer, and will agree to settle with the buyer on some future date by paying him an amount equal to the price of the security on that date.

Consequently, these assumptions allow us to create a hedged position consisting of a long position in the stock and a short position in the option, whose value will not depend on the price of the stock. The Black-Scholes methodology begins with the assumption that the underlying asset follows the logarithmic diffusion

$$\frac{dS}{S} = \mu\, dt + \sigma\, dz, (2.1)$$

where dS denotes the change in the asset price S over the infinitesimal time interval dt, μ and σ are the instantaneous mean and volatility, and dz represents a standard Brownian motion.

Definition 2.2: A standard Brownian motion (also known as standard Wiener process) is a stochastic process $\{W_t\}_{t\geq 0}$ (that is, a family of random variables W_t, indexed by nonnegative real numbers t, defined on a common probability space $(\Omega, \mathcal{F}, P)$) with the following properties:

(1) $W_0 = 0$.

(2) With probability 1, the mapping $t \longmapsto W_t$ is continuous in t.

(3) The process $\{W_t\}_{t\geq 0}$ has stationary, independent increments.

(4) The increment $W_{t+s} - W_s$ has the $N(0,t)$ distribution.

Here the term of independent increments means that for every choice of nonnegative real numbers

$$0 \leq s_1 < t_1 \leq s_2 < t_2 \leq \cdots \leq s_n < t_n < \infty$$

the increment random variables

$$W_{t_1} - W_{s_1}, W_{t_2} - W_{s_2}, \cdots, W_{t_n} - W_{s_n}$$

are jointly independent, and the term of stationary increments means that for any

$0 < s, t < \infty$, the distribution of the increment $W_{t+s} - W_s$ has the same distribution with $W_t - W_0 = W_t$.

It is not obvious that properties (1)-(4) in the definition of standard Brownian motion are mutually consistent, so it is not a priori clear that a standard Brownian motion exists. (The main issue is to show that properties (3)–(4) do not preclude the possibility of continuous paths.) That it does exist was first proved by Norbert Wiener in about 1920. His proof was simplified by Paul Lévy.

By assuming continuous trading, we can buy or sell any amount of stock even if the latter is not a round number. This kind of transaction is impossible in the real markets because obviously an investor can buy only round numbers of shares. Another assumption is that there are no transaction costs. With these two assumptions, an investor can follow a self-financing dynamic trading strategy to replicate a derivative security's future payoff exactly. Thus, to avoid profitable arbitrage the option value must equal the cost of the replicating portfolio. This leads to the fundamental partial differential equation (PDE) of contingent claims pricing:

$$\frac{\partial f}{\partial t} + \frac{\partial f}{\partial S} rS + \frac{1}{2} S^2 \sigma^2 \frac{\partial^2 f}{\partial S^2} = rf, \qquad (2.2)$$

where r is the risk-free interest rate, f is the option price, S the underlying asset and $\sigma^2$is the variance rate of the return of the stock.

Definition 2.3: In mathematics, a PDE is a differential equation that contains unknown multivariable functions and their partial derivatives.

In certain cases, such as pricing of European put options on non-dividend paying stocks, the PDE can be solved to give a closed-form valuation formula. Otherwise, an approximate option value can be obtained using finite difference in their solution techniques, as Michael J. Brennan and Eduardo S. Schwartz (1977) demonstrate in their investigation of American put pricing. The latter approach has been extended by Courtadon (1982b).

2.2 Finite Difference Methods

Definition 2.4: A finite difference is a mathematical expression of the form

$$P(x+b) - P(x+a). \quad (2.3)$$

Three forms of finite difference are commonly considered: forward, backward and central differences. In particular we have forward difference for $b = h, a = 0$, backward difference for $b = 0, a = -h$ and central difference for $b = \frac{1}{2}h,\ a = -\frac{1}{2}h$.

2.2.1 Explicit Finite Difference Method

The explicit finite difference technique for solving the PDE of (2.2) is equivalent to a trinomial tree procedure, though Hull and White (1990) find lattice methods more intuitive. However, according to Geske and Shastri (1985) the explicit finite difference method, with logarithmic transformations, is the most efficient approach when large numbers of stock options are being evaluated. The explicit difference technique transcends the implicit one because it is conceptually simpler, being a simple application of the trinomial lattice approach. The main disadvantage, though, is the method's lack of convergence to the solution of the differential equation as Δt tends to zero.

We next give a description of the explicit finite difference method. The stochastic process that the underlying asset S follows is

$$dS = \mu(S,t)\ dt + \sigma(S,t)\ \ dz, \qquad (2.4)$$

where dz is a Wiener process, μ and σ are the instantaneous proportional drift rate and volatility of S. We must note the similarity to formula (2.1). If λ is the market price of risk of S, then, as shown in Garman (1976) and in Cox, Ingersoll and Ross (1985a), the price of the derivative security f, which depends on a single stochastic variable S, must satisfy the following equation:

$$\frac{\partial f}{\partial t} + \frac{\partial f}{\partial S}(\mu - \lambda\sigma)S + \frac{1}{2}S^2\sigma^2\frac{\partial^2 f}{\partial S^2} = rf, \qquad (2.5)$$

where r is the risk-free interest rate. Both r and λ may be functions of S and t. When S is the price of a non-dividend paying stock, then $\mu - \lambda\sigma = r$ and (2.5) reduces to the Black Scholes (1973) PDE (2.2).

The next step of the explicit finite difference method is to construct a grid which will approximate the terms of equation (2.6). To implement the explicit finite difference method, a small time interval Δt, and a small change in S, ΔS, are chosen. A grid is then constructed for considering values of f when S is equal to:

$$S_0, S_0 + \Delta S, S_0 + 2\Delta S, \cdots, S_{max}$$

and time is equal to

$$t_0, t_0 + \Delta t, t_0 + 2\Delta t, \cdots, T,$$

where the parameters S_0 and S_{max} are the smallest and largest values of S respectively, considered by the model, t_0 is the current time, and T is the end of the life of the derivative security also known as maturity date.

Denoting $t_0 + i\Delta t$ by t_i, $S_0 + j\Delta S$ by S_j and the option value at the (i,j) point on the grid by $f_{i,j}$, the partial derivatives of f with respect to S at the node $(i-1,j)$ are approximated as:

$$\frac{\partial f}{\partial S} = \frac{f_{i,j+1} - f_{i,j-1}}{2\Delta S}, \qquad (2.6)$$

$$\frac{\partial^2 f}{\partial S^2} = \frac{f_{i,j+1} + f_{i,j-1} - 2f_{i,j}}{2\Delta S}, \qquad (2.7)$$

and the partial derivative with respect to t is approximated as

$$\frac{\partial f}{\partial t} = \frac{f_{i,j} - f_{i-1,j}}{2\Delta S}. \qquad (2.8)$$

We must note in this point that the equations for the implicit finite difference method are obtained in a similar way with the approximation of the latter partial derivative as follows

$$\frac{\partial f}{\partial t} = \frac{f_{i+1,j} - f_{i,j}}{2\Delta S}. \qquad (2.9)$$

Substituting (2.6)-(2.8) into (2.5) gives

$$f_{i-1,j} = a_{j-1} f_{i,j-1} + a_j f_{i,j} + a_{j+1} f_{i,j+1}, \qquad (2.10)$$

where

$$a_{j-1} = \frac{1}{1+r\Delta t}\left[-\frac{(\mu-\lambda\sigma)S_j\Delta t}{2\Delta S} + \frac{1}{2}\frac{S_j^2\sigma^2\Delta t}{\Delta S^2}\right],$$

$$a_j = \frac{1}{1+r\Delta t}\left[1 - \frac{S_j^2\sigma^2\Delta t}{\Delta S^2}\right], \text{ and}$$

$$a_{j+1} = \frac{1}{1+r\Delta t}\left[\frac{(\mu-\lambda\sigma)S_j\Delta t}{2\Delta S} + \frac{1}{2}\frac{S_j^2\sigma^2\Delta t}{\Delta S^2}\right].$$

With these substitutions we relate the value $f_{i-1,j}$ of the derivative security at time t_{i-1} to three alternative values of the derivative security at time t_i. If we define

$$p_{j,j-1} = \left[-\frac{(\mu-\lambda\sigma)S_j\Delta t}{2\Delta S} + \frac{1}{2}\frac{S_j^2\sigma^2\Delta t}{\Delta S^2}\right],$$

$$p_{j,j} = \left[1 - \frac{S_j^2\sigma^2\Delta t}{\Delta S^2}\right], \text{ and}$$

$$p_{j,j+1} = \left[\frac{(\mu-\lambda\sigma)S_j\Delta t}{2\Delta S} + \frac{1}{2}\frac{S_j^2\sigma^2\Delta t}{\Delta S^2}\right],$$

equation (2.10) becomes

$$f_{i-1,j} = \frac{1}{1 + r\Delta t}\left[p_{j,j-1}f_{i,j-1} + p_{j,j}f_{i,j} + p_{j,j+1}f_{i,j+1}\right]. \quad (2.11)$$

Apparently the explicit finite difference method is equivalent to a trinomial lattice approach if we interpret $p_{j,j-1}$, $p_{j,j}$ and $p_{j,j+1}$ as the probabilities of moving from S_j to S_{j-1}, S_j and S_{j+1} respectively. We must note at this point that these probabilities add to unity and give a drift rate of $(\mu - \lambda\sigma)S$. A variance rate of $S^2\sigma^2$ is implied if terms of $0(\Delta t^2)$ are ignored. Using this method we value f at time t_i as its expected value at time t_{i+1}, in a world where the drift rate of S is $(\mu - \lambda\sigma)$, discounted at the risk-free rate of interest.

2.2.2 Implicit Finite Difference Method

As mentioned before, the implicit finite difference method differs from the explicit one at the approximation technique for the first partial derivative of f with respect to t. Moreover, the implicit method requires the specification of several boundary conditions for the derivative security as $S \to 0$ and $S \to \infty$ while the explicit method does not require such conditions.

In practice, it is efficient to use $ln(S)$ rather than S as the underlying variable when finite difference methods are applied. When σ is constant, the instantaneous standard deviation of $ln(S)$ remains constant and independent of variables S, t. According to Hull and White (1990, p.90-91) convergence of these two finite difference methods is ensured by the use of $ln(S)$ rather than S and it is proved by Hull and White with the use of a theorem in Ames (1977, p.45).

2.3 AMM Valuation

In this section two lattice models will be analyzed, the binomial and trinomial lattice models. Although both models are very intuitive, the trinomial model will be the base for our AMM structure analysis. The main reason of this practice is that the binomial model has little flexibility to deal with more complex option problems. However, it is analyzed below in order to understand the evolution from binomial to trinomial models and finally understand AMM superiority of its predecessors.

2.3.1 Binomial Lattice Model

We assume that the price of the underlying asset follows a multiplicative binomial process over discrete periods. The rate of return on the underlying asset over each period can have two possible values, either $u-1$ or $d-1$, with probabilities q and 1-q, respectively, with u for up and d for down movement of the underlying asset.

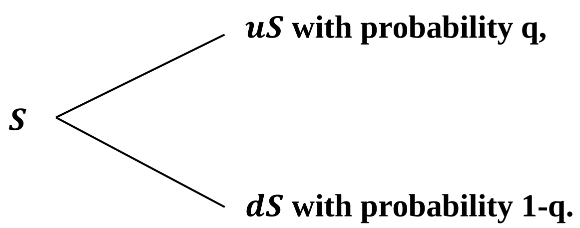
Definition 2.6: The rate of return R on a stock is calculated by the following formula:

$$R = \frac{S_{t+\Delta t} - S_t}{\Delta t}\ , \qquad (2.12)$$

where $S_{t+\Delta t}$ and S_t are the stock price at time $t+\Delta t$ and t, respectively.

We can now easily understand that for $\Delta t = 1$, i.e. in a single time step, if the current stock price is S, then the stock price will have two possible outcomes uS or dS as illustrated by Figure 2.1 below.

Figure 2.1:



Several further assumptions must be made. Firstly, the interest rate is assumed to be constant in order to lend as much money as an individual wishes at the specific rate. It is also assumed that there are no taxes, transaction costs or even any margin requirements, while short-selling of the stock is not forbidden. As long as the following inequality $u > r > d$ stands, no profitable riskless arbitrage opportunities are possible.

The most simplified approach of binomial model is a lattice of one step. Valuing a call option an a stock with the expiration date one period away we have a one-step lattice as illustrated by Figure 2.2 below.

Figure 2.2:

$\boldsymbol{C}$

$\boldsymbol{C_u = max[0, uS - K]}$ **with probability q,**

$\boldsymbol{C_d = max[0, dS - K]}$ **with probability 1-q.**

Here C is the current value of the call, $\boldsymbol{C_u}$ and $\boldsymbol{C_d}$ are the values of the option at the end of the period if the price of the stock goes up or down respectively, S is the value of the stock, and K the strike price.

In order to replicate a portfolio of equal payoff to the option's payoff we suppose that Δ shares of stock are required and B amount of stock's currency is invested in riskless bonds. The cost of this replication strategy will be $\Delta S + B$ while at the end of the period the value of this portfolio will be given by Figure 2.3.

Figure 2.3:

$\boldsymbol{\Delta S + B}$

$\boldsymbol{\Delta uS + (1 + r)B}$ **with probability q,**

$\boldsymbol{\Delta dS + (1 + r)B}$ **with probability 1-q.**

In order to achieve the equality of option's and replication portfolio's payoff we conclude to the following equations:

$$\Delta uS + B = C_u\,, \quad (2.13)$$

$$\Delta dS + B = C_d\,. \quad (2.14)$$

Subsequently solving these equations for Δ and B we derive:

$$\Delta = \frac{C_u - C_d}{(u-d)S}, \qquad (2.15)$$

$$B = \frac{uC_d - dC_u}{(u-d)(1+r)}. \quad (2.16)$$

Concluding, the value of the call option, C, cannot be less or greater than the current value of the replication portfolio $\Delta S + B$ because otherwise there would be a riskless profit with no net investment. We must note that if the value of the call option is greater than the value of the replication portfolio then it must be assumed that the investor who bought the call option we sold will not exercise immediately although he can. Therefore from equations (2.15) and (2.16) we derive that:

$$C = \Delta S + B$$

$$= \frac{C_u - C_d}{(u-d)} + \frac{uC_d - dC_u}{(u-d)(1+r)}$$

$$= \frac{\left[\left(\frac{r-d}{u-d}\right)C_u + \left(\frac{u-r}{u-d}\right)C_d\right]}{(1+r)} \; or$$

$$C = \frac{[pC_u + (1-p)C_d]}{(1+r)}. \quad (2.17)$$

Extending the binomial tree with more periods we will have a recursive procedure for finding the value of the option starting at the expiration date and working backwards. The generalized binomial valuation formula for a call option is:

$$C = \frac{\sum_{j=0}^{n}\left(\frac{n!}{j!\,(n-j)!}\right)p^{j}(1-p)^{n-j}max\left[0, u^{j}d^{n-j}S-K\right]}{(1+r)^{n}}. \quad (2.18)$$

Moreover in recombining trees we notice that the nodes at step n would have about the same prices with the nodes at step n+2 creating a peculiar "even-odd" property to convergence. So for example if we assume a call option with stock price 50, strike price 50, $r = 0.1$, $\sigma = 0.4$, time to maturity T=5/12 and a total of 50 steps then the non-monotonic convergence is exhibited in Diagram 2.1. The trinomial model also exhibits non-monotonic convergence, but not of such a striking form.

2.3.2 Trinomial Lattice Model

Although the binomial model is very intuitive, it has become outdated by the trinomial model which deals with more complex option problems. Due to trinomial model's excess in degrees of freedom in comparison to binomial model, the former is more useful and adaptable for many derivative applications.

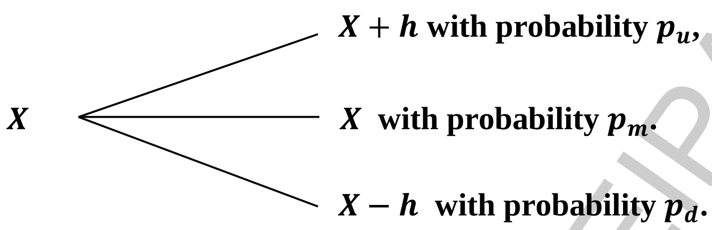
As mentioned earlier, the asset price is assumed to be lognormal, fact that practically means that the tree is based on the logarithm of S. If we define $X \equiv \ln(S)$ X is implied to be normally distributed. Furthermore under the risk neutrality assumption, X follows the process,

$$dX(t) = \left(r - q - \frac{\sigma^2}{2}\right)dt + \sigma\, dz, \qquad (2.19)$$

similar to the logarithmic diffusion followed by BS model in equation (2.1), where q denotes the instantaneous rate of dividend payout.

Constructing the trinomial tree, the underlying asset price is allowed to move to one of three values, designated as up (u), down (d) and middle (m). The risk neutral probabilities set, p_u, p_d and p_m, is associated with these branches providing greater rate of convergence when the tree is symmetrically constructed for some applications according to Figlewski and Gao (1999). Let the middle node to remain constant and the up and down moves to be of equal magnitude. By k we denote the length of time step and h the size of an up or down move. Moreover, in every step X goes to $X + h$ with probability p_u, to $X - h$ with probability p_d, and remains constant with probability p_m, as illustrated by Figure 2.4.

Figure 2.4:



Apparently the value of the time step $\boldsymbol{k}$ is determined by the formula $\boldsymbol{k} = \frac{\boldsymbol{T}}{\boldsymbol{N}}$, partitioning the total time to maturity to equal time steps.

Several constraints must be met for the model to be accurate. At every period the expected log price change must be proportional to the size of the step k multiplied by the coefficient of $\boldsymbol{dt}$ in equation (2.19) so that the change of $\boldsymbol{X}(\boldsymbol{t})$ becomes zero. Furthermore the standard deviation must be consistent with the known volatility of the underlying asset. Also the probabilities from one node to another must sum to unity.

In order to reduce the convergence error of the trinomial tree, the moments of the normal distribution must be matched with the expected values of the tree. As discussed in Cho and Lee (1995), moments beyond mean and variance must be used. The fourth moment, kurtosis, is selected over the third moment, skewness, due to the fact that all odd numbered moments of the trinomial model will be zero as a result of our symmetrical distribution selection. Consequently, we derive to the following system of equations:

$$1 = p_u + p_m + p_d \,,$$

$$E[X(t + k) - X(t)] = 0 = p_u h + p_m 0 + p_d(-h) \,,$$

$$E\left[\left(X(t+k)-X(t)\right)^2\right] = \sigma^2 k = p_u h^2 + p_m 0 + p_d (-h)^2 \,,$$

$$E\left[\left(X(t+k)-X(t)\right)^4\right] = 3\sigma^4 k^2 = p_u h^4 + p_m 0 + p_d (-h)^4. \quad (2.20)$$

where the four unknowns are the probabilities p_u, p_m, p_d and the size h of the up or down movement of X. Solving these equations for the unknown variables we conclude:

$$p_u = \frac{1}{6}, \qquad p_m = \frac{2}{3}, \qquad p_d = \frac{1}{6}, \qquad h = \sigma\sqrt{3k}. \qquad (2.21)$$

In order to determine the option price, given the asset price contingent payoffs at maturity and discounting backwards through the tree, we can derive the following formula that calculates the option price for a single node at date t and price X:

$$C(X,t) = e^{-rk}[p_u(h,k)C(X+h,t+k) + p_m(h,k)C(X,t+k) + p_d(h,k)C(X-h,t+k)]. \quad (2.22)$$

The latter equation is very similar with that of the binomial model at equation 2.17. Apparently the next step in this method is to create a recursive procedure for more than one time steps. We must note that in the current case the values of h and k are fixed, resulting to fixed probabilities. Otherwise, these probabilities could vary depending on the changes of h and at every step.

Therefore, using the model in practice presents us nonlinearity error when the true option value does not change proportionally as the asset price changes between a pair of nodes. As mentioned in the introduction gamma is defined as the rate of change of delta respected to the rate of change of the underlying asset price, while delta is the rate of change of the option price respected to the rate of change of underlying asset. Hence, we can easily understand that nonlinearity error will create a large gamma value, in absolute terms, in the current price terms. The nonlinearity error is greatest around the strike price at expiration for a European option, due to the form of its payoff function, while for an American option this type of error affects both the strike price at expiration and the prices that surround the strike price.

A solution proposed by Broadie and Detemple (1996) is a simple substitution of the prices at every node around expiration with the Black-Scholes value for the option price. Therefore, moving backwards on the tree using equation (2.22) we will derive the option price with greater accuracy diminishing the approximation error. At this point we must note that for these specific nodes where the substitutions take place, the option is considered as European because, practically, there is not enough time for early exercise. It must also be noted that a closed form solution for the option we are valuing should exist, making this technique difficult to adjust to more complex derivatives.

A typical procedure for reducing the approximation error in a lattice model is to decrease the length of the time steps. The disadvantage of this procedure, though, is a rapid increase in the number of nodes and consequently a massive increase on the computational effort required for the price calculation of each node. In order to quantify this negative effect we designate the initial node as $n = 0$. The binomial model for instance had $N + 1$ nodes at time step $n = N$ and a total of $[(N + 1)^2 + (N + 1)]/2 = (N^2 + 3N + 2)/2$ nodes. Respectively the trinomial model has $2n + 1$ nodes at time step $n = N$ and a total of $(N + 1)^2$ nodes. For example if h is proportional to $\sqrt{k}$ as shown in equation (2.21), then to decrease the size of the price step in half requires a tree with quadruple time steps and approximately sixteen-fold nodes.

The question that arises is if this many nodes are actually needed to achieve a certain accuracy level. More time steps are obligatory near some critical areas, such as the region of the strike price at expiration, but in other not so critical areas the large number of nodes is practically a computational waste. Finally, a hybrid tree (AMM) with a high resolution fine lattice in the region of the strike price at expiration and a coarse and fast lattice everywhere else was introduced by Figlewski and Gao (1999).

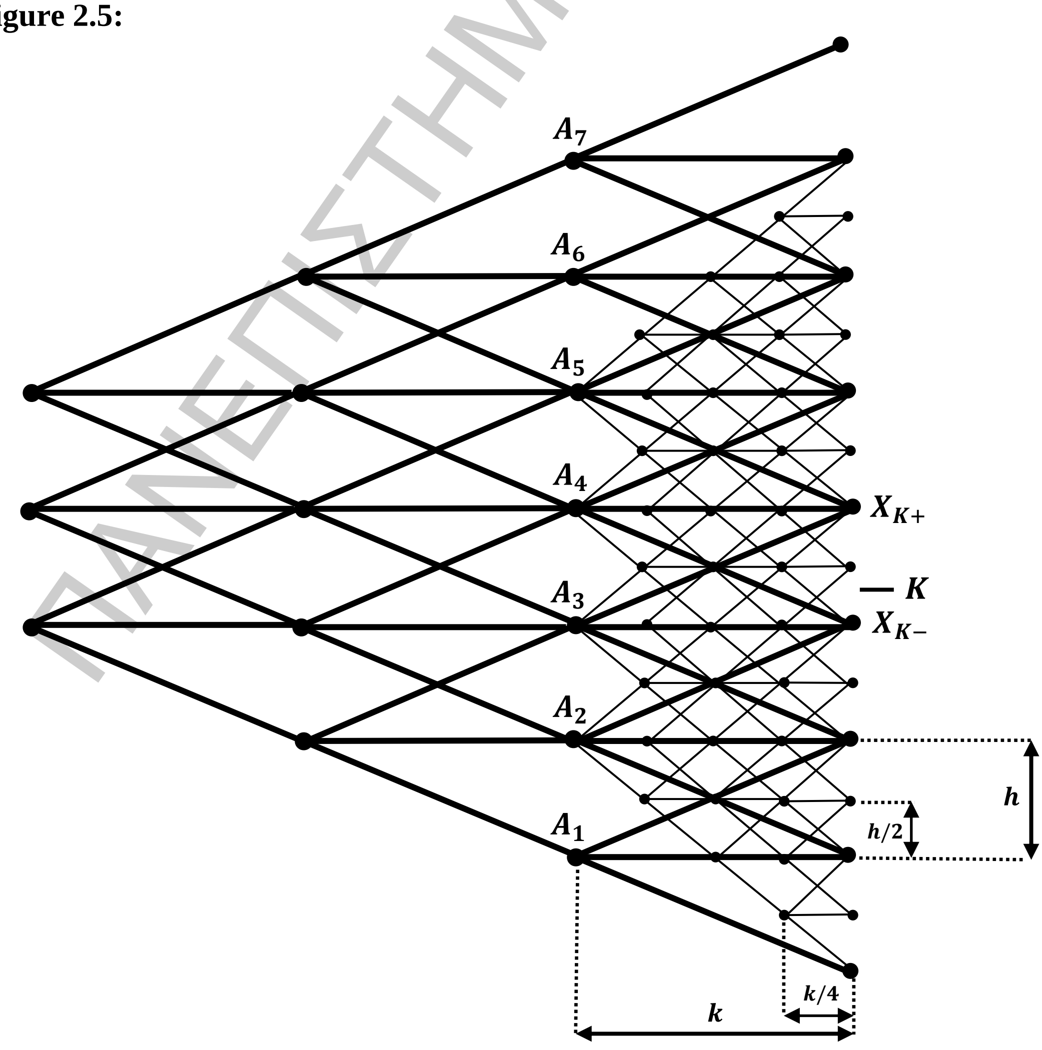
2.3.3 AMM (Trinomial Lattice-Based)

In order to create the AMM structure for an American put option we must adjust the finer mesh tree onto the general coarser tree. Shaping the AMM structure for a European put option pricing, as illustrated by Figure 2.5, will equip us with a guide for the generalization to the American put option pricing structure.

In Figure 2.5 the most crucial section of the AMM structure is illustrated; that is the section of the pricing lattice in the immediate vicinity of the strike price in the last few periods before expiration. As far as the coarse lattice is concerned, this is a lattice with price and time steps h and k respectively and is illustrated by heavy lines. Moreover, the fine mesh, with price and time steps $h/2$ and $k/4$ respectively, is illustrated by light lines covering all $T-k$ coarse nodes from which there are both fine mesh paths that end up in the money and fine mesh paths that end up out of the money.

We must note at this point that K is the strike price, while X_{K-} and X_{K+} are the two date T coarse mesh asset prices that bracket the strike price. Obviously the nodes of the fine mesh lattice at date T equip as with the desirable accuracy in the context that larger number of nodes are more possible to approximate the real price of the asset price.

Figure 2.5:



The objective of this structure is to build a fine mesh tree around the strike price at expiration and merge it with the coarse lattice so that the valuation information is transmitted properly. As mentioned before to decrease the price step to half, i.e. $h/2$, the time steps should be decreased to $k/4$. Another advantage of this particular set of finer mesh is that it guarantees that its nodes will overlap the coarse mesh nodes one step before expiration. The overlapped nodes, namely A_2 to A_5, will be calculated by the fine lattice, while the remaining nodes of the coarse lattice at this time step will be computed by the standard procedure of the terminal payoffs in the coarse lattice. This tactic ensures that the more accurate fine mesh values for the critical nodes will be fully exploited into the coarse lattice. The valuation of the asset price from that time step to the start of the coarse lattice proceeds by rolling back by the coarse lattice to the initial date.

Obviously, the fine mesh is added for dates in the period $T - k$ to T, only in the region around the strike price. Only the nodes of the coarse lattice from which start both fine mesh paths that end in the money and fine mesh paths that end up out of the money are covered by the fine mesh lattice. Apparently, a cover of more nodes would result to computational waste for the AMM structure. For example, it is unnecessary to calculate the value of A_6 through a fine mesh lattice at time step $T - k$ because every possible fine mesh path beginning at that point would end out of the money, i.e., at a node where the option payoff is linear and there is no nonlinearity error to correct. On the other hand, from node A_5, while the coarse lattice paths end up out of the money, a fine mesh path with only down moves between $T - k$ and T would end up below K and actually in the money. So as far as around K the option payoff is nonlinear, the fine mesh value for node A_5 will be more accurate than what the coarse mesh would have produced.

Reducing in half the price step h, each node at a time step t will end up in a radius of nodes of $4h$ at time step $t + k$ than $2h$ that was the radius of nodes at time step t for the coarse lattice. For instance, in Figure 2.5 the number of nodes to be calculated in the area between the time steps $T - k$ and T increased from 12 in the coarse lattice to 52 in the finer lattice. If we do not take into consideration the path that comes through A_7, simply by going always up, we actually have a trinomial lattice tree of a total of 20 nodes. Furthermore, if we employ an adaptive mesh the computational effort will be less than just adding one time step in the coarse tree. For larger number of nodes the use of an adaptive mesh would increase the number of calculations by less than 0.4% for a 100 step trinomial tree.

Another advantage of the AMM structure is that it is isomorphic at successive levels of refinement, i.e. once an adaptive mesh tree is constructed it is simple to construct another finer adaptive mesh lattice that could be added on the existing tree. For example, in Figure 2.5 an extra adaptive mesh of price steps of size $h/4$ could be added at time period $T-k/4$ to T, while the total cost of this further refinement is just 40 additional node calculations.

In Table 2.1 the comparison between Binomial and Trinomial models' performance versus the AMM performance, whether there is only one layer of fine mesh or two, is illustrated. These AMM are added around the strike price at expiration. In this comparison theoretical values are computed for a test set of 27 European put options. One may wonder why American put options are not used in this example. The reason for this tactic is that an exact benchmark can compute European options and compare it with our approximations. In such a way we can estimate the accuracy of our model. In practice there is no real need to estimate the value of European puts because we already know a closed form formula for their value. As mentioned before, though, American put options act like European put ones near their expiration because there is no possibility to exercise.

For the models in Table 2.1, the initial asset price is set to 40, the riskless interest rate is 5% which actually corresponds to a 4.88% continuous rate, and we assume that no dividends are paid. There are 27 combinations of European put options, simply by combining every element of the following triads of information: we assume three strike prices of 35, 40 and 45; three different maturity dates of 1, 4 and 7 months; and three different possible volatilities of 0.20, 0.30 and 0.40. So actually in Table 2.1 we can see a column of root mean squared errors (RMSE) for the computed values through Binomial, Trinomial and AMM with one or two layers of fine mesh compared to the exact Black and Scholes valuation. We can also see the number of nodes needed for the computation of each model and use it as a comparison for the time needed for the completion of calculations of each model. Moreover, lattices with 25, 100, 250 and 1000 time steps are illustrated.

As one can see in these results the accuracy of the price calculation is significantly improved through the use of AMM, subject to quite small change in the number of nodes and subsequently in the execution time. For example the model with the double fine mesh attached of only 25 time steps is much more accurate than a standard Trinomial of 250 time steps and only a little less accurate than a 1000 step Binomial that requires about 250 times more calculation time.

If we compare only the Trinomial-based models, we can see that they have almost the same number of nodes and as a result about the same execution time. Obviously in comparison with the Binomial, the Trinomial-based models are slower but far more accurate. As far as accuracy is concerned, the AMM model with one layer of fine mesh is about four times as accurate as the standard Trinomial, while the AMM model with the two layers of fine mesh is about four times as accurate as the AMM with one layer of fine mesh.

This particular AMM approach has little effect on delta and gamma calculations which will be analyzed further in Chapter 4. At this point we must note that delta and gamma are not computed as numerical derivatives by simply perturbing the starting asset price. As it will be explained in Chapter 4, that procedure is inaccurate and is greatly affected by the nonlinearity error.

A solution was provided by Pelsser and Vorst (1994), where an extension of the tree, by one period prior to the initial date, is required for the subsequent calculation of the Greek letters from the node values within the extended tree. The calculations then require differencing option values at asset prices exactly one price step apart, minimizing accordingly the effect of nonlinearity error since nonlinearity error at expiration affects both option prices similarly. More extensively in Chapter 4 another adaptive mesh structure will be applied to the initial nodes of the tree in order to improve the estimates of delta and gamma risk exposures.

3. Barrier Option Pricing

In the AMM structure that was presented in the previous chapter accuracy was greatly improved with very little computational cost by adding a fine mesh in a critical region of the asset price valuation trinomial tree. In the present chapter we will show how an adaptive mesh extension of the prior structure can be used for pricing barrier options or other derivatives whose values depends on whether the asset price reaches a specified level at any point during the option's life. Simple and double barrier option valuation will be studied.

As mentioned in more exotic derivatives numerical approximation is usually the only method that applies to the valuation of the asset price. Closed form solutions rarely exist for even plain vanilla European options let alone for more complex derivatives. The concept of this project is to enhance a simple structure of a trinomial lattice with an adaptive mesh procedure in order to achieve the wanted accuracy with much less computational effort, especially for procedures like calculating implied volatilities which involve intense computations.

3.1 Single Barrier Option

The case of a barrier option pricing when the initial asset price is close to the barrier is the standard problem at which we will apply the adaptive mesh structure. By definition a barrier option is contingent upon whether the underlying asset price has reached the specified barrier price at some earlier point during the option's lifetime. Otherwise, the latter pays off at expiration like an ordinary call or put. A down and out call option with strike price K and barrier price H, for example, has the same payoff at expiration as a European call, if the stock price remains higher than H throughout its entire life. In an opposite case, if the price falls even once below H the option is knocked out and expires worthless, regardless of the underlying asset price at expiration. We must note that the barrier price H of the example is also called out-strike.

Barrier options have become widespread, especially for foreign currency contracts. Derivatives that make use of barriers in their pricing structure were described by Rubinstein (1991) and briefly explained by Gastineau and Kritzman (1996), namely capped options, ladder options and interest rate corridors.

3.1.1 Trinomial Lattice Pricing Procedure

In this type of options nonlinearity error creates obstacles in the valuation procedure. Specifically in barrier options the nature of the derivative changes according to the price barrier, so we can understand that the option price is directly correlated with the magnitude of the price steps. In other words, the discreteness of the price on the tree affects directly the valuation procedure.

As demonstrated in Diagram 2.1 there was an even-odd behavior of the convergence of the binomial model for regular options to the Black and Scholes one. Likewise, for the trinomial model the convergence follows a slightly different pattern which has a similar effect that will be demonstrated subsequently.

A European down and out call option will be used to illustrate the valuation problems of barrier options, due to the existence of a closed form equation valuation formula that can serve as a benchmark for the accuracy of the evaluation method. At this point it must be noted that this closed form was derived by Merton (1973) and is illustrated below:

$$C_{DO}(S,K,T,r,\sigma,H) = C_{BS}(S,K,T,r,\sigma) - \left(\frac{H}{S}\right)^{2\left(r-\frac{\sigma^2}{2}\right)} C_{BS}\left(\frac{H^2}{S},K,T,r,\sigma\right), (3.1)$$

where C_{DO} denotes the down and out call value and C_{BS} denotes the Black and Scholes call formula.

In comparison with the valuation procedure of Subsection 2.3.2, the corresponding trinomial valuation procedure uses the same variables as before with one addition, that of H which is used as the option's knock out barrier provided that is below the price S. Useful information for the knock out barrier price is that it remains constant throughout the derivative's lifetime. The key part for the valuation procedure is to have each layer of nodes fall at the same price in every time step. In order to achieve this we must built the trinomial tree around the initial log stock price without adjusting for the mean, therefore, $X(t) = \ln(S_t)$ for all t. Due to this modification, however, we are not able to set the kurtosis of the tree distribution equal to the risk neutral distribution, but, we can still match the mean and variance with the risk neutral distribution. So, our main difficulty in the present moment is to find a value for h.

Similar to the trinomial procedure for a plain vanilla option, we try to form here a system of equations for the trinomial tree using equations (2.20) as a guideline. As we mentioned earlier we can't no longer use the fourth equation of (2.20) regarding kurtosis, the fourth moment. Thus, after we derive a system of three equations and three unknown variables, we result in the solutions illustrated below:

$$p_u(h,k) = \frac{1}{2}\left(\sigma^2\left(\frac{k}{h^2}\right) + a^2\left(\frac{k^2}{h^2}\right) + a\left(\frac{k}{h}\right)\right),$$

$$p_d(h,k) = \frac{1}{2}\left(\sigma^2\left(\frac{k}{h^2}\right) + a^2\left(\frac{k^2}{h^2}\right) - a\left(\frac{k}{h}\right)\right),$$

$$p_m(h,k) = 1 - p_u(h,k) - p_d(h,k) \quad (3.2)$$

where $a = r - q - \sigma^2/2$ and q denotes the instantaneous rate of dividend payout.

It can easily be noticed that the triad of probabilities is also a function of h which still remains unknown. Nevertheless, for a given k, only specific values of h produce positive probabilities for all three nodes simultaneously. Generally h and k are constants, so from the third equation of (3.2) we derive:

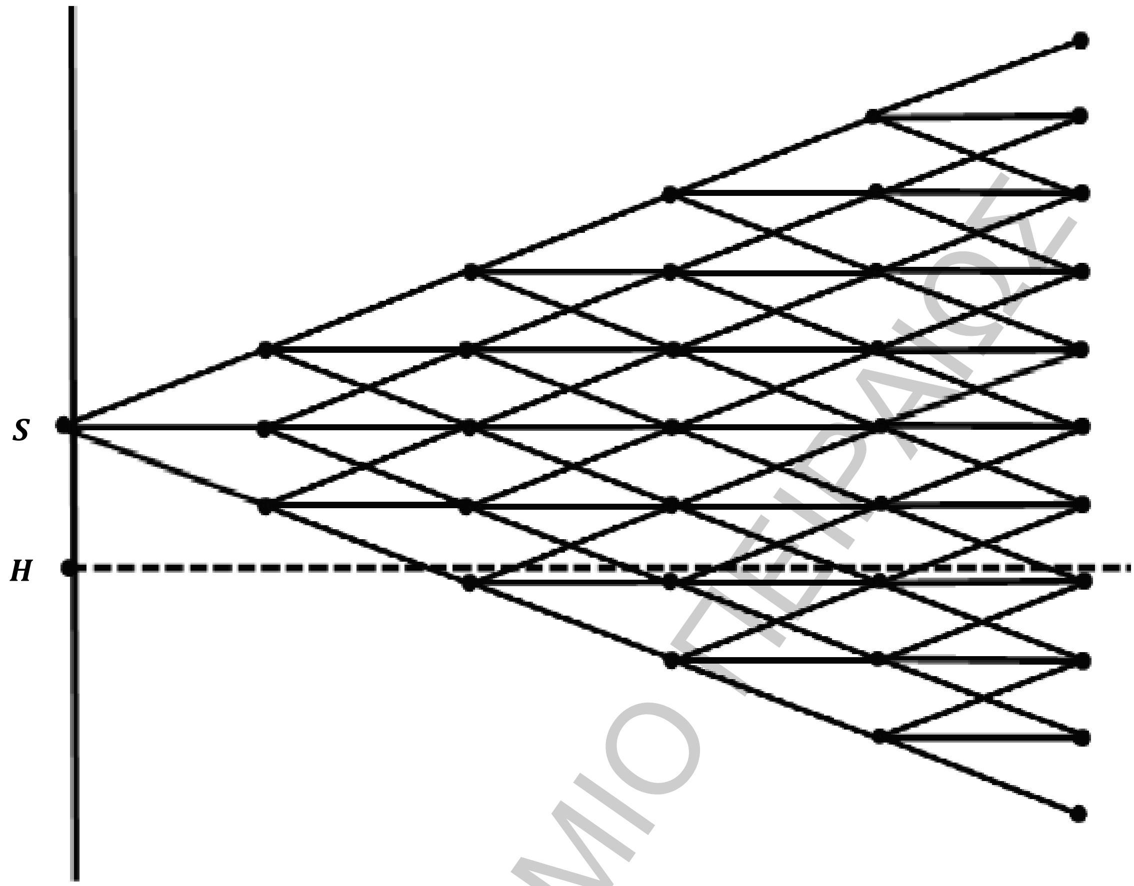
$$p_m = 1 - p_u - p_d \approx 1 - \sigma^2 k/h^2. \quad (3.3)$$

To guarantee the no negativity assumption for the above probabilities h should be of the same order as $\sqrt{k}$. In order to make calculations more understandable we set $\lambda = h^2/(\sigma^2 k)$, which apparently is the last term of equation (3.3) inverted. For $\lambda > 1$ we achieve the latter assumption. Let $\lambda = 3$ and consider the standard trinomial model implied by equations (2.22) and (3.2) in order to value a down and out call with $S = 100$, $K = 100$, $T = 1$ year, $r = 10\%$, $\sigma = 0.25$ and $H = 90$.

In Diagram 3.1 the convergence of the standard trinomial method for the valuation of a down and out call in comparison with the Black and Scholes closed form pricing equation is illustrated. We observe the approximation method first to approach the Black and Scholes benchmark and then suddenly to jump away from it. The typical error is very large even for a trinomial tree of one thousand time steps, which is the approximate equivalent of more than one million node calculations.

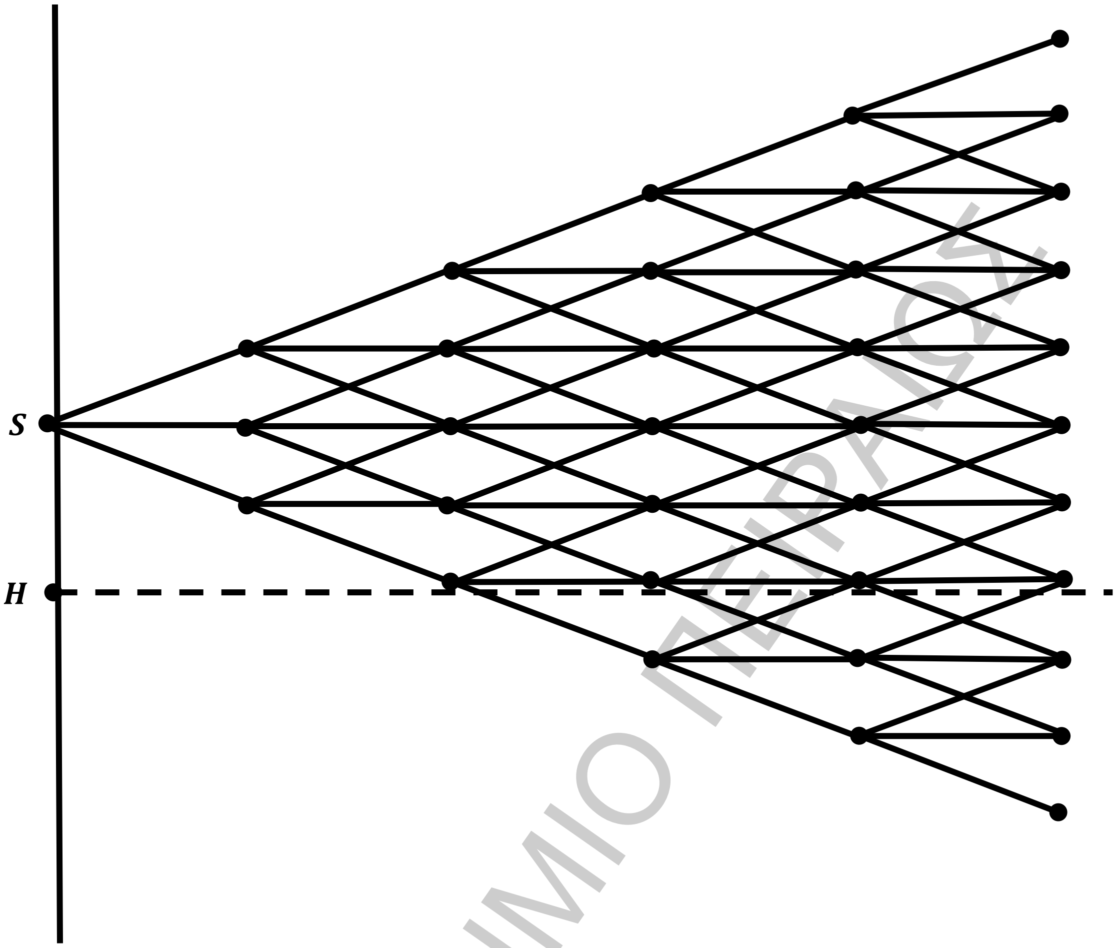
In an attempt to find the cause of this problem we create two figures, Figure 3.1 and 3.2. In the first figure the price barrier H is set above the node that we come across if we move two steps downwards. Practically, this means that whenever the price falls two steps below the initial price S, then the option is knocked out. This fact stops our valuation procedure for a number of nodes that could otherwise be exploited. Therefore, to exploit more information from the tree we need to increase the nodes that are above the barrier line.

Figure 3.1:



In an attempt to create more steps above the knock out we increase the number of time steps by reducing their size and we derive Figure 3.2, where the price step is actually reduced since the barrier price remains constant. It is easily seen that three steps downwards are now needed for the option to be knocked out using the trinomial tree as a guide. Therefore, with the addition of a little computational effort arising from the increase in the number of time steps, a tree with slightly increased number of price steps is produced. The trinomial tree of Figure 3.2 decreases significantly the probability of the option being knocked out and there is accordingly an increase in its estimating value. If our example was an in-option that pays off at maturity only if the barrier has been previously reached, there would be a decrease in the estimating value of the tree by a procedure of this type.

Figure 3.2:



Moreover, depending on the value N of time steps, the price step is the factor that determines how close we are to the barrier. The ideal situation, obviously, would be for the barrier to coincide with a horizontal layer of nodes, which implies a zero price change for the middle node in every movement throughout the lattice and a perfect selection of N. These optimal values for a binomial method have been computed by Boyle and Lau (1994), but as mentioned earlier the binomial method provides us with limited freedom and subsequently cannot be extended for pricing derivatives that include barriers.

However, as illustrated in Ritchken (1995), the performance of this model can be enhanced by restricting variable λ to take values that produce an integer number of price steps between the current asset price and the barrier. Figlewski and Gao (1999) refer to this method as the Restricted Trinomial Model (RTM) and they show that it can be fitted to also match a second barrier also, having more degrees of freedom, but further generalization typically does not work.

Difficulties will emerge, though, if the barrier price is very close to the initial asset price. If the asset price falls below the barrier price even after just one downward price step, in this case the option will be automatically knocked out as far as the trinomial method is concerned. Then an upper bound for the price step must be found in order to overcome such difficulties and it could be no other than $\ln(S_0) - \ln(H)$. As a consequence, this maximum price step selection determines a minimum number N of time steps, that may be very large for small price step values in the case of an initial asset price being very close to the barrier price.

For the down and out call illustrated in Figure 3.2 with $S = 100$, $K = 100$, $T = 1$ year, $r = 10\%$, $\sigma = 0.25$, H=90, assume that the current asset price is much closer to the barrier. If, for example, $S_0 = 90.5$ then the maximum price step that allows us to move one step down before hitting the barrier and knocking out the option is $h = \ln(S_0) - \ln(H) = \ln(90.5) - \ln(90) = 0.00554$. Following this procedure and setting λ equal to 3, from equation $\lambda = h^2/(\sigma^2 k)$ we derive $k = 0.000164$. Practically, if there are 250 trading days in a year and 7 trading hours in a day, this k value means that we have a time step of approximately 17.2 trading minutes. Accordingly, for a year that has 365 days, whether the market is open or closed, this specific value of k translates into a time step of 86.2 calendar minutes. Furthermore, the resulting lattice has $1/k = 6{,}097$ time steps, which means $6{,}098^2 = 37{,}185{,}604$ node calculations. Not to mention that if one wanted to be more accurate and use two price steps to reach the barrier, the lattice should have four times as many time steps, $24{,}388$ to be precise, and almost 600 million nodes.

Obviously, this procedure leads to very slow convergence. Thus we have to improve the standard Trinomial to a more flexible model in order to solve this problem, because even with 1000 time steps we result in a very large pricing error. The estimated option value is 1.102 while the exact value from Merton's formula (3.1) is 0.642 which is about 71.7% mispricing. So as shown in Diagram 3.1 convergence is not stable because for any relatively low N the produced value is not correct. In particular, the option value approaches almost the correct value and then jumps when the number of time steps changes.

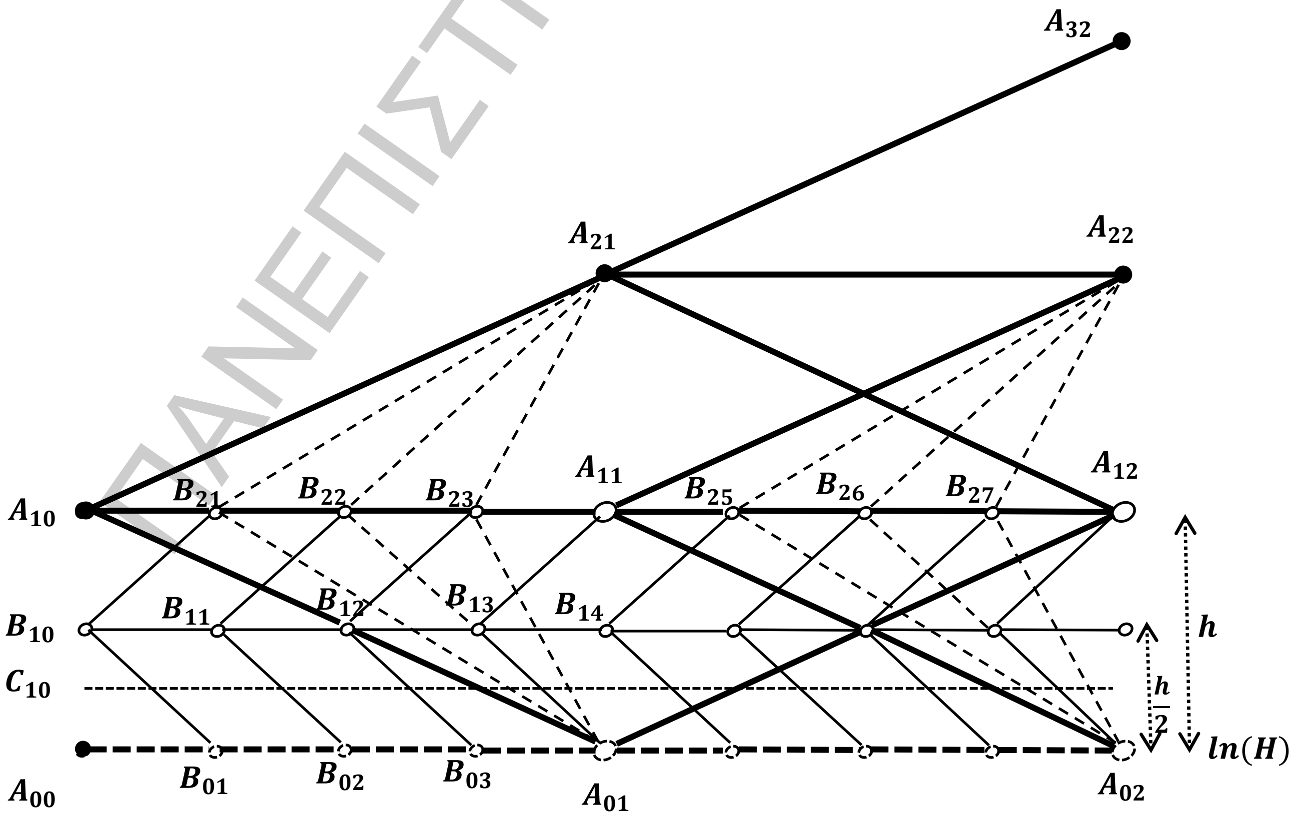
3.1.2 Simple AMM Lattice

The problem of slow convergence can be solved using an adaptive mesh lattice. In this case the high resolution lattice must be near the barrier price and should be adjusted to the coarser tree so that the rest of the tree can still be used for computing. The logic followed here is the opposite to the ordinary option AMM. The information in this case emerges from the coarser lattice and it leads to the finer lattice, while in the ordinary option AMM the information flowed from the high resolution lattice to the coarser one.

The adaptive mesh structure we wish to construct is illustrated in Figure 3.3. The bold lined tree is the coarse mesh lattice, where the nodes are labeled by A_{ij}, with i and j the number of the coarse mesh price steps over the barrier and the number of the coarse time steps, respectively.

Obviously, the A nodes lying on the bold dashed line are the ones that fall exactly on the barrier price. The adaptive mesh nodes are noted by the letter B and create the light-lined tree with a reduced price step to the half of the price step of the coarser tree.

Figure 3.3:



At first we compute the values of the coarse tree, creating the prices for all the A labeled nodes. Next, we use the coarse mesh lattice to compute the option values of the fine mesh lattice at the price levels $\ln(H)$ and $\ln(H)+h$ for time intervals of length $k/4$. Then, we compute the remaining fine mesh nodes for the price $\ln(H)+h/2$ at the same time intervals. The light concrete and dashed lines are indicating the nodes that were used for pricing each node at the left of each branch. Following this procedure we can calculate the option price at B_{10}. We must note at this point that the option values at the B_{0j} nodes are all equal to zero since they fall on the knock out barrier. On the contrary, in the case of barrier options that pay a rebate when they are knocked out these values are set equal to the rebate price. The fine dotted line starting with C_{10} at price $\ln(H)+\frac{h}{4}$ indicates where the second level of fine mesh would be placed.

Further details about the structure illustrated in Figure 3.3 are given below. The nodes B_{24} and B_{28} coincide with the nodes A_{11} and A_{12} of the coarse tree so we avoid recalculating them. In contrast, nodes B_{21}, B_{22} and B_{23} are computed rolling backwards from nodes A_{01}, A_{11} and A_{21}. Obviously, we will use equation (2.22) for this backward calculation, but some changes must be made otherwise we would create just one value instead of three. The key difference between nodes B_{21}, B_{22} and B_{23} is that they are falling on dates which are $3k/4$, $2k/4$ and $k/4$, respectively, from the next coarse lattice time step. Henceforth, three different sets of probabilities are connected to the latter nodes. To obtain these probabilities we replace k in the set of equations (3.2) by each node's respective time step size.

In order to make the preceding procedure understood, consider for example the calculation of the node B_{23}. Using $k/4$ instead of k in the set of equations (3.2) and we derive the following set of equations:

$$p_u(h,k/4)=\frac{1}{2}\left(\sigma^2\left(\frac{k/4}{h^2}\right)+a^2\left(\frac{k^2/16}{h^2}\right)+a\left(\frac{k/4}{h}\right)\right),$$

$$p_d(h,k/4)=\frac{1}{2}\left(\sigma^2\left(\frac{k/4}{h^2}\right)+a^2\left(\frac{k^2/16}{h^2}\right)-a\left(\frac{k/4}{h}\right)\right),$$

$$p_m(h,k/4)=1-p_u(h,k/4)-p_d(h,k/4). \quad (3.4)$$

The next step is to use these probabilities in equation (2.22) and derive that the value of the option at node B_{23} is:

$$C(B_{23}) = e^{-rk/4}\big(p_u(h,k/4)C(A_{21}) + p_m(h,k/4)C(A_{11}) + p_d(h,k/4)C(A_{01})\big), \quad (3.5)$$

where by using nodes as variables we mean the vector of the price step and time step of each node in order to make the notation more easily understood.

Consequently, having computed the nodes B_{0j} and B_{2j} for every j, we define a new set of probabilities for the derivation of nodes at the price step of size $h/2$ and time steps of length $k/4$. Similarly to the prior derived probability set as an example, we present the application of the procedure for node B_{10} :

$$p_u(h/2,k/4) = \frac{1}{2}\left(\sigma^2\left(\frac{k}{h^2}\right) + a^2\left(\frac{k^2/4}{h^2}\right) + a\left(\frac{k/2}{h}\right)\right),$$

$$p_d(h/2,k/4) = \frac{1}{2}\left(\sigma^2\left(\frac{k}{h^2}\right) + a^2\left(\frac{k^2/4}{h^2}\right) - a\left(\frac{k/2}{h}\right)\right),$$

$$p_m(h/2,k/4) = 1 - p_u(h/2,k/4) - p_d(h/2,k/4). \quad (3.6)$$

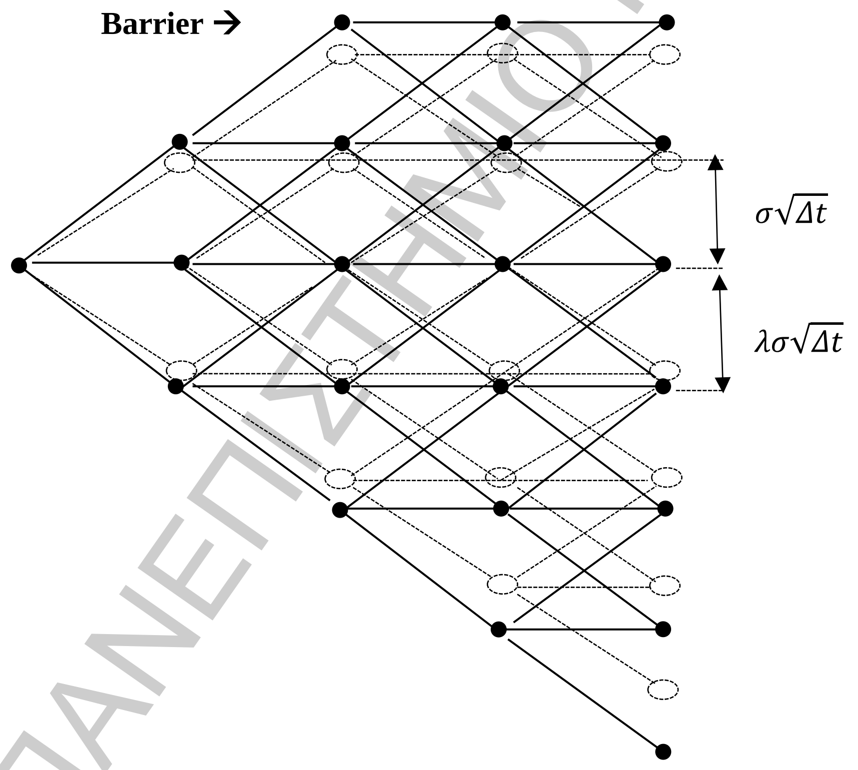
Finally, in order to compute the nodes B_{1j} we start from the node at expiration and we use the recursive scheme of (2.22) rolling back through the tree and eventually deriving the option value of B_{10} as:

$$C(B_{10}) = e^{-rk/4}\big(p_u(h/2,k/4)C(B_{21}) + p_m(h/2,k/4)C(B_{11}) + p_d(h/2,k/4)C(B_{01})\big). \quad (3.7)$$

3.1.3 General AMM Lattice

Having constructed in Figure 3.3an adaptive mesh lattice with only one level of finer mesh, we can easily construct an adaptive mesh lattice with more than one levels of finer mesh simply by adding every time an additional level with a price step that is the half of that of the former level. The dashed line starting with C_{10} in Figure 3.3 shows exactly where the middle nodes of the second level of fine mesh would be. The base though to every level of fine mesh lattice is the coarse tree which must be set up carefully so that the levels of fine mesh that will follow are as accurate as we want. In order to achieve this effectiveness we must ensure that the current asset price is exactly one price step above the barrier and the number of coarse mesh steps in the tree is an integer.

Figure 3.4:



In Subsection 3.1.1 we have set $\lambda = h^2/(\sigma^2 k)$, while we arbitrarily imposed λ to be 3. Because of the peculiarity of barrier options we can no longer select freely, h and k so that the above relationship is just satisfied. In Ritchken (1995) λ is the stretch parameter, because it is actually a factor that defines how much we should stretch the tree so that the tree's price nodes lie exactly on the out barrier. Figure 3.4 illustrates exactly that stretch for the simple barrier model.

An approximation procedure can be used at this point so that we can define k. At first, we calculate the non-integer number of steps the targeted λ would yield and then lengthen the time step just enough to eliminate the rounding error. A simple general calculation formula presented in Figlewski and Gao (1999) for the calculation of the time step k is:

$$k = T/int((\lambda\sigma^2/h^2\ )T). \quad (3.8)$$

The procedure followed in order to create an AMM with several levels of fine mesh is the following. At first, we define how many levels of fine mesh are needed. Let M be that integer number. Obviously for the example presented in the previous subsection the value of M is equal to unity. The maximum price step is set to $\ln(S_0) - \ln(H)$, and by reducing in half for every new level of fine mesh we derive the following formula for the price step h of each level:

$$h = (\ln(S_0) - \ln(H))/2^M. \quad (3.9)$$

We construct the A nodes starting with the initial node with asset price $X = \ln(S_0)$. Consequently, we construct the B, C etc. nodes by adding every new level of finer mesh on the former lattice. As far as consistence of this method is concerned the proof is illustrated in Figlewski and Gao (1999) and can easily be applied to other AMM structures.

The number of calculations needed for every level of fine mesh can be easily found if we have in mind Figure 3.3. Apart from the A nodes we see three groups of B nodes, the upper, the middle and the down ones. For the upper and lower groups three new possible values must be computed for each node that do not coincide with an A node. For the middle group there are four new nodes for every node in the coarse tree. Summing the calculations needed we get $3 + 3 + 4 = 10$ nodes for each time step and $10N$ in total. Therefore, a lattice consisted of 100 steps contains $101^2 = 10{,}201$ nodes from the coarse mesh and $10 \cdot 100 = 1{,}000$ nodes from the fine mesh, resulting in a total of $11{,}201$ nodes to be computed. In contrast, if we did not make use of AMM and decide to increase the resolution in a standard trinomial model by halving the price step and by quadruplicating the number of time steps then the calculation of $401^2 = 160{,}801$ nodes will be required. Practically we avoid the calculation of $160{,}801 - 11{,}201 = 149{,}600$ nodes, an amount of nodes by no means negligible in comparison to the $11{,}201$ nodes that are going to be computed with the AMM technique.

In case the accuracy needed is not reached we may always add some new levels of fine mesh on the lattice. So counting the nodes we need 10 new nodes for every time step and in general if we have m levels of fine mesh then $10 \cdot 4^{m-1} \cdot N$ new nodes will be created. More details about the computational comparison are illustrated in Table 3.1.

Concluding, the advantage of the AMM is that although the standard trinomial method produces smaller distribution approximation errors, the AMM approaches the standard trinomial method's nonlinearity error. This occurs because at the critical areas near the barrier both models have the same size of price step for the layer adjacent to the barrier.

3.2 Double Barrier Option

For the discrete double barrier option pricing we can distinguish two distinct cases. In the first case the up and down barrier are far away from each other. So there is no intersectional area between the adaptive mesh lattices for the up and down barrier. The procedure followed is exactly the same as for the simple barrier with the only difference that is applied two times, one for the upper barrier and one for the lower barrier. The second case, though, is very interesting because the two AMM lattices intersect with one another. This usually happens when the barriers are extremely close. Figure 3.5 illustrates exactly this situation.

In order to have a simple illustration, only the AMM lattice for the lower barrier is shown in Figure 3.5. However, this does not mean that the AMM lattice for the upper barrier does not exist. The coarse lattice which is covered with A nodes is the lattice with the wider links. The smaller solid links define the links between B

nodes of the first layer of AMM and finally the dashed lines define the branches of the second layer of AMM with the C nodes.

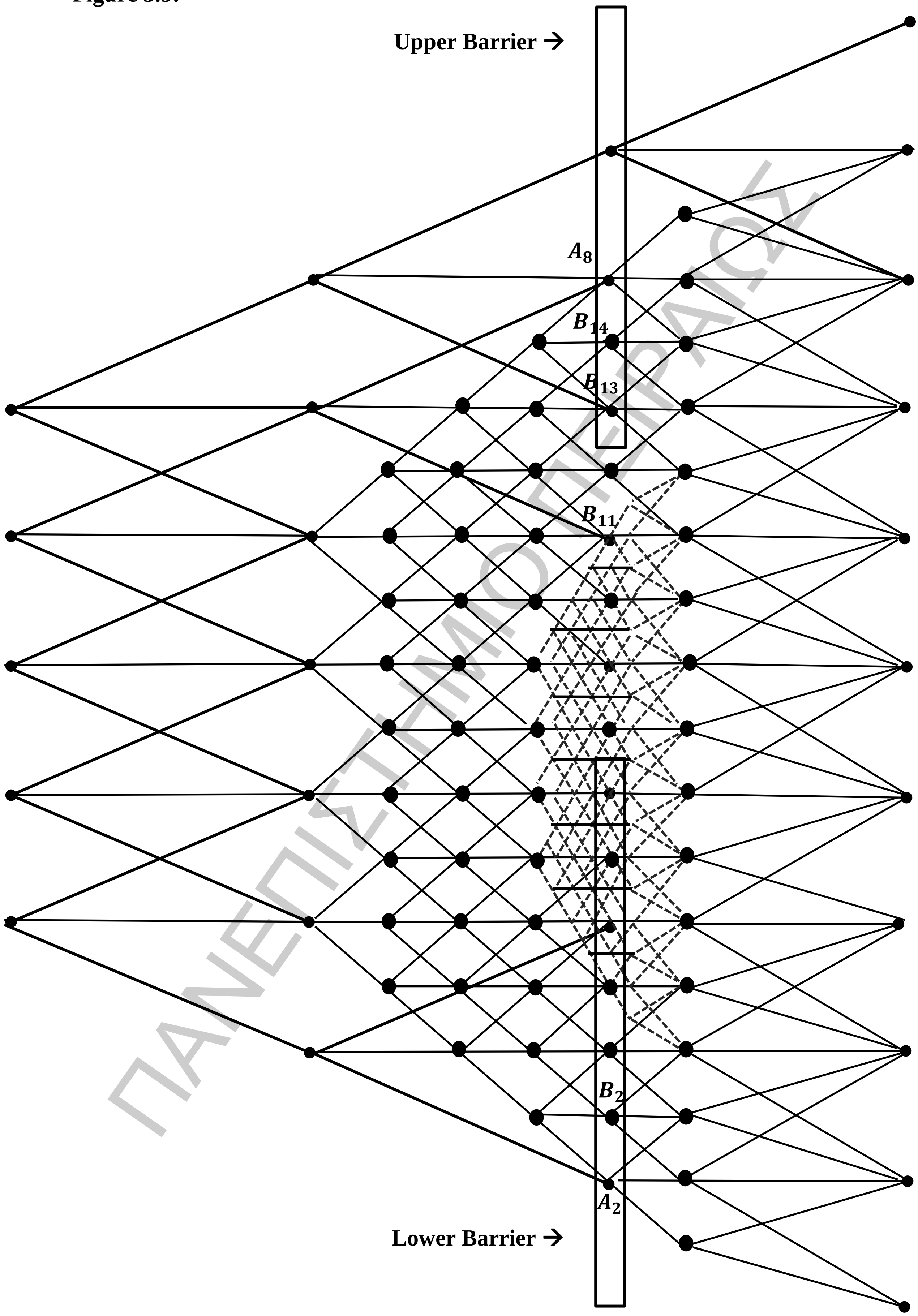
If the upper barrier is so close to the lower that node B_{11} belongs to the intersection of the nodes of the two AMM then the second layer of AMM will also be affected. As Figure 3.5 shows the proximity of the upper barrier to the AMM 1 layer of the lower barrier automatically knocks out the nodes B_{13}, B_{14} and B_{15} which coincides with A_8. In order to avoid the double calculation of the same nodes we must use a common AMM lattice for both barriers in the intersecting areas.

The environment of this type of a double barrier is quite unfriendly for the formation of an algorithm that could be programmed to a computer. Nevertheless, in case someone wants to try and create an algorithm some facts observed for this case of AMM lattice are the following:

1) If two m-level adaptive meshes are combined, then the two (m-1)-level adaptive meshes should be combined.
2) If the two m-level adaptive meshes are individual, then the two (m+1)-level adaptive meshes should also be individual.

Obviously for double barrier options there are two sections where nonlinearity error would emerge from; the sections adjacent to the barriers. In order to decrease nonlinearity error with a standard trinomial model we would half the price step but this action would have a quadruple negative effect on the number of nodes to be calculated, making this procedure unbearable in terms of computational effort. However, if we attach a set of adaptive mesh layers to the coarse lattice of Figure 3.5 then we would need 60 new nodes to be calculated. These 60 nodes are practically all the B nodes of the first layer of the two adaptive meshes. From these 60 nodes though many would be knocked out so we do not need any extra calculation for the next layer of fine mesh. Moreover, by increasing the number of nodes maximum by 120 nodes, if the upper and lower barrier lattices are individual, we can decrease the nonlinearity error in half.

Figure 3.5:



4. Gamma and Delta Calculation

A precise calculation of delta and gamma is very important for researchers and traders. On the one hand, researchers usually focus on the valuation of an option through numerous option pricing models but they could also focus on risk management through option valuation. On the other hand, traders make adjustments to volatility and other input parameters in order to setup a valuation model that will match in values the observed market prices. As mentioned in Subsection 1.2.3, risk exposure is measured mostly by delta and gamma, i.e. the change in option value given a small change in asset price and the change in delta, respectively. If we have acquired a closed form valuation model then delta and gamma can theoretically be obtained in closed form following the equations (1.1) and (1.2).

In this chapter we will demonstrate different approaches for estimating delta and gamma in a lattice model. The main differences of these approaches are the accuracy and speed of convergence. Using an adaptive mesh technique, accuracy improves significantly.

4.1 Trinomial Lattice

If we use a binomial model then it will be easy to estimate delta because the number of shares of a riskless portfolio is practically one estimator of delta. On the contrary, in a trinomial model there is no comparable delta estimator because, unlike binomial, the trinomial model is not based on option replication.

In the trinomial model the procedure for delta and gamma estimation is the following. At first we create an interval around the initial log price, using a small amount ε, and begin to form new lattices from the edges of this specific interval. So we have three lattices starting from asset prices $X_0, X_0 + \varepsilon$ and $X_0 - \varepsilon$, respectively. This procedure, though, creates very noisy estimates due to the nonlinearity error and in the same manner as with option pricing; the estimates do not converge monotonically. Also, this procedure demands a second and a third recalculation of the entire lattice for the estimation of both delta and gamma.

Alternatively, as long as we use log prices for the new asset prices, delta and gamma calculation formulas can be altered as follows:

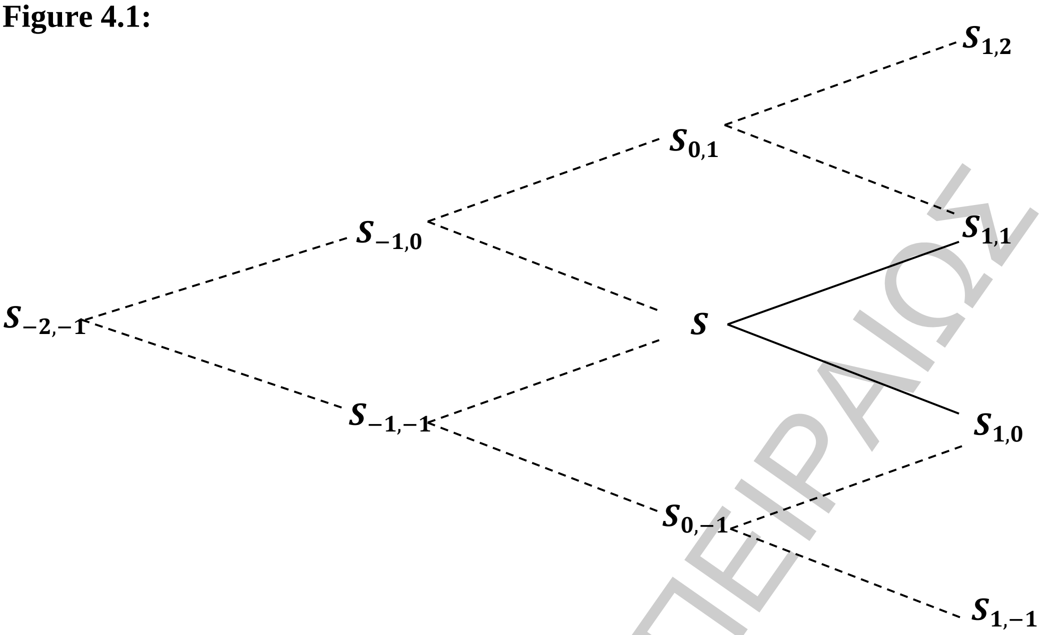
$$\Delta = \frac{\partial C}{\partial S} = \frac{\partial C}{\partial \ln(S)} \frac{1}{S} \approx \frac{C(X_0 + \varepsilon\,) - \mathrm{C}(X_0 - \varepsilon)}{2\varepsilon} \frac{1}{\mathrm{S}},$$

$$\Gamma = \frac{\partial \Delta}{\partial S} = \frac{\partial^2 C}{\partial S^2} = \frac{\partial C}{\partial S}\left(\frac{\partial C}{\partial \ln(S)} \frac{1}{S}\right) = \left(\frac{\partial^2 C}{\partial (\ln(S))^2} - \frac{\partial C}{\partial \ln(S)}\right) \frac{1}{S^2},$$

$$\approx \left(\frac{C(X_0 + \varepsilon) + C(X_0 - \varepsilon) - 2C(X_0)}{\varepsilon^2} - \frac{C(X_0 + \varepsilon) - C(X_0 - \varepsilon)}{2\varepsilon}\right) \frac{1}{S^2}, \quad (4.1)$$

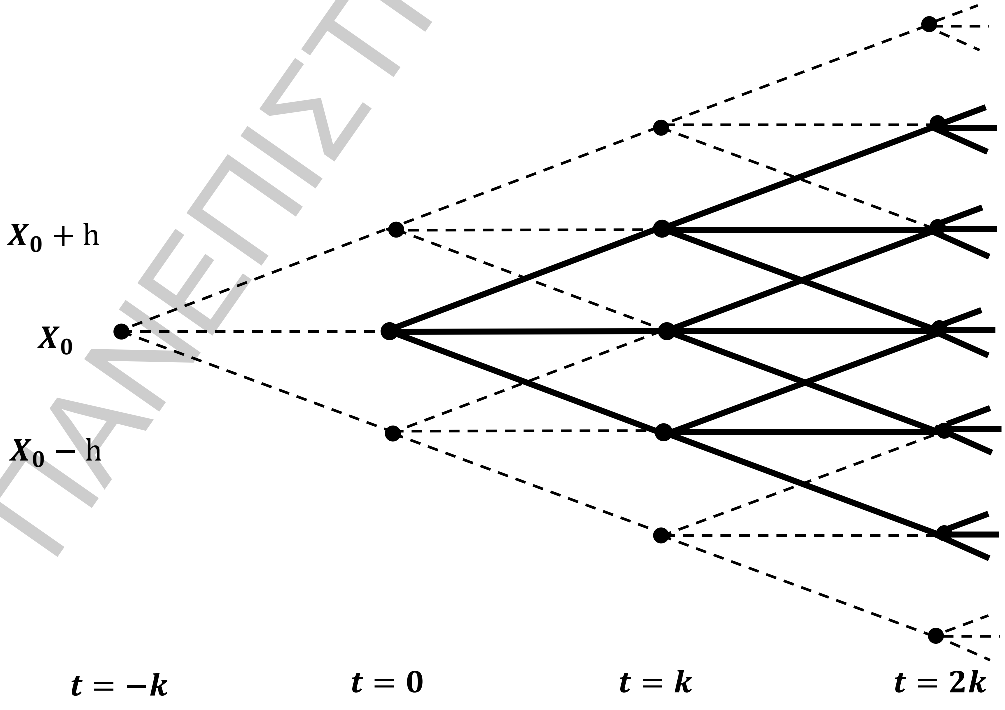
where we have employed a central finite difference approach for the partial derivatives. The problem arising here is that we have to compute two whole trees one for each perturbation of X_0. Figlewski and Gao (1999) generalized an idea presented in Pelsser and Vorst (1994), whose concept for a binomial model is to start two time steps earlier from time 0, as we usually begin, and construct a tree in such a way that the middle node of the new tree's second time step would coincide with the original initial asset price as shown in Figure 4.1.

Figure 4.1:



In Figure 4.1 the extensions of the binomial tree are the dashed lines and the original tree is the solid line tree. Also, the initial asset price is denoted by S. Generalizing for the trinomial lattice we construct the tree illustrated in Figure 4.2.

Figure 4.2:



You may notice that for the trinomial tree only one step backwards is needed because its middle node connection enables the tree to fall onto the initial price node in a single step. Using the same notation, the dashed lines are the newly created extensions of the tree, while the solid ones are the original tree branches. Apparently, extending the tree one period backwards is a much simpler procedure than the creation of two whole new lattices based on $\boldsymbol{X_0} + \mathrm{h}$ and $\boldsymbol{X_0} - \mathrm{h}$. The benefit of this tactic is a total of $2N + 2$ node calculations; in fact, we do not include in these calculations the single node in time $t = -k$ because it has no actual use in practice. In order to obtain delta and gamma values we substitute the price step h for ε in the set of equations (4.1).

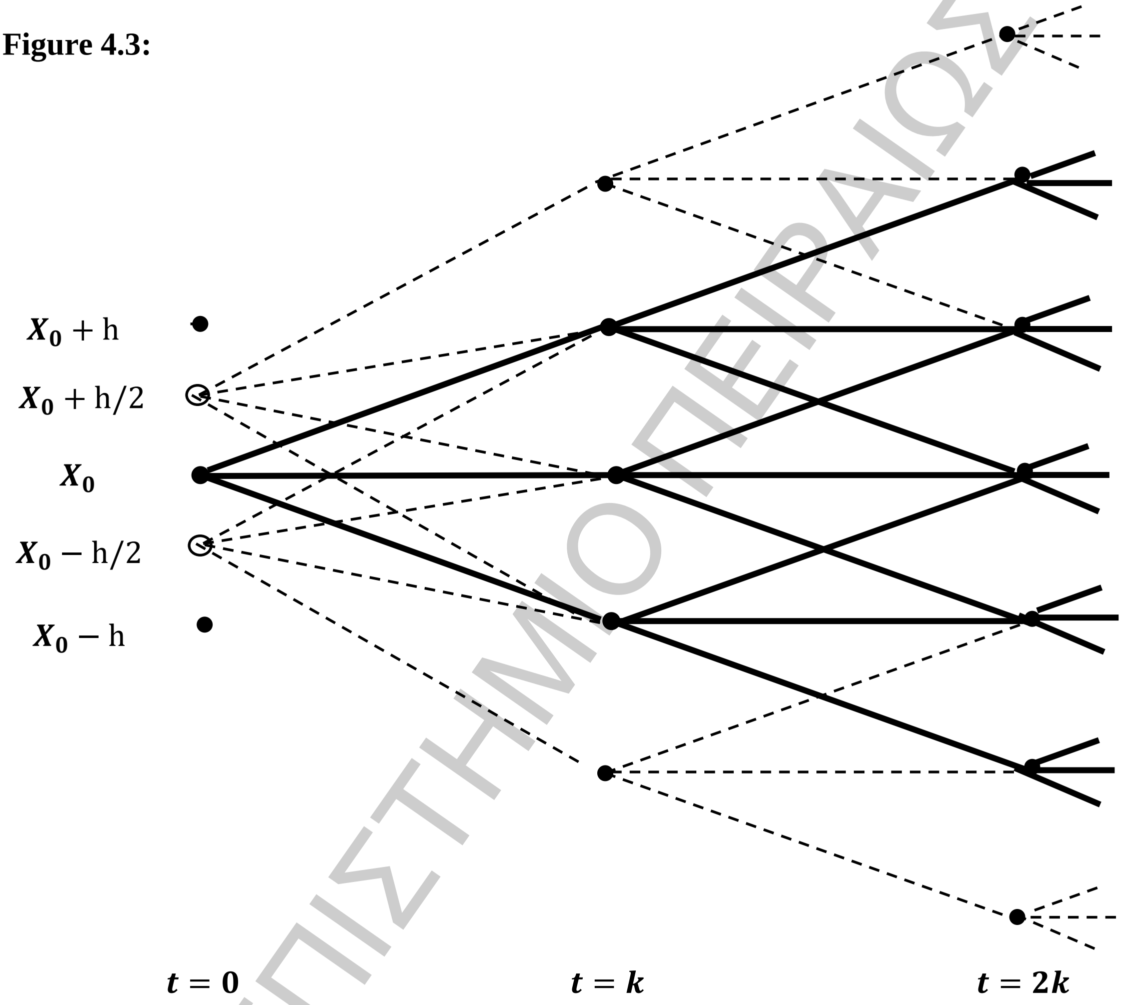
Using the same test set as in Table 2.1, Table 4.1 compares different approaches of estimating gamma and delta with trinomial lattices for 4 different N. In a similar way as in Table 2.1, the root mean squared errors are relative to the values produced by the closed form formula of Black and Scholes. The first two lines of every set in Table 4.1 are using a perturbation of $\varepsilon = 0.001$ and $\varepsilon = 0.01$, respectively. Every third line gives results for the method of extending the tree one step back in time. We expect that the perturbation methods should take at least two times as long as simply pricing the option. But, an interesting effect is produced, since pricing with smaller ε the accuracy is less than that of pricing with larger ε. The explanation given to this phenomenon is that the division with very small numbers, like $\varepsilon = 0.001$, magnifies the nonlinearity error effect. So it is not recommended to use small ε in the valuation of gamma and delta. Nevertheless, the trinomial tree extension takes less time to compute and actually gives more accurate results, especially for gamma.

4.2 AMM (Quadrinomial Lattice-Based)

In the present section the use of an adaptive mesh lattice around the initial asset price will be illustrated. At first, we will create only one layer of adaptive mesh to increase our accuracy on the estimation of delta and gamma. Secondary, more layers of adaptive mesh will be added on the trinomial tree. A quadrinomial model will be used for the added adaptive mesh because the initial nodes of the AMM will fall between a pair of nodes of the trinomial lattice, making it impossible to select three nodes of the trinomial to connect with each initial node of AMM.

4.2.1 Simple AMM Lattice

Figure 4.3:



In this section we shall show how an adaptive mesh model with a region of fine mesh around the initial asset price allows numerical derivatives to be computed using any perturbation ε subject to only a very small increase in the total number of node calculations. In Figure 4.3 a lattice set up much like the one in Figure 4.2 is illustrated, with perturbation trees that overlap the original lattice and add new nodes only at the highest and lowest prices in each period diminishing accordingly the computational effort. The difference here is that these new sections of lattice begin at time 0 at asset prices of $X_0 + h/2$ and $X_0 - h/2$. These deviations from the original asset price X_0 are actually half as large as in the example of Figure 4.2.

One may observe four links starting from the new nodes in Figure 4.3, while the links starting from the original asset price are only three according to the definition of a trinomial model. This new linking procedure is a quadrinomial branching for the two new nodes. The extended tree shown in Figure 4.2 adds a node at the price $X_0 + h$ for time 0 from which the only links would be $X_0 + 2h, X_0 + h$ and X_0. In contrast, the quadrinomial branching links the time 0 and price $X_0 + h/2$ node with any of the four possible nodes in front of it. This kind of modification makes all three original tree nodes accessible from the new ones. So the perturbation used in delta and gamma calculation is actually reduced in half without any change on the number of new nodes demanded.

The use of quadrinomial branching solves the problem of negative probabilities in the case of trinomial branching from a node like $X_0 + h/2$; a node that does not belong in the original tree. Nevertheless, quadrinomial branching necessitates the calculation of a fourth probability. These four probabilities attached to the four possible next period nodes are $p_{uu},\ p_u,\ p_d,\ p_{dd}$. These probabilities link the new node $X_0 + h/2$ to the k time nodes placed at the prices that are $+\frac{3}{2}h, +\frac{h}{2}, -\frac{h}{2}$ and $-\frac{3}{2}h$ far respectively, from itself.

Using the system of equations (2.20), we defined earlier the three probabilities for the trinomial lattice and pinpointed the relationship between h and k. In the quadrinomial case, though, the price and time steps are predetermined by the construction of the lattice. So there is not any extra degree of freedom available. In the same way of thinking these four probabilities must obey the following conditions:

(1) The expected return over the next time step is the riskless rate
(2) The second moment is consistent with the volatility σ of the underlying asset
(3) The skewness, i.e. the third moment, is equal to zero like in the normal distribution
(4) The probabilities have a total of 1.

Quantifying the above assumptions, we obtain

$$\frac{3}{2}hp_{uu}+\frac{1}{2}hp_u-\frac{1}{2}hp_d-\frac{3}{2}hp_{dd}=0,$$

$$\left(\frac{3}{2}h\right)^2\ p_{uu}+\left(\frac{1}{2}h\right)^2\ p_u+\left(\frac{1}{2}h\right)^2\ p_d+\left(\frac{3}{2}h\right)^2\ p_{dd}=\sigma^2k,$$

$$\left(\frac{3}{2}h\right)^3\ p_{uu}+\left(\frac{1}{2}h\right)^3\ p_u-\left(\frac{1}{2}h\right)^3\ p_d-\left(\frac{3}{2}h\right)^3\ p_{dd}=0,$$

$$p_{uu}+p_u+p_d+p_{dd}=1. \quad (4.2)$$

If we solve this system of four equations with four unknowns, keeping in mind that $h^2=3\sigma^2k$, we conclude that:

$$p_{uu}=p_{dd}=1/48,$$

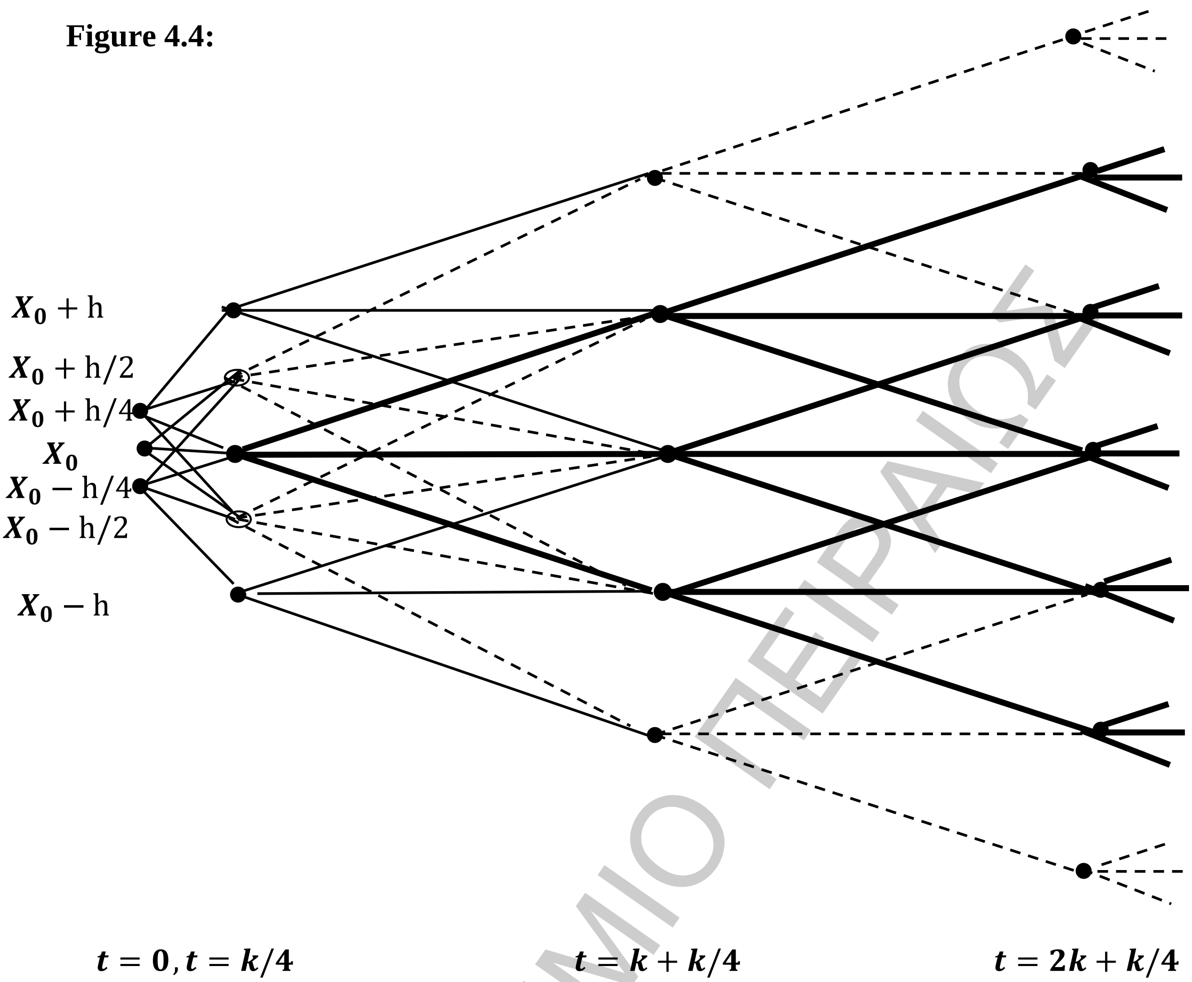
$$p_u=p_d=23/48. \quad (4.3)$$

4.2.2 General AMM Lattice

In this subsection we will illustrate the procedure through which more layers of fine mesh can be added in the trinomial tree. In Figure 4.4 a lattice with two layers of fine mesh is illustrated.

In Figure 4.3 we have illustrated the lattice with one layer of fine mesh attached at the beginning of the tree, using a price step of h and a time step of k, which was extended by two nodes at time 0 being $h/2$ far from the initial price X_0. So the final extended lattice has 3 nodes at time 0 whose prices are $h/2$ apart. These nodes are linked with the five nodes of time k which differ by h.

Figure 4.4:



In order to achieve the second layer of fine mesh we have to add a new section of lattice with a price step of $h/2$ and a time step of $k/4$ and then connect it to the five nodes in the first layer of AMM lattice. The initial asset price X_0 remains at the center of the lattice at time 0. From this particular node three branches are connected to three fine mesh nodes at time $k/4$ with prices $X_0 + h/2$, X_0 and $X_0 - h/2$, respectively.

Following the same reasoning as in the creation of the first layer of the adaptive mesh we place at time 0 nodes one quarter of a price step above and below X_0; practically this means nodes for price steps of $X_0 + h/4$, X_0 and $X_0 - h/4$. Facing the same problem as in the first layer of adaptive mesh, we cannot create a middle branch for the two new nodes following a trinomial branching procedure, so we use quadrinomial branching. Therefore three nodes at time 0 connect to five nodes at time $t = k/4$ making necessary the calculation of nodes at price $X_0 + h/2$ and $X_0 - h/2$ at time $t = k/4$. We must note at this point that these two nodes were not necessary for the first adaptive mesh lattice calculation.

The bold lines in Figure 4.4 indicate the coarse lattice with price step size h and time step size k, the dashed lines represent the first adaptive mesh added and the thinner solid lines are the branches of the second layer of adaptive mesh. Following the same procedure we could create a third layer of adaptive mesh with a finer time step of $k/16$, and three new nodes at time 0 with prices $X_0 + h/8$, X_0 and $X_0 - h/8$, respectively. In general if a middle branch can be used for the move from the one node to the next then we use trinomial branching, if not we use quadrinomial branching.

This was the procedure to build the AMM lattice with two layers. Now we are going to create a general AMM lattice with N coarse steps and M AMM layers. The only exception on the layers is the first layer which does not demand the addition of a fine mesh with shorter time step. So for every layer $m > 1$ of adaptive mesh, there will be a time step of length $k/4^{m-1}$. Furthermore, the time length of the extended tree will be $(N + \frac{1}{4} + \cdots + 1/4^{M-1})^k$. If the option's maturity T and volatility σ are given, then the values for h and k are derived from the following formulas:

$$k = \frac{T}{N - 1 + \sum_{m=1}^{M} 1/4^{m-1}},$$

$$h = \sigma\sqrt{3k}. \quad (4.4)$$

In order to make the lattice easier to understand from a computational point of view, denote the price and time steps for the m^{th} level of adaptive mesh as $h_m = h/2^{m-1}$ and $k_m = k/4^{m-1}$. The very first node must be placed at time 0 and have the price X_0, while the two nodes that will be used for the computation of both delta and gamma have price values of $X_0 + h_m/2$ and $X_0 - h_m/2$, respectively. Every three nodes of this type are then linked to five nodes of the next time step which is k_m far from the present time step. Obviously for the nodes at X_0we use the trinomial branching procedure. After the nodes and branches for the first k_m steps are produced, the next step in the procedure is to double the price step and multiply the time step by four following the same technique of branching and node creation. At the point where there is no more fine mesh there will be connection to the coarse lattice and then connection eventually to the maturity date of the option.

As far as performance is concerned Table 4.2 illustrates the difference of a simple trinomial lattice to several other adaptive mesh models. A first try with the addition of adaptive meshes at the beginning of the coarse lattice improved the estimates of delta and gamma. But, with a second thought these sizes are affected by the nonlinearity effect presented mainly near the expiration time. So a second set of adaptive mesh layers were added at expiration so that delta and gamma values would be more accurate than before. One main difference is that the accuracy is improved by almost 67% for delta and 50% for gamma in the AMM 1 model, without any effect in the execution time. Although with a second addition of fine mesh delta is a little improved, gamma on the other hand is not.

Continuing our experimentation by adding adaptive meshes both in the beginning and at the maturity of the option we observe radical improvement in delta. For the case of gamma, the calculation becomes more accurate when only one adaptive mesh layer is added on the beginning and the end. If more layers are added not only it does not improve, but the accuracy also deteriorates for every added layer. For every new layer of fine mesh added we must note that the execution time is slightly increased incomparable to the large improvement in accuracy. Thus the benefit from the use of AMM lattice is significant.

5. Conclusion

Throughout the years the option valuation formula of Black and Scholes has been widespread, but as the market derivatives evolved it became even more difficult to create a new closed form formula for their valuation. American options and more exotic options could not be priced with the use of a closed form formula, even though the no-arbitrage condition continued to hold. This was the beginning of a new era; the era of lattice-based models. These models being intuitive and flexible offered an easy solution in the problem of derivative valuation, especially for researchers who could not fit the idea of Black and Scholes to their needs.

As illustrated before, two of these lattice models are the Binomial and Trinomial. Using these models as a flexible base there was an improvement in computation performance for a certain level of accuracy. Still, though, many important problems of pricing common derivative instruments remained theoretically solvable but practically infeasible with the standard methods. The main cause for this problem is that they required a vast amount of computations for a certain level of accuracy.

A partial solution to this problem is given by the AMM. So in this thesis several structures of AMM were exploited. At first we used an AMM for the valuation of an American put option and we managed to attach an adaptive mesh lattice near the expiration of the option. Furthermore, in Chapter 2 we illustrated how to construct one or multiple layers of adaptive mesh lattice near the barrier of a down and out barrier option. Moreover, an AMM structure for double barrier options was also illustrated in the second part of the latter chapter. Following the same context, procedures for the valuation of delta and gamma of options were illustrated in Chapter 4. There is also a sharp improvement in the accuracy of our calculations for the delta and gamma when we add an adaptive mesh near the maturity of the option.

As illustrated explicitly, these adaptive mesh extensions improved in a major way their predecessors; the naked Binomial and Trinomial models. When two AMM were combined, one near the initial price of the asset price and one near the expiration date, significant improvements occurred for both the valuation of the option price and the estimation of delta and gamma.

AMM structures can also be adapted in even more complex derivative valuation due to their flexibility. In the bibliography, Ahn, Figlewski and Gao (1999) and Chih-Jui Shca in his thesis have illustrated AMM structures for the valuation of such derivatives. Proposals for research are endless on this sector of expertise. Future research could be occupied with more complex derivatives like barrier options with variable barrier or discontinuous barriers. Furthermore, in other derivatives it may not be obvious which areas are critical in order to need an adaptive mesh attached. So, an index for such areas should be created. If such an index exists and can be used, then we would be able to target exactly the critical areas, where more details are needed, and discharge our valuation model from the computational effort.

REFERENCES

1. Ahn, D. G., Figlewski, S. and Gao, B. (1999) 'Pricing Discrete Barrier Options with an Adaptive Mesh Model', *Journal of Derivatives,* Vol. 6, pp. 33-44.
2. Black, F., Scholes, M. (1973). 'The Pricing of Options and Corporate Liabilities', *Journal of Political Economy*, Vol. 81, pp. 637-659.
3. Boyle, P. (1988) 'A Lattice Framework for Option Pricing with Two State Variables', *Journal of Financial and Quantitative Analysis*, Vol. 35, pp. 1-12.
4. Boyle, P. and Lau, S. H. (1994) 'Bumping Against the Barrier with the Binomial Method', *Journal of Derivatives,* Vol. 1, pp. 6-14.
5. Brennan, M., Schwartz, E. (1977) 'The Valuation of American Put Options', *Journal of Finance,* Vol. 32, pp.449-462.
6. Broadie, M., Detemple, J. (1996) 'American Option Valuation: New Bounds, Approximations, and a Comparison of Existing Methods', *Review of Financial Studies,* Vol. 9, pp.1211-1250.
7. Broadie, M., Glasserman, P. and Kou, S. G. (1997) 'A Continuity Correction for Discrete Barrier Options', *Mathematical Finance,* Vol. 7, pp. 325-349.
8. Broadie, M., Glasserman, P. and Kou, S. G. (1999) 'Connecting Discrete and Continuous Path-Dependent Options', *Finance Stochastics,* Vol. 3, pp. 55-82.
9. Canina, L., Figlewski, S. (1993). 'The Informational Content of Implied Volatility', *Review of Financial Studies,* Vol. 6, pp. 659-681.
10. Cheuk, T. H. F. and Vorst, T. C. F. (1995) 'Complex Barrier Options', *Journal of Derivatives,* Vol. 4, pp. 8-22.
11. Cho, H., Lee, K. (1995) 'An Extension of the Three-Jump Process Model for Contingent Claim Valuation', *The Journal of Derivatives,* Vol. 3, pp. 102-108.
12. Cox, J. and Ross, S. (1976) 'The Valuation of Options for Alternative Stochastic Processes', *Journal of Financial Economics,* Vol. 3, pp. 145-166.
13. Cox, J., Ross, S. and Rubinstein, M. (1979) 'Option Pricing: A Simplified Approach', *Journal of Financial Economics,* Vol. 7, pp. 229-264.
14. Cox, J. and Rubinstein, M. (1985) *Options Markets,* Englewood C M , NJ.
15. Figlewski, S., and Gao, B. (1999) 'The Adaptive Mesh Model: A New Approach to Efficient Option Pricing', *Journal of Financial Economics,* Vol. 53, pp. 313-351.

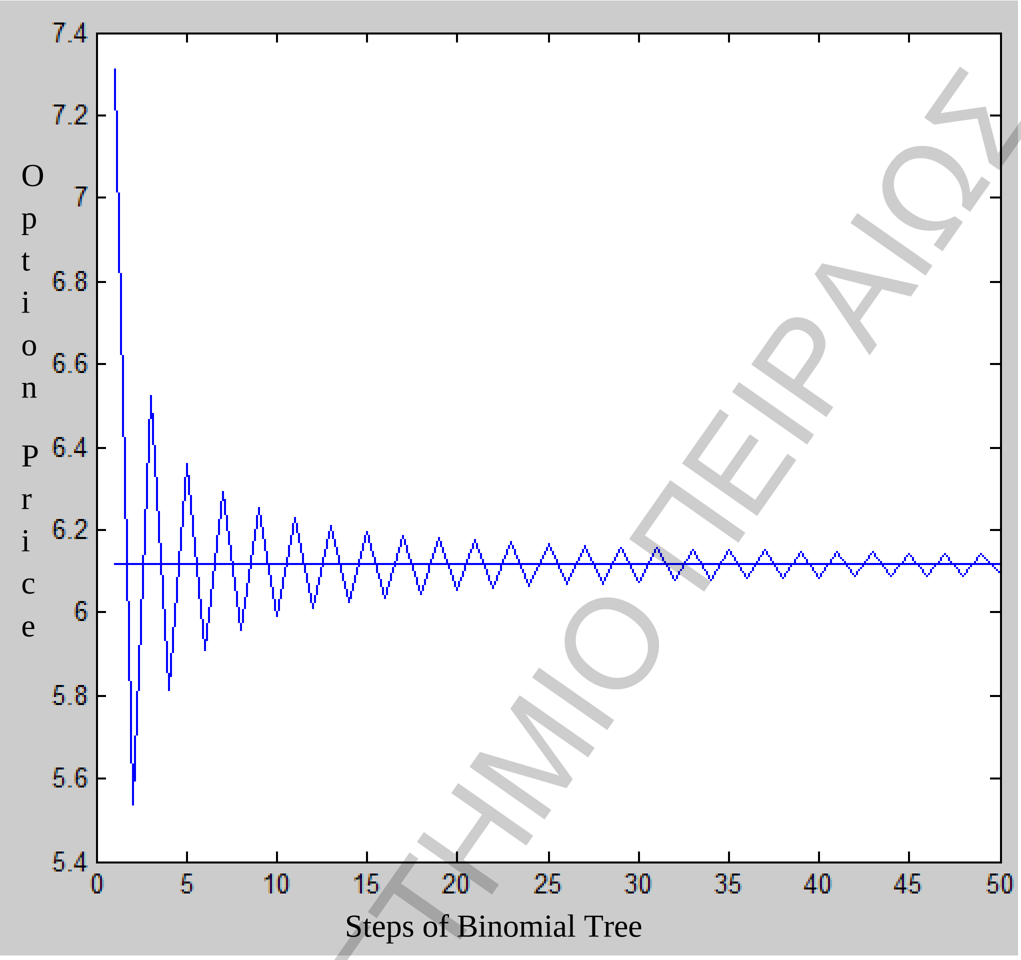
16. Fletcher, C. (1991) *Computational Techniques for Fluid Dynamics: Fundamental and General Techniques*, Springer, Berlin.
17. Garman, M. and Kohlhagen, S. (1983) 'Foreign Currency Option Values', *Journal of International Money and Finance*, Vol. 2, pp. 231-237.
18. Gastineau, G., Kritzman, M. (1996) *Dictionary of Financial Risk Management*, New Hope, Pennsylvania.
19. Geske, R., Johnson, H. (1984) 'The American Put Option Valued Analytically', *Journal of Finance*, Vol. 39, pp. 1511-1524.
20. Geske, R. (1977) 'The Valuation of Corporate Liabilities as Compound Options', *Journal of Financial and Quantitative Analysis*, Vol. 12, pp. 541-552.
21. Heynen, R. C. and Kat, H. (1996) 'Discrete Partial Barrier Options with a Moving Barrier', *Journal of Financial Engineering*, Vol. 5, pp. 199-209.
22. Hull, J. (2002) *Options, Futures and Other Derivatives*, Upper Saddle River, New Jersey.
23. Hull, J. and White, A. (1990) 'Valuing Derivative Securities Using the Explicit Finite Difference Method', *Journal of Financial and Quantitative Analysis*, Vol. 25, pp. 87-100.
24. Hull, J. and White, A. (1995) 'The Impact of Default Risk on the Prices of Options and Other Derivative Securities', *Journal of Banking and Finance*, Vol.19, pp. 299-322.
25. James, R. and James, E. (1982) *Mathematical Dictionary*, Van Nostrand Reinhold, New York.
26. Kamrad, B. and Ritchken, P. (1991) 'Multinomial Approximating Models for Options with k-State Variables', *Management Science*, Vol. 37,pp. 1640-1652.
27. Merton, R. C. (1973) 'Theory of Rational Option Pricing', *Bell Journal of Economics and Management Science*, Vol. 4, pp. 141-183.
28. Moler, C. (2004) *Numerical Computing with MATLAB*, Mathworks, New York.
29. Parkinson, M. (1977) 'Option Pricing: The American Put' *Journal of Business*, Vol. 50, pp. 21-36.
30. Pelsser, A. and Vorst, T. (1994) 'The Binomial Model and the Greeks'. *The Journal of Derivatives*, Vol. 1, pp. 45-49.
31. Reiner, E., and Rubinstein, M. (1991) 'Breaking Down the Barriers', *Risk*, Vol. 4, pp. 28-35.

32. Ritchken, P. (1995) 'On Pricing Barrier Options', *Journal of Derivatives*, Vol. 3, pp. 19-28.

33. Wei, J. (1998) 'Valuation of Discrete Barrier Options by Interpolations', *Journal of Derivatives*, Vol. 6, pp. 51-73.

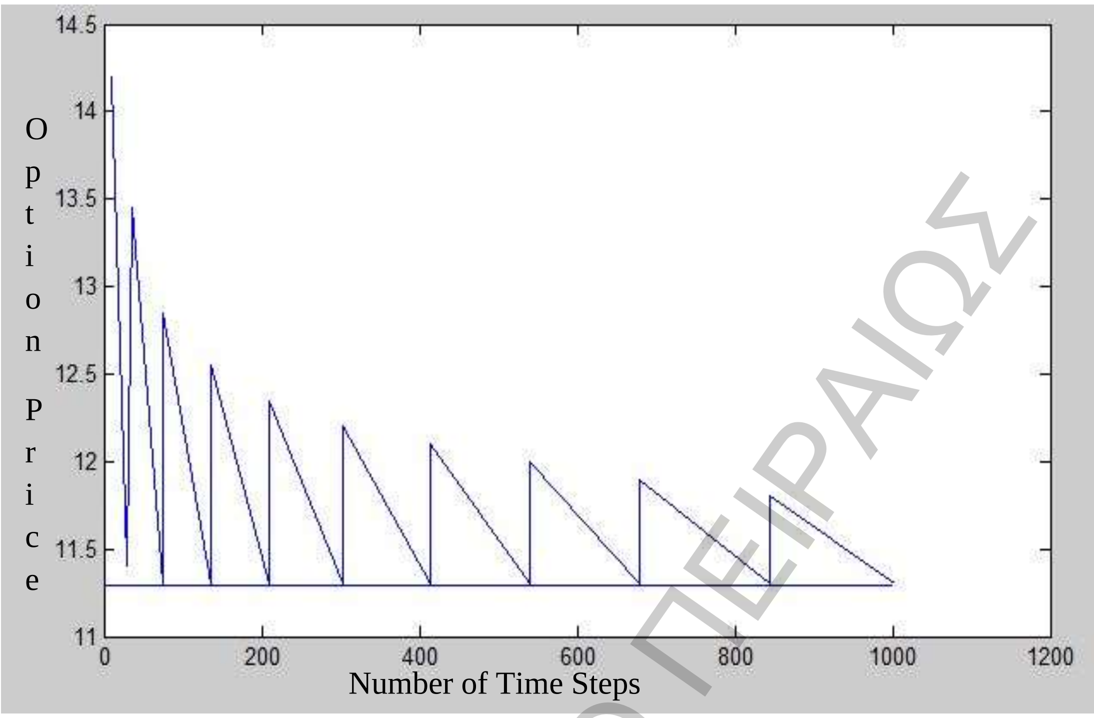
DIAGRAMS

Diagram 2.1



The constant value is the theoretical valuation of the option and the oscillated line is the binomial model's approximation.

Diagram 3.1



The constant value is the theoretical option value computed using equation (3.1) and the oscillated line is the trinomial model's approximation of the option value for different values of time steps.

TABLES

Table 2.1

Approximation Model	Time Steps	Price RMSE	Number of Nodes
Binomial	25	0.020841	351
Trinomial	25	0.012025	676
AMM 1	25	0.002812	716
AMM 2	25	0.000615	756
Binomial	100	0.004929	5151
Trinomial	100	0.002770	10201
AMM 1	100	0.000600	10241
AMM 2	100	0.000151	10281
Binomial	250	0.002214	31626
Trinomial	250	0.001360	63001
AMM 1	250	0.000245	63041
AMM 2	250	0.000057	63081
Binomial	1000	0.000448	501501
Trinomial	1000	0.000244	1002001
AMM 1	1000	0.000056	1002041
AMM 2	1000	0.000016	1002081

We compare an average of 27 European put option valuations through different models against the Black and Scholes valuation. These models are Binomial, Trinomial, Trinomial with one level of fine mesh (AMM 1) and Trinomial with two levels of fine mesh (AMM 2). Further analysis is presented in Subsection 2.3.3.

Table 3.1

Approximation Model	N=25	N=100	N=400
Standard Trinomial	676	10,201	160,801
AMM M=1	926	11,201	164,801
Equivalent Trinomial	10,816	163,216	2,572,816
AMM M=2	1,926	15,201	180,801
Equivalent Trinomial	173,056	2,611,456	41,165,056
AMM M=3	5,926	31,201	244,801
Equivalent Trinomial	2,768,896	41,783,296	658,640,896
AMM M=4	21,926	95,201	500,801
Equivalent Trinomial	44,302,336	668,532,736	10,538,254,336

Comparison of the number of nodes in a barrier option AMM versus a standard trinomial model with the same size of price step as in the finest level AMM mesh.

Table 4.1

N	Approximation Model	Delta RMSE	Gamma RMSE	Execution time (s)
25	**Trinomial ε = 0.001**	0.034958	0.499567	0.0200
	Trinomial ε = 0.01	0.020866	0.101378	0.0200
	Trinomial tree extension	0.003337	0.000428	0.0100
100	**Trinomial ε = 0.001**	0.015642	0.242340	0.2810
	Trinomial ε = 0.01	0.006151	0.044298	0.2800
	Trinomial tree extension	0.000846	0.000144	0.0920
250	**Trinomial ε = 0.001**	0.009286	0.242539	1.5820
	Trinomial ε = 0.01	0.001689	0.026351	1.5830
	Trinomial tree extension	0.000346	0.000061	0.5298
1000	**Trinomial ε = 0.001**	0.004631	0.120938	24.6160
	Trinomial ε = 0.01	0.000656	0.004602	24.6650
	Trinomial tree extension	0.000079	0.000015	8.1918

Performance of the trinomial model in estimating delta and gamma for European put options. The same set of 27 European put options used in Table 2.1 is used as illustrated in Figlewski and Gao (1999).

Table 4.2

Approximation Model	N	t = 0	t = T	Delta RMSE	Gamma RMSE	Execution Time (s)
Trinomial	25	0	0	0.003337	0.000428	0.0090
		1	0	0.001087	0.000333	0.0090
		2	0	0.000810	0.000400	0.0090
		1	1	0.000845	0.000080	0.0120
		2	2	0.000205	0.000113	0.0131
		3	3	0.000053	0.000120	0.0140
	100	0	0	0.000846	0.000144	0.0931
		1	0	0.000265	0.000068	0.0922
		2	0	0.000188	0.000077	0.0931
		1	1	0.000210	0.000020	0.0942
		2	2	0.000056	0.000028	0.0971
		3	3	0.000014	0.000027	0.0991
	250	0	0	0.000346	0.000061	0.5358
		1	0	0.000115	0.000029	0.5288
		2	0	0.000085	0.000031	0.5287
		1	1	0.000079	0.000006	0.5308
		2	2	0.000023	0.000010	0.5878
		3	3	0.000005	0.000011	0.5738
	1000	0	0	0.000079	0.000015	8.5072
		1	0	0.000021	0.000006	8.3711
		2	0	0.000016	0.000007	8.3640
		1	1	0.000023	0.000002	8.3170
		2	2	0.000005	0.000003	8.3130
		3	3	0.000001	0.000003	8.3751

Performance of trinomial AMM models in estimating delta and gamma for European puts. More details in Subsection 4.2.2 as illustrated in Figlewski and Gao (1999).

MATLAB ALGORITHMS

Algorithm for diagram 2.1

```
So=50; K=50; r=0.1; sigma=0.4; T=5/12; N=50;
BlsC=blsprice(So,K,r,T,sigma);
for i=1:N
        LatticeC(i)=LatticeEurCall(So,K,r,T,sigma,i);
end
plot(1:N,ones(1,N)*BlsC);
hold on
plot(1:N,LatticeC);

function [price,lattice] = LatticeEurCall(So,K,r,T,s,N)
DT=T/N;
u=exp(s*sqrt(DT));
d=1/u;
p=(exp(r*DT)-d)/(u-d);
lattice=zeros(N+1,N+1);
for i=0:N
    lattice(i+1,N+1)=max(So*(u^i)*(d^(N-i))-K,0);
end
for j=N-1:-1:0
    for i=0:j
        lattice(i+1,j+1)=exp(-r*DT)*(p*lattice(i+2,j+2)+(1-
p)*lattice(i+1,j+2));
    end
end
price=lattice(1,1);
end
```

European Call Valuation through binomial model

Algorithm for diagram 3.1

```
So=100; K=100; r=0.1; sigma=0.4; T=1; N=1000; H=90;
a=(H/So)^(2*(r-((sigma^2)/2)));
b=((H^2)/So);
BlsC=blsprice(So,K,r,T,sigma)-a*blsprice(b,K,r,T,sigma);
for i=1:N
        LatticeC(i)=TriCDO(So,K,r,T,sigma,i,H);
end
plot(1:N,ones(1,N)*BlsC);
hold on
plot(1:N,LatticeC);
```

```
function [price,lattice] = TriCDO(So,K,r,T,s,N,H)
k=T/N;
h=s*sqrt(3*k);

a=r-(s^2)/2;

lattice=zeros(2*N+1,N+1); u=exp(h);

d=1/u;

pu=sqrt(k/(12*s^2))*a+1/6;
pd=-sqrt(k/(12*s^2))*a+1/6;
pm=2/3;

for i=0:2*N
    if So*exp((i-N))> H
    lattice(i+1,N+1)=max(So*exp((i-N)*s*sqrt(3*k))-K,0
else
    lattice(i+1,N+1)=0;
    end
end

for j=N-1:-1:0
    for i=0:2*j
        if So*exp((i-j))> H
```

```
        lattice(i+1,j+1)=exp(-
r*k)*(pu*lattice(i+3,j+2)+pm*lattice(i+2,j+2)+pd*lattice(i+1,j+2));
        else
        lattice(i+1,j+1)=0;
        end
    end
end
price=lattice(1,1);
end
```

Down and out call barrier valuation through trinomial model

Algorithm for diagram 3.1

```
stpr=[35,40,45];
mada=[1/12,4/12,7/12];
vol3=[0.20,0.30,0.40];

l=1;

for i=1:3
    for j=1:3
        for k=1:3
            data(l,1)=stpr(i);
            data(l,2)=mada(j);
            data(l,3)=vol3(k);
            l=l+1;
        end
    end
end
l=1;
So=40;
r=0.05;
N=[25,100,250,1000];
for i=1:3
    for j=1:3
        for k=1:3
            for ni=1:4

BinVal(l,ni)=LatticeEurPut(So,data(l,1),r,data(l,2),data(l,3),N(ni))
;

TriVal(l,ni)=TriPut(So,data(l,1),r,data(l,2),data(l,3),N(ni));

[dummy,Bls(l,ni)]=blsprice(So,data(l,1),r,data(l,2),data(l,3));
            end
            l=l+1;
        end
    end
```

```
end

for i=1:4
    rmsebin(i)=sqrt((mean(BinVal(:,i)-Bls(:,i))^2));
    rmsetri(i)=sqrt((mean(TriVal(:,i)-Bls(:,i))^2));
end


function [price,lattice] = LatticeEurPut(So,K,r,T,s,N)
DT=T/N;
u=exp(s*sqrt(DT));
d=1/u;
p=(exp(r*DT)-d)/(u-d);
lattice=zeros(N+1,N+1);
for i=0:N
    lattice(i+1,N+1)=max(K-So*(u^i)*(d^(N-i)),0);
end
for j=N-1:-1:0
    for i=0:j
        lattice(i+1,j+1)=exp(-r*DT)*(p*lattice(i+2,j+2)+(1-
p)*lattice(i+1,j+2));
    end
end
price=lattice(1,1);
end


function [price,lattice] = TriCall(So,K,r,T,s,N)
k=T/N;
h=s*sqrt(3*k);

a=r-(s^2)/2;
lattice=zeros(2*N+1,N+1);
u=exp(h);
d=1/u;
pu=sqrt(k/(12*s^2))*a+1/6;
pd=-sqrt(k/(12*s^2))*a+1/6;
pm=2/3;

for i=0:2*N
```

```
    lattice(i+1,N+1)=max(K-So*exp((i-N)*s*sqrt(3*k)),0);
end
for j=N-1:-1:0
    for i=0:2*j
        lattice(i+1,j+1)=exp(-
r*k)*(pu*lattice(i+3,j+2)+pm*lattice(i+2,j+2)+pd*lattice(i+1,j+2));
    end
end
price=lattice(1,1);
end

function [price,lattice] = AMM1ordput(So,K,r,T,s,N)
k=T/N;
h=s*sqrt(3*k);

a=r-(s^2)/2;
lattice=zeros(2*N+1,N+1);
u=exp(h);
d=1/u;
pu=sqrt(k/(12*s^2))*a+1/6;
pd=-sqrt(k/(12*s^2))*a+1/6;
pm=2/3;

flag=0;

for i=0:2*N
    if flag==0
        if So*exp((i-N)*s*sqrt(3*k))>K
            flag=1;
            fia=i+1;
            num=8;
        end
        if So*exp((i-N)*s*sqrt(3*k))==K
            flag=1;
            fia=i+1;
            num=7;
        end
    end
    lattice(i+1,N+1)=max(K-So*exp((i-N)*s*sqrt(3*k)),0);
end
```

```
for i=1:num
    A(i)=lattice(fia-5+i,N+1);
end
B=zeros(1,num);
B=AMM(A,So,K,r,k/4,s);
for i=0:fia-6
    lattice(i+1,N)=exp(-
r*k)*(pu*lattice(i+3,N+1)+pm*lattice(i+2,N+1)+pd*lattice(i+1,N+1));
end
for i=1:num-4
    lattice(fia-6+i,N)=B(i);
end
for i=fia-6+num-4+1:2*(N-1)
    lattice(i+1,N)=exp(-
r*k)*(pu*lattice(i+3,N+1)+pm*lattice(i+2,N+1)+pd*lattice(i+1,N+1));
end

for j=N-2:-1:0
    for i=0:2*j
        lattice(i+1,j+1)=exp(-
r*k)*(pu*lattice(i+3,j+2)+pm*lattice(i+2,j+2)+pd*lattice(i+1,j+2));
    end
end
price=lattice(1,1);
end


function [B]=AMM(A,So,K,r,k,s)
[dummy,sized]=size(A);
step=(-A(2)+A(1))/2;
h=s*sqrt(3*k);
a=r-(s^2)/2;
u=exp(h);
d=1/u;
pu=sqrt(k/(12*s^2))*a+1/6;
pd=-sqrt(k/(12*s^2))*a+1/6;
pm=2/3;

if sized==8
    amml=zeros(15,4);
    for i=1:15
```

```
        amml(i,4)=A(1)+i*step
    end
    for j=6:-1:4
    for i=0:2*j
        amml(i+1,j-3)=exp(-r*k)*(pu*amml(i+3,j-2)+pm*amml(i+2,j-
2)+pd*amml(i+1,j-2));
    end
    end
    for i=0:3
        B(i+1)=exp(-
r*k)*(pu*amml(2*i+3,1)+pm*amml(2*i+2,1)+pd*amml(2*i+1,1))
    end
end
if sized==7
    amml=zeros(13,4);
    for i=1:13
       amml(i,4)=A(1)+i*step;
    end
    for j=5:-1:3
    for i=0:2*j
        amml(i+1,j-2)=exp(-r*k)*(pu*amml(i+3,j-1)+pm*amml(i+2,j-
1)+pd*amml(i+1,j-1));
    end
    end
    for i=0:3
        B(i+1)=exp(-
r*k)*(pu*amml(2*i+3,1)+pm*amml(2*i+2,1)+pd*amml(2*i+1,1));
    end
end
end
```

European put option valuation through binomial, trinomial, AMM1, AMM2 valuation