Numerical Valuation of Discrete Barrier Options with the Adaptive Mesh Model and Other Competing Techniques

Advisor: Prof. Yuh-Dauh Lyuu

Chih-Jui Shea

Department of Computer Science and Information Engineering
National Taiwan University

Abstract

This thesis develops an Adaptive Mesh Model for pricing discrete double barrier options. Adaptive Mesh Model is a kind of trinomial tree lattice that applying higher resolution to where nonlinearity errors occur. After the Adaptive Mesh Model for discrete single barrier options was proposed in 1999 by Ahn, Figlewski, and Gao, there is no further research has been done in Adaptive Mesh Model for discrete barriers. Furthermore, numerical data are also scarce in the paper of Ahn et al.. This thesis bases on the lattice structure of Ahn et al. and extends the Adaptive Mesh Model to price discrete double barrier options. Besides, there is no close-form solution for discrete barrier options such that many methods have been suggested and declared to price discrete barrier options fast and accurately but no one can tell exactly that what method is the best. We also make a complete comparisons of the Adaptive Mesh Model with other methods no matter in accuracy or in efficiency. Our numerical data shows that the Adaptive Mesh Model is generally surpassed the other tree lattice methods and the BGK formula approach, and exceed the quadrature method in efficiency with accurate enough outcomes.

Keywords: Adaptive Mesh, numerical valuation techniques, discrete barrier options, double barrier options, trinomial trees, enhanced trinomial trees, BGK model, quadrature method, option pricing

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Barrier options have become more and more popular. They are not only desirable in speculation but also in risk management because of lower costs than their plain vanilla counterparts. The typical analytic pricing formulas for single barrier options are derived assuming continuously monitoring of the barrier. However, in real market barrier conditions of options are generally monitored discretely but there is no close-form solution. Many numerical methods have been proposed to price discrete monitored barrier options including the Adaptive Mesh Model. Since the Adaptive Mesh Model for pricing discrete single barrier options is first proposed in 1999 [14], the concept of adaptive mesh has been widely discussed but further research is absence. Also, numerical results of the Adaptive Mesh Model is rare in the original paper. Hence, in this thesis we do not only implement the Adaptive Mesh Model of Ahn et al. but also extend it to price discrete double barrier options. Besides, we compare the Adaptive Mesh Model to other competing methods with extensive numerical data both in efficiency and accuracy.

In Chapter 1 and Chapter 2, we shortly set the background and the concept of barrier options. Chapter 3 introduces the Adaptive Mesh Model starting from two kinds of approximation errors (i.e. distribution error and nonlinearity error) generally existing in lattice models and then using Adaptive Mesh Model to ease the nonlinearity error in both European and American puts. In the latter part of Chapter 3, we propose the Adaptive Mesh Model for pricing not only single but also discrete double barrier options. At Last in Chapter 4 we compare the Adaptive Mesh with other competing methods in pricing discrete barrier options numerically and end up with the conclusions in Chapter 5.

Chapter 2

Barrier Options

2.1 Barrier Option Basics

A barrier option is a kind of path-dependent options that comes into existence or is terminated depending on whether the underlying asset's price S reaches a certain price level H called "barrier". A knock-out option ceases to exist if the underlying asset reaches the barrier, whereas a knock-in option is activated if the barrier is reached by underlying asset. According to the relative position of H and S, there are four kinds of typical barrier, which are outlined below.

1. Down and Out: knock-out options with $H < S$.

2. Down and In: knock-in options with $H < S$.

3. Up and Out: knock-out options with $H > S$.

4. Up and In: knock-in options with $H > S$.

Besides, based on how frequently the barrier condition is checked, one barrier can be continuous or discrete. Once a continuously monitored barrier is reached the option is immediately knocked in or out, while in discretely monitored conditions, barriers only come into effect in those monitored time, e.g. close of every market day, every quarter, every month, or every half year.

Barrier options have become quite popular especially in the foreign exchange markets. One of the barrier option's advantage is its cheaper price. Take a down-and-out barrier call option for example, a trader with a bull perspective view on the market may regard the condition of the barrier being reached as quite unlikely and be more interested in it than the regular one. Or a hedger may buy a barrier contract to hedge a position with a natural barrier, e.g. the foreign currency exposure on a deal that will take place only if the exchange rate remains above a certain level.

2.2 Pricing of Barrier Options

Barrier options were first traded on the OTC market in the late 60s, but the first analytical formula for a down and out call option was proposed by Merton (1973) [1] which was followed by the more detailed paper by Reiner & Rubinstein (1991) [2] providing the formulas for all 4 types of barrier on both call and put options. However, the analytic formulas mentioned above only present methods to price barrier options in continuous time, but often in the market, the asset price is discretely monitored. In other words, they specify fixed times for monitored of the barrier.

Although discretely monitored barrier options are popular and important, pricing them is not as easy as their continuous counterparts. There is essentially no closed solution, except using m-dimensional normal distribution function (m is the number of monitored points), which can hardly be computed easily if, for example, $m > 5$ (see Reiner (2000) [3] and closed-form valuation equations for discrete barrier options in Heynen and Kat (1996) [4]). When it comes to Direct Monte Carlo simulation, it takes too much time to produce accurate enough results.

To deal with these difficulties, Broadie, Glasserman and Kou (1997) propose a continuity correction for discretely monitored barrier options, and justify the correction both theoretically and numerically. They adjust the barrier in the closed-form equations of continuous barrier options to account for discrete sampling as follows:

$$H' = He^{\alpha\sigma\sqrt{\frac{T}{m}}}$$

It is so-called BGK barrier adjustment model. For up-barrier options, the value of α is 0.52826, whereas for down-barrier options, the value of α is -0.5826, where m is the number of times the underlying asset price is monitored over the time period T [5].

Like most other path-dependent models, barrier options can be priced by tree lattice techniques such as binomial or trinomial by solving the PDE using a generalized finite difference method. However, even in continuously monitored barrier options the convergence of lattice approach is very slow and require a quite large number of time steps to obtain a reasonably accurate result. It is because the barrier being assumed by the tree is different from the true barrier. Define the inner barrier as the barrier formed by nodes just on the inside of the true barrier and the outer barrier as the barrier formed by nodes just outside the true barrier. Fig. 2.1 shows the inner and outer barrier for a binomial and trinomial tree when the true barrier is horizontal and constantly monitored. The usual tree calculations implicitly assume the outer barrier is the true barrier because the barrier condition is first met on the outer barrier.

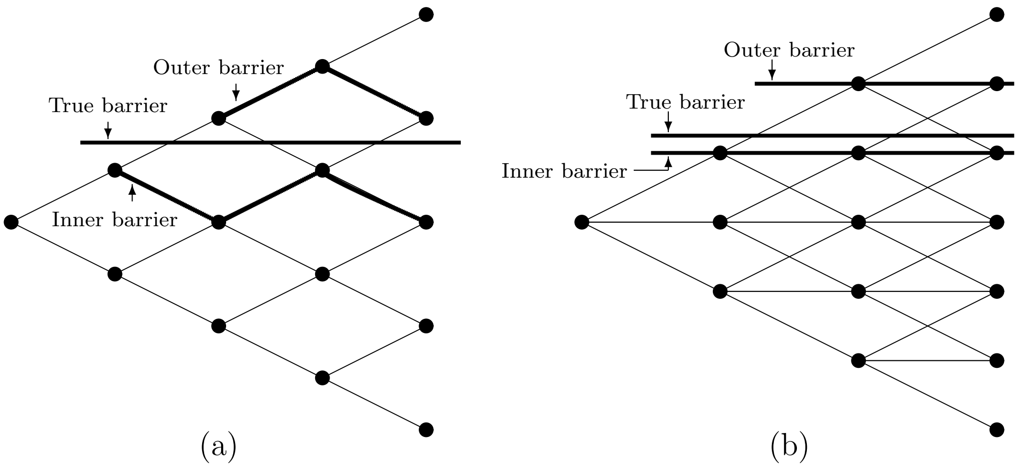


Figure 2.1: **Barrier assumed by tree lattice**
(a) Barriers assumed by binomial trees. (b) Barriers assumed by trinomial trees.

Bolye and Lau [6] describe this condition and propose a method to constrain the time steps that make the true barrier coincide with or occur just above the underlying asset price level in trees. Nevertheless, the time step constraint makes the lattice impracticable to compute because of the incredible large number of time steps when the initial asset price is too close to the barrier. On the other hand, the constraint of time step number is also annoying. In 1995, Derman et al. propose an adjusting for nodes not lying on barriers by assuming the barrier calculated by the tree is incorrect[8]. Ritchken (1995) [9] offers another approach under trinomial framework introducing a "stretch" parameter into the lattice, which changes the price step just enough to place nodes on the barrier. Cheuk and Vorst [10] also introduce a deformation of the trinomial tree permitting one to adjust the location of nodes differently in each time period, and allows great flexibility in matching a time-varying barrier. Although those methods have been proposed, a quite slow convergence rate still occur when they are used to price discretely monitored barrier options.

For pricing discrete barrier options, Wei (1998) [12] offers an approximation approach based on interpolating between the formula for a barrier option with the highest number of monitored points that can be handled with the analytic formula and the continuous case (infinite monitored dates). Broadie, Glasserman and Kou (1999) develop the *enhanced trinomial model* from Ritchken's lattice framework. Like their earlier paper in 1997 [5], they shift the discrete barrier at level H to a new barrier at level $H' = He^{\pm 0.5\lambda^* \sigma \sqrt{h}}$ (with + for an up option and - for a down option), where $\lambda^* \triangleq \sqrt{3/2}$ and h is the size of one time step [11]. Both these techniques, however, can be used only for European options, and in Broadie et al.'s model, the "barrier-too-close" problem still exists.

Figlewski and Gao (1999) [13] present the adaptive mesh model (AMM) as an efficient trinomial lattice approach to deal with "barrier-too-close" problem in continuous barrier options. Furthermore, in the same year, an another kind of adaptive mesh model is proposed for pricing discrete barrier options by Ahn, Flglewski, and Gao [14]. The AMM model is very powerful in both efficiency and flexibility and is going to be discussed further in this thesis.

Besides, there is the quadrature method presented by Andricopoulos et al. (2003) [15] using somewhat multinomial-like integral method to price discrete barrier options with speed and accuracy which can also deal with barrier-too-close problem. We will numerically compare it with the AMM model later.

Chapter 3

The Adaptive Mesh Model

3.1 Approximation Error in Lattice Models

Although lattice models provide powerful, intuitive and asymptotically exact approximations to the theoretical option values under Black-Scholes assumptions, there are essentially two related but distinct kinds of approximation errors in any pricing techniques of lattice framework, which we refer to as distribution error and nonlinearity error, where the latter can be minimized by the adaptive mesh model with slight computation increase.

1. **Distribution error**: The lattice model approximates the true asset price distribution with continuous lognormal density by a finite set of nodes with probabilities. Even though the mean and variance of the continuous distribution are matched by the discrete distribution of lattice model, the discrepancy between discrete and continuous distribution still produces distribution error in option value.

2. **Nonlinearity error**: The finite set of nodes with probabilities used by lattice model can be thought as a set of probability weighted average option price over a range of the continuous price space around the node. If the option payoff function is highly nonlinear, evaluating the nonlinear region with only one or several nodes would gives a poor approximation to the average value over the whole interval.

Fig. 3.1 illustrates the two sources of error graphically around at the money nodes of a one year European put at expiration date with the initial asset price $S_0 = 100$, the exercise price $X = 100$, riskless rate $r = 0.1$ and volatility $\sigma = 0.25$. The solid line represents the option payoff. The gray shaded bars represent the nodes in the trinomial lattice, corresponding to

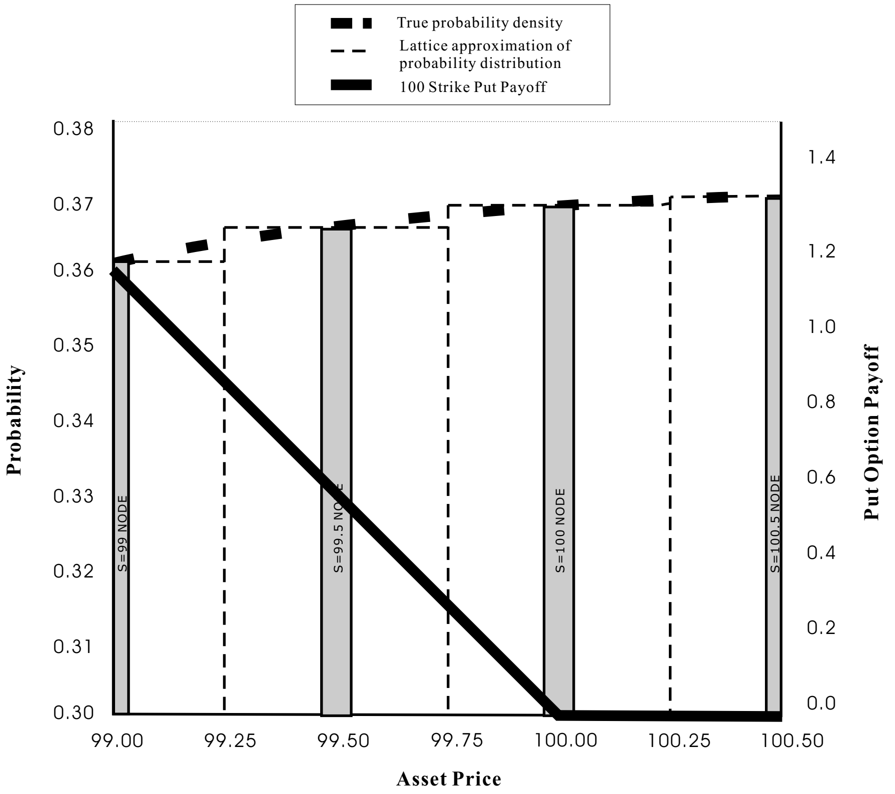


Figure 3.1: **Distribution error and nonlinearity error around the at-the-money nodes at maturity date.**

asset prices of 99.0, 99.5, 100.0, and 100.5. The heavy dashed line represents the lognormal density over this region of the price space. The light dashed bars indicate how the probability density is discretized over this price range. The contribution of a particular node to the option value equals the value of the node probability multiplies the option payoff at the asset price for that node. The distribution error arises from the difference between the heavy dashed line and the light dashed line. At the $S = 100$ node, the nonlinearity error is caused by undervaluing the probability weighted average payoff to zero in this interval [13].

The adaptive mesh model presented in this thesis can significantly reduce the nonlinear error over a given region of the tree.

3.2 Building the Model

Now we start to build a lattice model to price plain vanilla options using adaptive mesh mechanism around the nonlinear payoff region of exercise price at maturity. The essence of the AMM is to use a relatively coarse lattice throughout the option life and insert meshes with higher resolution into the tree where the nonlinear error is contributed. It is important for the fine mesh structure (higher resolution mesh) to be isomorphic so that additional, still finer mesh can be added using the same procedure. This allows increasing resolution in a given region of the lattice as much as one wishes without requiring the step size changes elsewhere.

Here we introduce an isomorphic AMM structure that can be easily applied to each region of the lattice. Trinomial tree is chose as the base lattice to approximate the risk neutral distribution because it has more degrees of freedom and has proven to be more useful and adaptable for many derivative applications. Because the asset price is assumed to be lognormal, the tree is based on the log of asset price S. Define $X = \ln(S)$, which implies that X is normally distributed. Under risk neutral assumption, X follows the standard diffusion process:

$$dX(t) = \alpha dt + \sigma dz$$

where $\alpha = r - q - \sigma^2/2$, σ denotes volatility, dz is standard Brownian motion, and r and q are the riskless interest rate and dividend yield.

In trinomial tree, there are three different branches for any node to move to next time state, which are called up (u), down (d), and middle (m). For deduction's convenience, we change the variable X by $X' = X - \alpha t$. X' is the mean-adjusted value of the log of asset price and the mean of X' would be 0 at any time state. Hence, The trinomial tree of X' is symmetric. Let k denote the length of a time step (decided by the option's maturity T, and the number of time steps N to be used for the tree with $k = T/N$) and h be the size of an up and down move. Thus over one time period X goes to $X + h$ with probability p_u, to $X - h$ with probability p_d, and remain unchanged with probability p_m.

Matching the mean, variance, and summing up all probabilities to be one, there are three constraints must be obeyed by the three next state prices and three probabilities at each node of tree.

$$\begin{aligned} &1 = p_u + p_m + p_d, \\ &E[X'(t+k) - X'(t)] = 0 = p_u h + p_m 0 + p_d(-h), \\ &E[(X'(t+k) - X'(t))^2] = \sigma^2 k = p_u h^2 + p_m 0 + p_d(-h)^2. \end{aligned} \tag{3.1}$$

By solving Eq. (3.1) we can get the following relations:

$$\begin{aligned} p_u &= \frac{\sigma^2 k}{2h^2}, \\ p_m &= 1 - \frac{\sigma^2 k}{h^2}, \\ p_u &= \frac{\sigma^2 k}{2h^2}. \end{aligned} \tag{3.2}$$

Besides, because the tree of X' is symmetric distributed, all odd-numbered moments of the trinomial will be zero, as they are for the normal. Therefore, we can set the kurtosis in the tree equal to that of the normal.

$$E[(X'(t+k) - X'(t))^4] = 3\sigma^4 k^2 = p_u h^4 + p_m 0 + p_d(-h)^4. \tag{3.3}$$

Applying the relations in Eq. (3.2) into Eq. (3.3) for the probabilities yields:

$$\begin{aligned} h &= \sigma\sqrt{3k}, \\ p_m &= 2/3, \\ p_u &= p_d \quad = 1/6. \end{aligned} \tag{3.4}$$

This is the trinomial process approximating the asset price distribution:

$$X'_{t+k} - X'_t = \begin{cases} h, & \text{with probability}\, p_u = 1/6 \\ 0, & \text{with probability}\, p_m = 2/3 \\ -h, & \text{with probability}\, p_d = 1/6. \end{cases}$$

which implies the process of X

$$X_{t+k} - X_t = \begin{cases} \alpha k + h, & \text{with probability}\, p_u = 1/6 \\ \alpha k, & \text{with probability}\, p_m = 2/3 \\ \alpha k - h, & \text{with probability}\, p_d = 1/6. \end{cases} \tag{3.5}$$

The option value at a given asset price and time, $V(X,t)$ is computed from the values at the three successor nodes as:

$$\begin{aligned} V(X,t) = \exp(-rk)[&p_u V(X+\alpha k+h, t+k) + p_m V(X+\alpha k, t+k) \\ &+ p_d V(X+\alpha k-h, t+k)]. \end{aligned} \tag{3.6}$$

Note that for generality Eq. (3.6) allows that the probabilities may vary with h and k, even though in the current case of Eq. (3.5) they are fixed.

3.3 Application of the Adaptive Mesh Model to Plain Vanilla Options

For a European option, nonlinearity error is around the exercise price at expiration. It turns out that an American option's nonlinearity error is also largely accounted for by the error in the last time step, for the prices that bracket the strike price. Besides, for an American option there is also an approximation error with regard to where the early exercise occur. However, by "smooth pasting" property [16] of the American option value, this kind of approximation error does not translate into significant error in valuing option because the values of option price nodes is not highly nonlinear around the early exercise boundary.

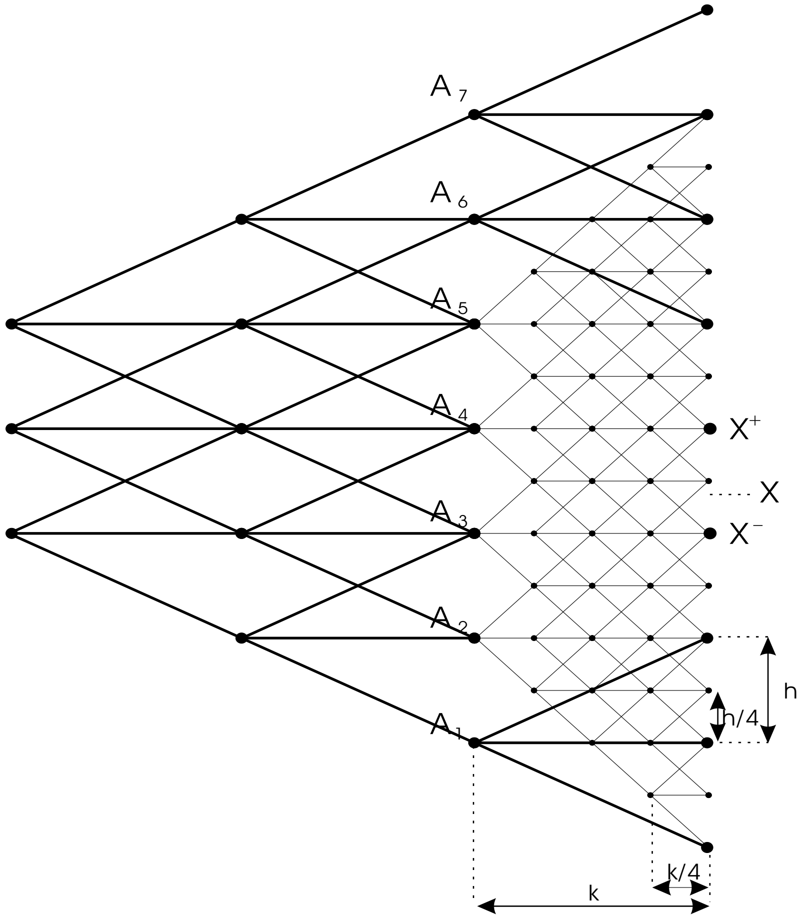


Figure 3.2: **An AMM for a put option around exercise price at expiration.**

While there is already a well-known analytic solution by Black and Sholes for pricing European option, we do not only take an European put option but also take an American put option with AMM mechanism applied around exercise price at maturity date as examples. Fig. 3.2 illustrate the critical region of Adaptive Mesh trinomial tree that we wish to construct to value a put option. The base coarse lattice, with price and time steps h and k, is represented by heavy lines, and light lines represent the finer mesh with price step size $h/2$ and time step size $k/4$. The finer mesh covers all coarse nodes at time state $T-k$, from which there are both fine-mesh paths that end up in-the-money and out of-the money at expiration, i.e. A_2, A_3, A_4, and A_5 in this figure. X is the strike price, and X^+ and X^- are the two date T coarse-mesh node asset price that bracket the strike price. In finer mesh, X^+ is the highest out of-the money node that branches from A_2 whereas X^- is the lowest in-the-money price node from A_2. Since all branches starting from nodes below A_1 all end up in-the-money and paths start from nodes above A_6 are all expired at the end, there is no need to fine the mesh.

The finer mesh is set up with one-half price size of the previous coarser mesh. To cut the price step size in half with maintaining the relationship in Eq. (3.5), the time step price must be set one-quarter of the size of the coarser one. By the isomorphism of AMM, the trinomial tree lattice introduced in Fig. 3.2 can cut into any finer level as one wish in the same manner. Thus, if we set the base mesh as level 0, then the finer mesh of level M has price step size $h_M = h/2^M$ and time step size $k_M = k/2^M$.

In the traditional trinomial tree model, there are $(N+1)^2$ nodes of price computation in total, where N is the number of price steps. Therefore, cutting the price step in half to reduce the nonlinear error makes N become quadrupled (h is proportional to $\sqrt{k}$) which implies 16 times computation amount than before. On the other hand, as we can see in Fig. 3.2 adding one level of adaptive mesh model to cut the nonlinearity error down only needs 40 more nodes of price computation in critical region (9, 11, 13, and 7 for time states $T-3/4k$, $T-2/4k$, $T-1/4k$, and T).

Fig. 3.3 shows the convergence behavior of an in-the-money American put which is priced by Adaptive Mesh Model. The Label of AMM-M means the AMM model of level M. The yellow line represents the Traditional Trinomial Approach, while the blue line is the convergence behavior of Adaptive Mesh Model of level 2. Although there is the approximation error contributed by early exercise, AMM model still can improve the convergence behavior with a little more calculations in American put options. If we rule out the influence of early exercise, it comes to the European put option whose convergence behavior is presented in Fig. 3.4 where we can see that a higher level of AMM model gives rise to a better convergence rate.

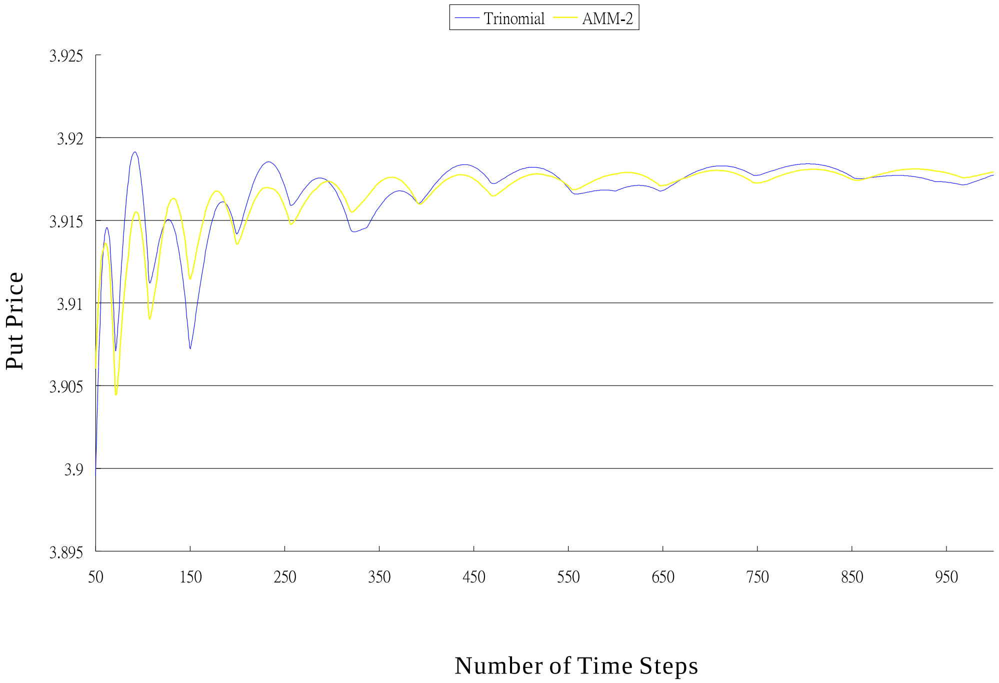


Figure 3.3: **The AMM model convergence for at-the-money American put.**

$S = 100$, $X = 100$, $\sigma = 20\%$, $r = 10\%$, dividend yield $q = 0\%$, and $T = 0.5$ year.

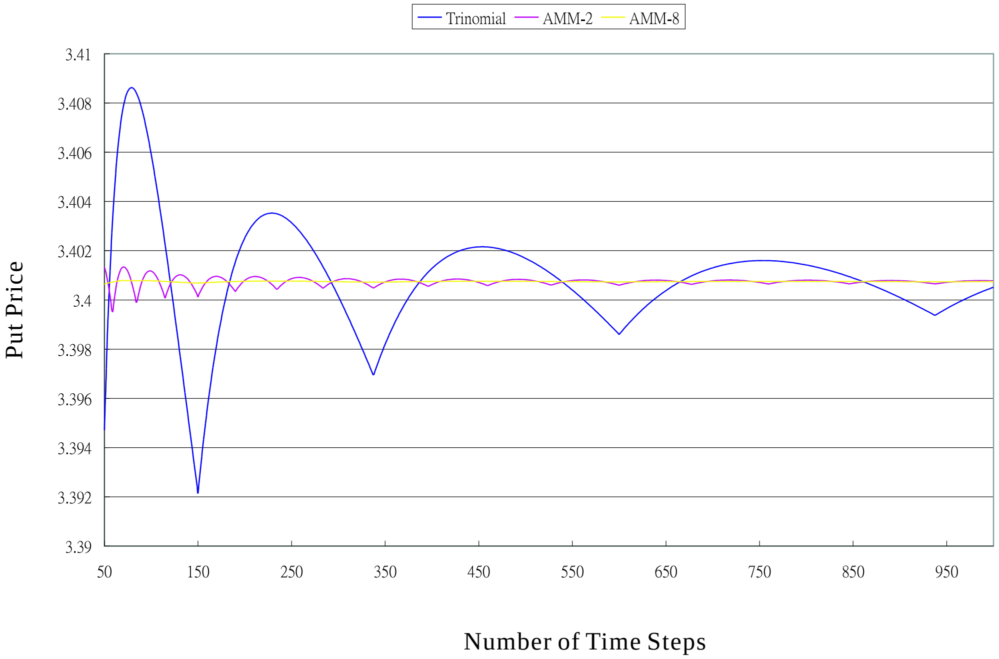


Figure 3.4: **The AMM model convergence for at-the-money European put.**
$S = 100$, $X = 100$, $\sigma = 20\%$, $r = 10\%$, dividend yield $q = 0\%$, and $T = 0.5$ year.

3.4 Extending the AMM Model to Discrete Single Barrier Options

With only a little modification and extension, the model of Fig. 3.2 can be extended to price discretely monitored barrier options. Fig. 3.5 shows how this is done in a discretely monitored down-and-out barrier call by an AMM structure with one level of fine mesh around the barrier. The coarse mesh nodes are labelled as A and the finer mesh nodes are labelled as B. As to the subscript, $A_{j,k}$ node means the k-th coarse mesh price node at time state j. The lattice before time state $j+1$ are of the same structure as Fig. 3.2 with only the exercise price is replaced by the barrier price at time $j+1$. Lattice between time states $j+1$ and $j+2$ connect the fine mesh lattice back to the coarse lattice.

There are two kinds of B-level nodes at time state $j+1$. Some are at the same positions as A-level nodes, and the other are between two coarse nodes. If we directly connect all B-level nodes at time $j+1$ to all A-level nodes at $j+2$. The former can intuitively use trinomial method and the latter may use quadrinomial branching mechanism such as in [13]. But when we want to cut the mesh finer (i.e. add more level to the tree), it seems like too complicated under this kind of lattice structure. Hence, the mechanism in Fig. 3.5 is presented with isomorphism of adding any finer meshes. The coarse time step is divided into two subperiods. The first subperiod is one fine mesh time step ,and the second is three-quarters of a coarse time step. That is, the first of length $k/4$ and the second of length $3k/4$ in the example lattice in Fig. 3.5.

Branching for the first subperiod is the same as at other B-level nodes. However, it also leads to two kinds of $B_{5,*}$ nodes. For those nodes lying at the same price level of coarse nodes, the trinomial branching method is straightforward. The node values can be obtained from Eq. (3.6) with a price step h and a time step $3k/4$. And the branch probabilities of $p_u = p_d = 1/8$ and $p_m = 3/4$ can be derived from Eq. (3.2) with replacing k with $3k/4$ in probability equations of p_u, p_m, and p_d, and maintaining the relationship $h = \sigma\sqrt{3k}$ (because it has been fixed by coarse mesh). Let $k' = 3k/4$ Hence, the new trinomial process for those nodes is as follows:

$$X_{t+k} - X_{t+k/4} = \begin{cases} \alpha k' + h, & \text{with probability } {p_u}' = 1/8 \\ \alpha k', & \text{with probability } {p_m}' = 3/4 \\ \alpha k' - h, & \text{with probability } {p_d}' = 1/8. \end{cases} \quad (3.7)$$

Notice that the kurtosis is no longer matched by the process in Eq. (3.7) because matching the mean, variance, constraining the all probabilities to be

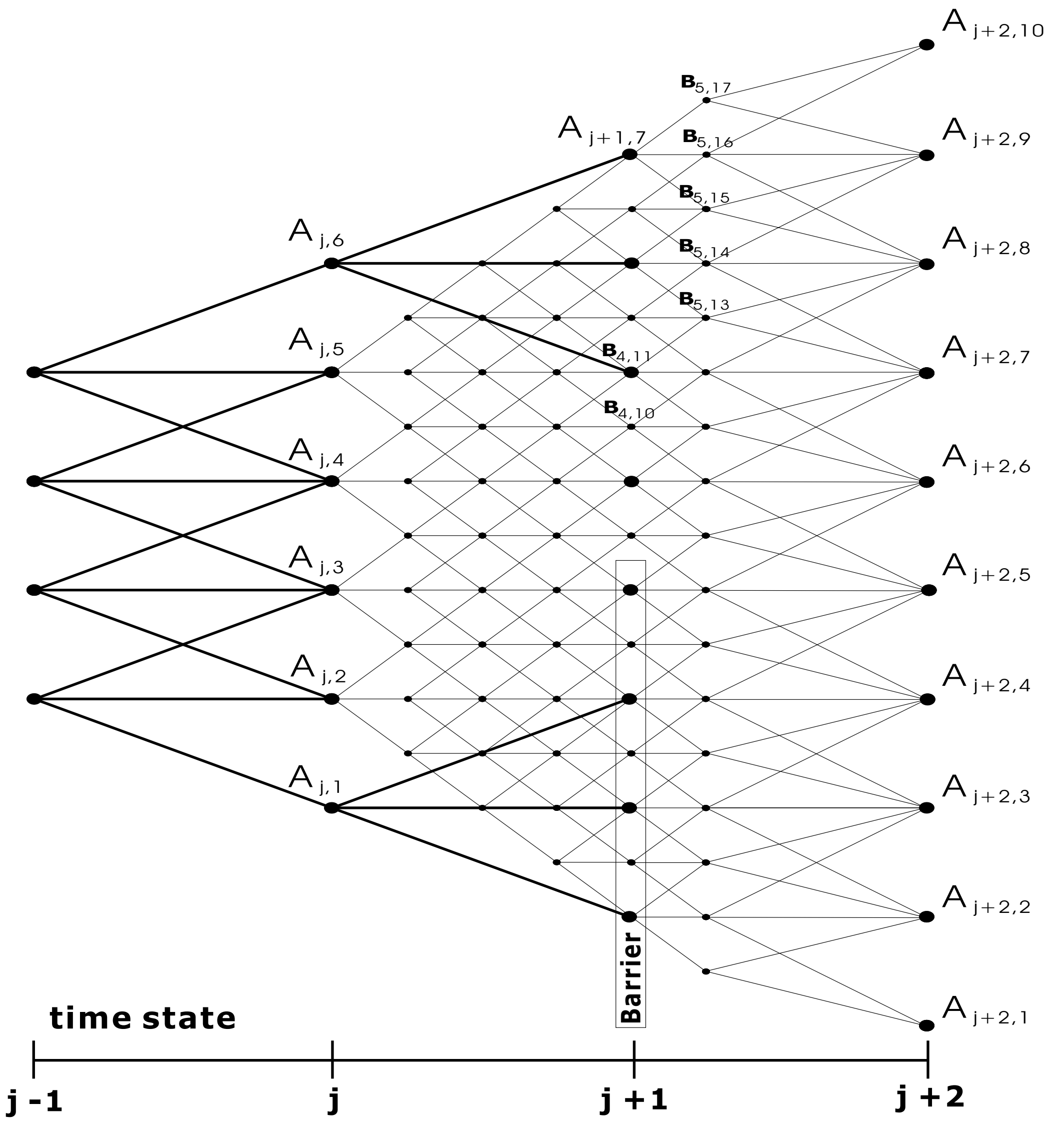


Figure 3.5: **An AMM for discrete down-and-out barrier call options.**

one, and maintaining the relationship of $h = \sigma\sqrt{3k}$ have used four degree of freedom while there are only five variables p_u, p_m, p_d, h, and k.

For the nodes lying between two coarse node price levels, a quadrinomial branching mechanism should be applied. For example, we should connect $B_{5,13}$ to four A-level nodes at time $t+2$, i.e. $A_{j+2,6}$, $A_{j+2,7}$, $A_{j+2,8}$,and $A_{j+2,9}$ with price increments of $3h/2$, $h/2$, $-h/2$, and $-3h/2$. Matching the mean, variance, and adding four branch probabilities to be one give us the following three equations under the condition of $h = \sigma\sqrt{3k}$.

$$\begin{aligned}
&p_{uu} + p_u + p_d + p_{dd} = 1, \\
&p_{uu}(\alpha k' + 3h/2) + p_u(\alpha k' + h/2) + p_d(\alpha k' - h/2) + p_{dd}(\alpha k' - 3h/2) = \alpha k', \\
&p_{uu}(3h/2)^2 + p_u(h/2)^2 + p_d(-h/2)^2 + p_{dd}(-3h/2)^2 = \sigma^2 k'.
\end{aligned} \tag{3.8}$$

which can be solved as:

$$\begin{aligned}
p_{uu} &= p_{dd} = 0, \\
p_u &= p_d = 1/2.
\end{aligned} \tag{3.9}$$

Surprisingly the solution in Eq. (3.9) collapses the quadrinomial branching into binomial one, as follows:

$$X'_{t+k} - X'_{t+1/4} = \begin{cases} \alpha k' + 3\sigma h/2, & \text{with probability}\, p_{uu} = 0 \\ \alpha k' + \ \sigma h/2, & \text{with probability}\, p_u = 1/2 \\ \alpha k' - \ \sigma h/2, & \text{with probability}\, p_d = 1/2 \\ \alpha k' - 3\sigma h/2, & \text{with probability}\, p_{dd} = 0. \end{cases}$$

The isomorphic structure of the fine mesh allows us to add the next layer, with price and time steps $h_C = h/4$ and $k_C = k/16$, using exactly the same procedure as described above. Fig. 3.6 illustrates the resulting lattice structure. As we can see from this figure, we don't care where the barrier is and merely cut the lattice finer by adding more levels into the structure around where the payoff function value is significantly nonlinear. Hence, the "barrier-too-close" problem does not exists under the AMM lattice mechanism.

We can see the convergence behavior of a single discrete Down-and-Out barrier European Call in Fig. 3.7 with $S = 100$, $X = 100$, down-and-out barrier $H = 90$, $\sigma = 20\%$, $r = 10\%$, $q = 0$, and $T = 0.5$ year. The monitoring frequency F of discrete barrier is 6 which means the barrier is checked $0.5year/6$, that is monthly. The number of time steps starts from

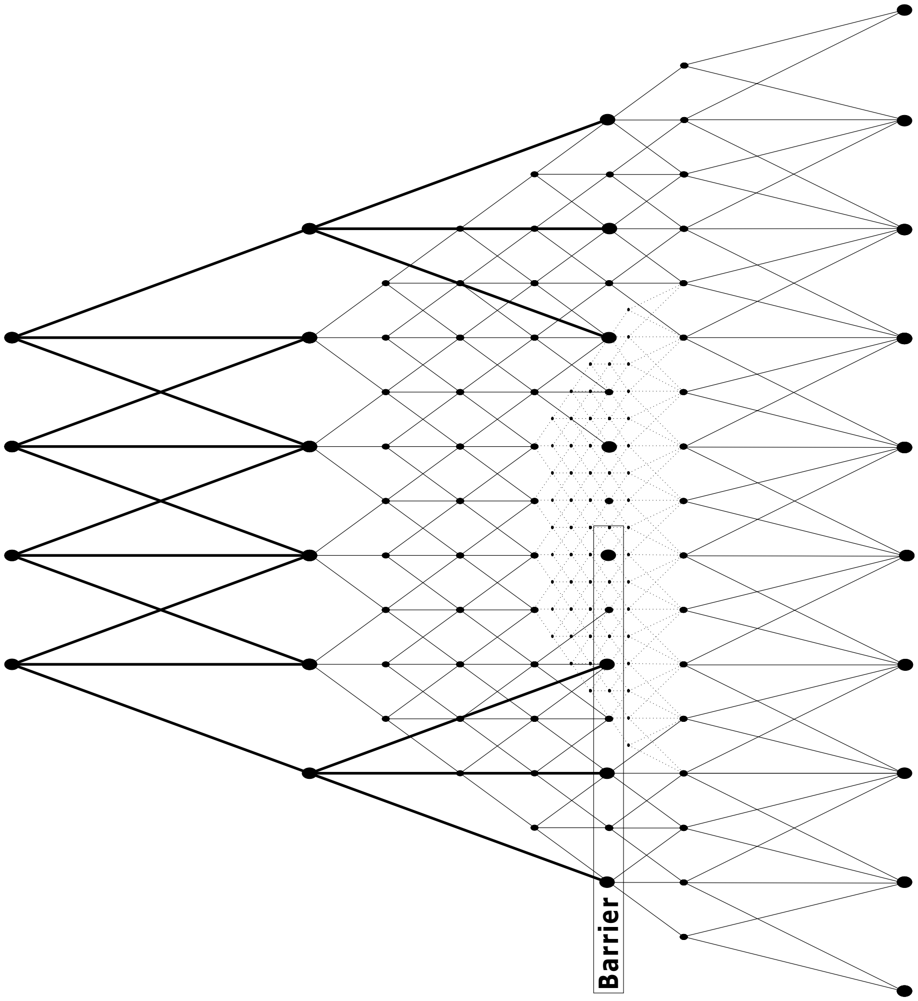


Figure 3.6: **An AMM of level 2 for discrete down barrier options.**

24 and ends with 1200. The blue line with huge zig-zag phenomenon represents the convergence result of the trinomial model. The trinomial model we use here is the same as AMM lattice with AMM level set to be zero. The convergence of AMM level 2 in purple line is also somewhat sawtooth-like. When it comes to level 8, the yellow line looks almost like a straight line. As the result shows, the adaptive mesh model can not only contribute a better convergence rate but also be helpful to eliminate the zig-zag occurrence in convergence behavior when we apply a higher level of AMM mechanism to the lattice model.

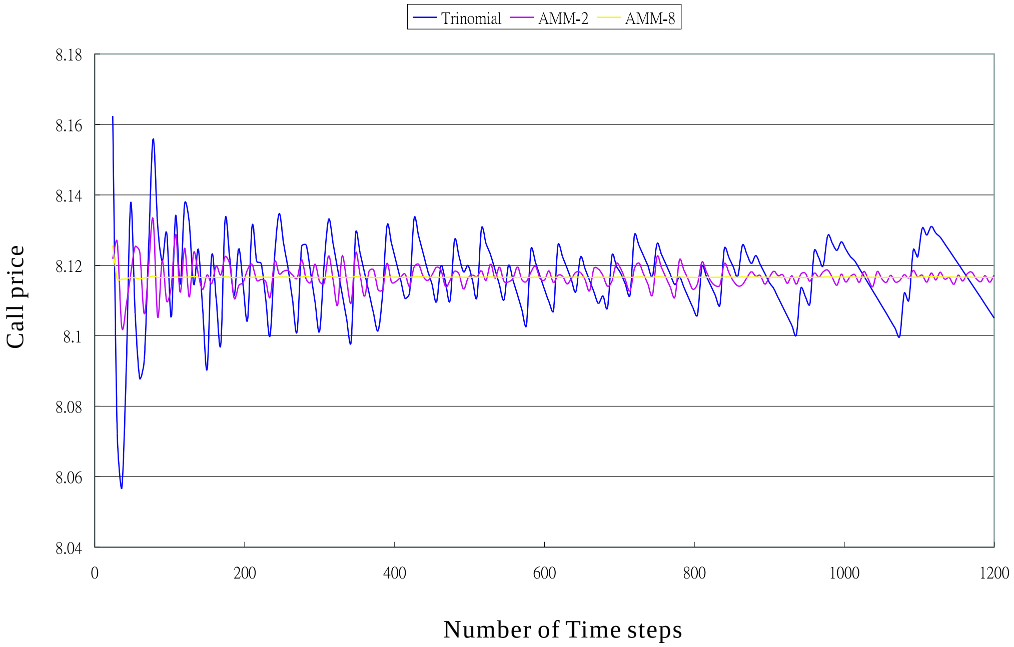


Figure 3.7: **The AMM Model convergence for a single discrete down-and-out barrier European call.**
$S = 100,\ X = 100,\ H = 90,\ \sigma = 20\%,\ r = 10\%$, $q = 0\%$ and monitoring frequency $F = 6$.

3.5 Further Extending to Discrete Double Barrier Options

It is very intuitive for us to extend the AMM structure introduced in previous section to price double discrete barrier options simply by applying the same adaptive mesh technique around both barriers. Fig. 3.8 shows the lattice structure in most of the time. There are two fine mesh area. One is the adaptive mesh of level 1 for high barrier ,and the other is for low barrier. These two fine mesh work individually and will not influence any node's value of each other because there is no intersectional area in them. Moreover, no matter the initial asset price is how close to either barriers, the lattice model in Fig. 3.8 still functions well enough depending on the level of AMM model.

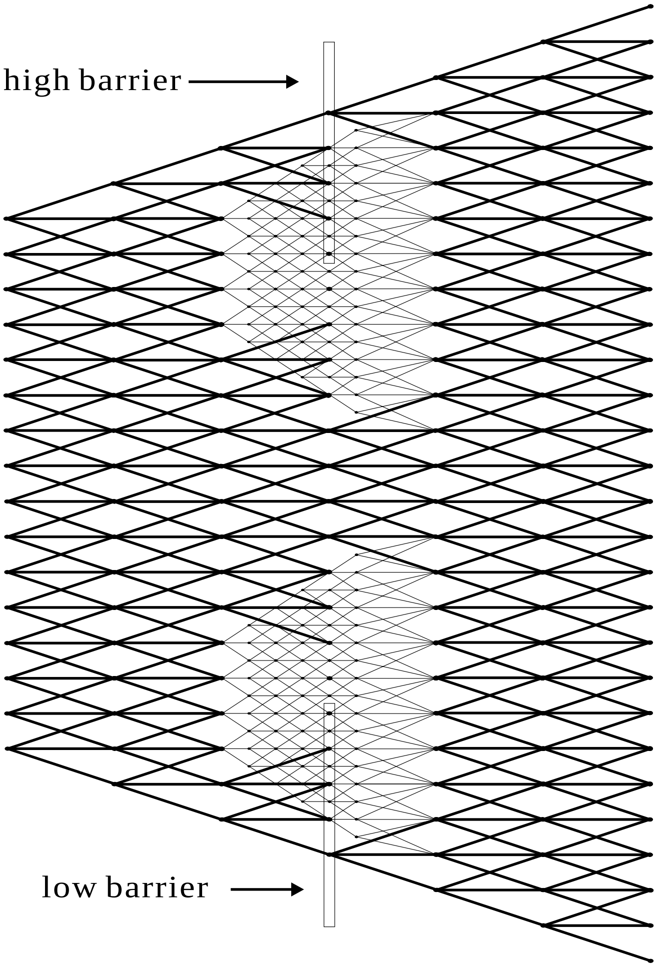


Figure 3.8: **An level 1 AMM Model for double discrete barrier options.**

However, there is still an extremely special situation that should be dealt with. It occurs when two barriers are so close that there are both barriers in fine mesh at the same time. Fig. 3.9 depicts this kind of situation. For illustration convenience this picture only present the adaptive mesh mechanism around down-barrier while the high barrier part is omitted (but it still exists). There are three kinds of meshes with different resolutions in Fig. 3.9, that is it is a level 2 AMM model. The base lattice is in bold lines and its nodes are labelled as A_i, where i is the index from the bottom node of base mesh in this figure. The thin lines represent the first level mesh with nodes named as B_i, where i is the index from the bottom node of first level mesh. At last, the second level mesh is drawn by dotted line without any node label. As the picture shows, the high barrier is so close to the low barrier that some B-level nodes such as B_{13}, B_{14}, and $B_{15}(= A_8)$ are knocked out to be zero by it. When the high barrier goes much closer to the low barrier such as the price level below B_{11}, not only the node values of first level fine mesh but also those of the second level fine mesh would be affected by the high barrier. Depending on how close the two barriers would be, further level of fine meshes would be influenced. On the other hand, if we move the high barrier away from low barrier such as the price level between the levels of A_8 and A_9, there will be no inter-influence of the two barrier ,and the adaptive mesh model for the two barriers can be built up in individual finer lattice. Nevertheless, building the finer mesh individually would not be a good idea in this case. Because now the price level of high barrier is between nodes A_8 and A_9 and the lowest first level AMM model lattice node of high barrier at barrier monitored date would be B_9, there will be lots of fine mesh nodes in the intersection area of two first level fine mesh that would be calculated twice giving rise to redundant computation. The aim of AMM mechanism is to reduce the nonlinearity error with limiting the increase of computation amount as much as possible. Therefore, this kind of situation with redundant node calculation is not desirable and the two level one fine mesh should be combined to be one.

Since AMM mechanism have been notorious for its unfriendly complicated structure for programmer to implement it, those situations describe above in double barrier options highly enhance the difficulties in programming. We are going to list some facts we have observed which would be quite helpful for those who want to implement the AMM mechanism in pricing double barrier options. First of all, let us define some inter-statuses of the two fine meshes of high barrier and low barrier. If we call the two meshes are ”individual”, it means that there is not any intersectional price node of both two fine mesh of the same level at the barrier monitored date. On the other hand, the two meshes are ”combined” while there is at least one overlapped price node of the two fine meshes at barrier monitored date. We set the base lattice to be combined. Here we list the facts below:

1. If the two m-level meshes are combined, the two $(m-1)$-level meshes will be also combined.

2. If the two m-level meshes are individual, the two $(m+1)$-level meshes will be also individual.

3. There are only two situations that a level of meshes should be checked combined or individual. One is when we stand at a level of combined mesh and want to know the inter-status of next level meshes. The other is when we stand at two individual m-level meshes and wonder whether the two $(m-1)$-level meshes are combined or individual.

In discrete double barrier options there are two areas that would generate nonlinearity error at every barrier monitored dates, one is around high barrier and the other is around low barrier. The error would be cumulative that incorrectness is even mounting when the barrier condition is checked more frequently. If we want to reduce the nonlinearity error contributed by barriers in traditional trinomial mechanism, halving the price step may quadruple the number of time steps and would make the node value calculation become 16 times. However, with AMM mechanism be applied, the same nonlinearity error elimination can be accomplished by a reasonably small increase of computation amount. Adding a finer mesh needs 60 extra nodes to be calculated. We can get the number by summing up all B-level nodes in Fig. 3.9, but among these 60 nodes there are some nodes that need no extra calculation such as those nodes being knocked out at barrier check date and their branching nodes at next fine mesh time state. Hence, we can say that to cut the nonlinearity error half would increase the computation amount not more than 120 nodes calculation (in worst case, fine meshes of the two barriers are individual and 60 for each one). Fig. 3.10 depicts the convergence behavior of AMM model for a double discrete out-barrier European call. The number of time steps starts from 6 and ends with 1,200.

The AMM model for double barrier options proposed in this section can be also applied to the moving double barrier without any modification. The ”moving” here means the two barrier can differ at different monitoring dates. The essence of AMM mechanism is to reduce the nonlinearity error by adding the density of lattice around the critical area. Hence, as we can find in the mechanism proposed above that no matter the barriers are fixed or not, the adaptive mesh mechanism just build up the finer mesh around the barriers. It is an example of the strong flexibility of AMM model.

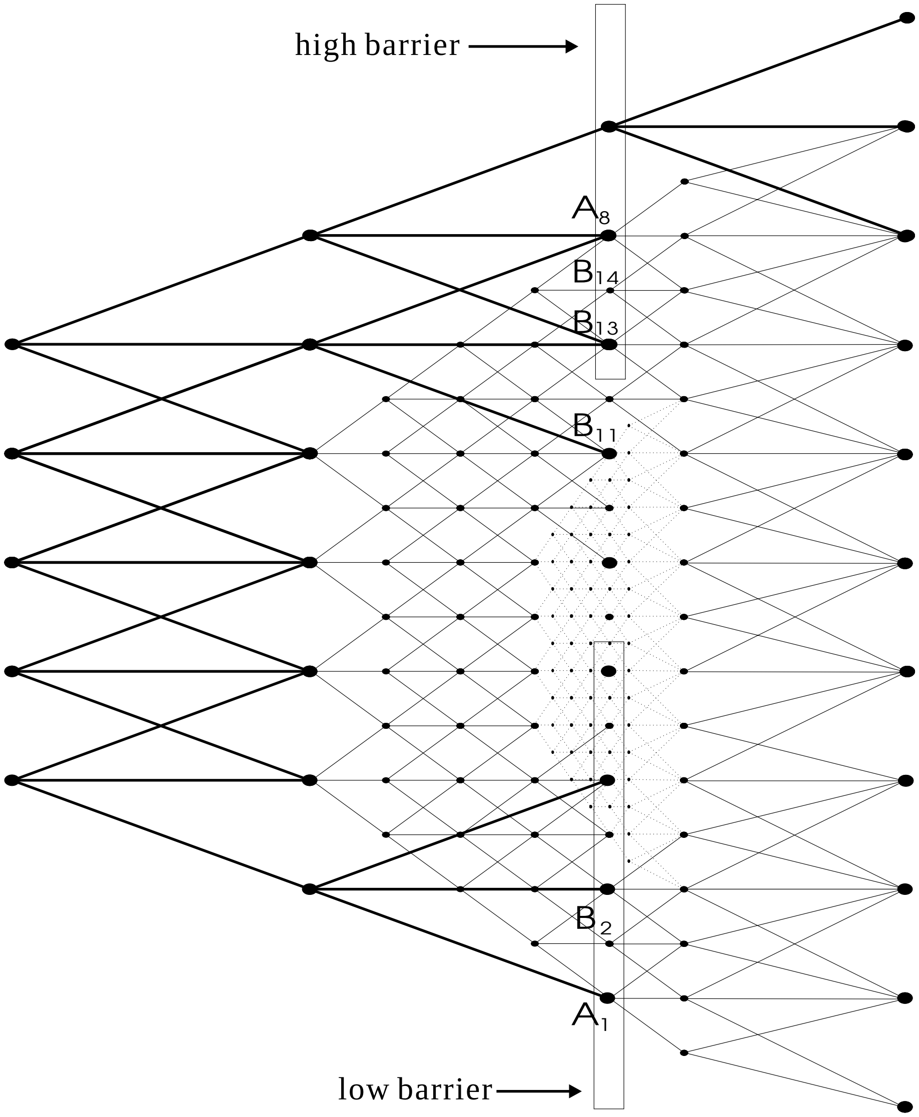


Figure 3.9: **An level 1 adaptive mesh model for double discrete barrier options.**

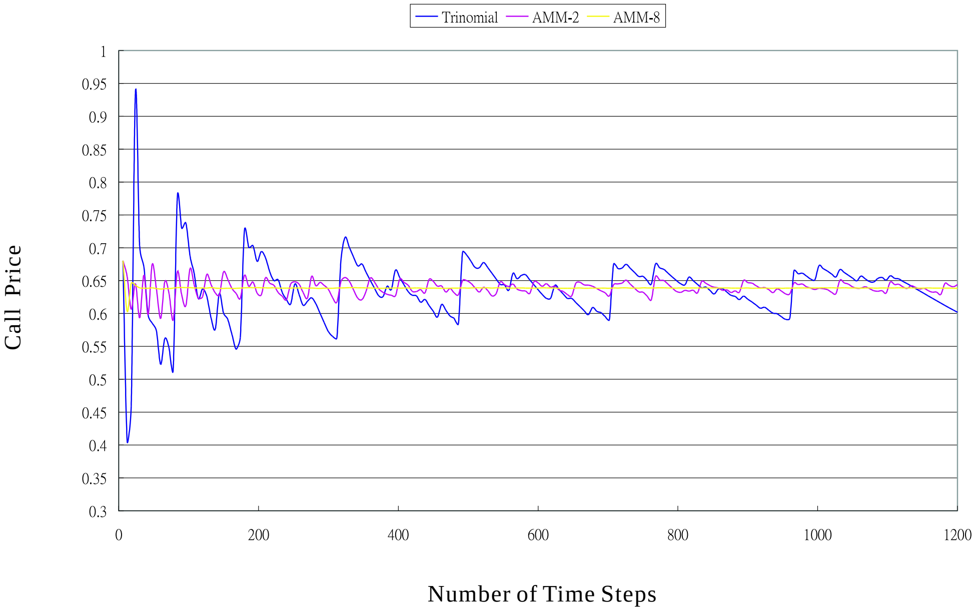


Figure 3.10: **The AMM Model convergence for a double discrete out-barrier European call.**
$S = 100$, $X = 100$, up-barrier $H = 110$, down-barrier $L = 90$, $\sigma = 20\%$, $r = 10\%$, $q = 0\%$, $T = 0.5$ year and monitoring frequency $F = 6$.

Chapter 4

Numerical Results

In this chapter we compare the AMM model with other mechanisms divided into three different categories: trinomial tree lattice mechanisms, the BGK formula approach, and the quadrature method. There are three trinomial tree mechanisms to compete with AMM model. The first is the trinomial method for ordinary options provided by Kamrad and Ritchken (1991) [17]. The second is a tree lattice with a stretch parameter proposed by Ritchken (1995) [9] for continuously monitored not only single but also double barrier options. However, with only a little modification the same mechanism can be also applied to discrete barrier options where this paper is mainly focused. At last, the Broadie, Glasserman, and Kou's Enhanced Trinomial Tree mechanism [11] is implemented to compare with AMM. In the category of formula-based approach, Broadie, Glasserman, and Kou also propose a continuity correction to the formula of continuous barrier option which is called BGK model for pricing discrete barrier options[5]. Finally, the quadrature method firstly suggested by Andricopoulos et al. (2003) is carried out. The quadrature method has characteristics of multinomial lattice and finite difference method and is especially powerful in pricing of discretely monitored derivatives.

All competing methods in this chapter are implemented in C++ programs running on a PC with an Intel Pentium 4 3.2GHz CPU and 1.0 GB of RAM.

4.1 Trinomial Tree Lattice Mechanisms

4.1.1 The Ritchken Trinomial Tree Mechanism

In [9] Ritchken proposes an approximated tree lattice for continuous barrier options. Let us set $X = \ln S$, where S is the underlying asset price. The

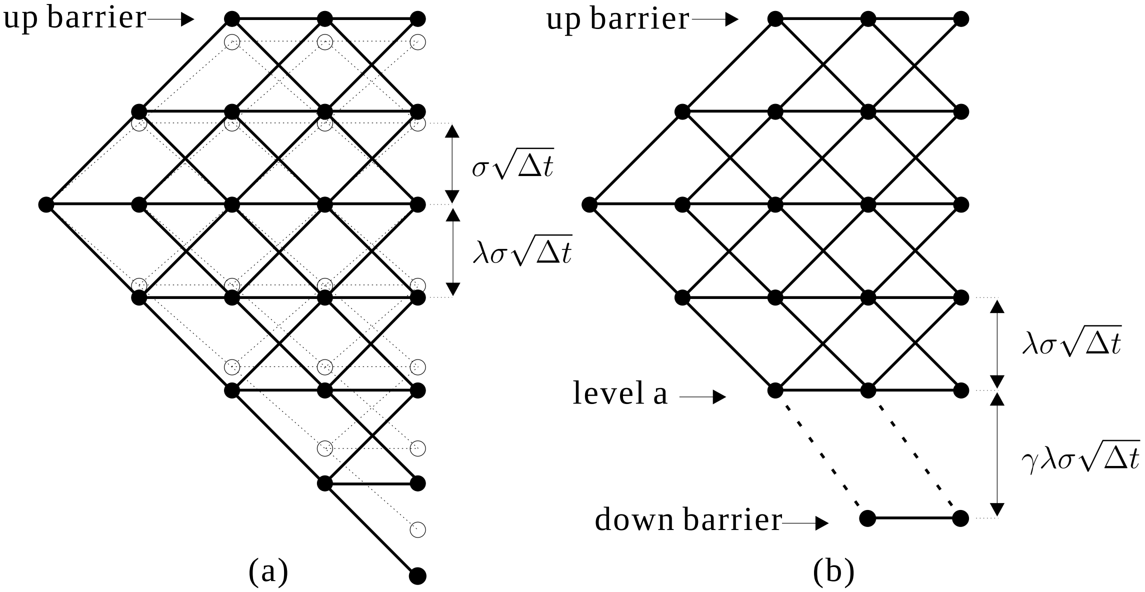

up barrier
$\sigma\sqrt{\Delta t}$
$\lambda\sigma\sqrt{\Delta t}$
(a)
up barrier
$\lambda\sigma\sqrt{\Delta t}$
level a
$\gamma\lambda\sigma\sqrt{\Delta t}$
down barrier
(b)

Figure 4.1: **The Ritchken Trinomial Tree for continuous barrier options.**

(a)Single barrier options. (b)Double barrier options.

Ritchken's trinomial process is defined as below:

$$X_{t+\Delta t} - X_t = \begin{cases} \lambda\sigma\sqrt{\Delta t}, & \text{with probability } p_u \\ 0, & \text{with probability } p_m \\ -\lambda\sigma\sqrt{\Delta t}, & \text{with probability } p_d. \end{cases} \tag{4.1}$$

and p_u, p_m, and p_d are

$$\begin{aligned} p_u &= \frac{1}{2\lambda^2} + \frac{\alpha\sqrt{\Delta t}}{2\lambda\sigma}, \\ p_m &= 1 - \frac{1}{\lambda^2}, \\ p_d &= \frac{1}{2\lambda^2} - \frac{\alpha\sqrt{\Delta t}}{2\lambda\sigma}. \end{aligned}$$

where $1 \leq \lambda < 2$ and α and σ are defined as before.

λ is the stretch parameter that controls the gap between layers of prices on the lattice and can be adjusted to make the lattice "hit" a single barrier as shown in Fig. (a). As to double barrier options, Ritchken in the same paper also proposes an additional stretch parameter, γ, to make the second barrier be hit by lattice. The tree lattice of Ritchken for the double barrier option is presented in Fig. (b). Let X^a denotes the variable X at level a. We have the process

$$X^a_{t+\Delta t} - X^a_t = \begin{cases} \lambda\sigma\sqrt{\Delta t}, & \text{with probability } p'_u \\ 0, & \text{with probability } p'_m \\ -\gamma\lambda\sigma\sqrt{\Delta t}, & \text{with probability } p'_d. \end{cases} \quad (4.2)$$

where $1 \leq \gamma < 2$.

Matching up the mean and variance for these nodes leads to

$$\begin{aligned} p'_u &= \frac{b + a\gamma}{1+\gamma}, \\ p'_m &= 1 - p'_u - p'_d, \\ p'_d &= \frac{b-a}{\gamma(1+\gamma)}. \end{aligned}$$

where $a = \frac{\alpha\sqrt{\Delta t}}{\lambda\sigma}$ and $b = \frac{1}{\lambda^2}$.

However, those mechanism proposed by Ritchken are all for continuous barrier options. For those barriers monitored discretely we should not only calculate lattice nodes between price levels of up-barrier and down-barrier but also take into account those nodes above up-barrier and below down-barrier. It would be no problem for us using the process in Eq. (4.1) except for nodes at the same level of down-barrier. The process for the nodes at down-barrier level should be as follows:

$$X^{Hd}_{t+\Delta t} - X^{Hd}_t = \begin{cases} \gamma\lambda\sigma\sqrt{\Delta t}, & \text{with probability } p''_u \\ 0, & \text{with probability } p''_m \\ -\lambda\sigma\sqrt{\Delta t}, & \text{with probability } p''_d. \end{cases} \quad (4.3)$$

where

$$\begin{aligned} p''_u &= \frac{a+b}{\gamma(\gamma-1)}, \\ p''_m &= 1 - p''_u - p''_d, \\ p''_d &= \frac{b - a\gamma}{\gamma+1}. \end{aligned}$$

with a and b defined as earlier.

4.1.2 The Enhanced Trinomial Tree Mechanism

Broadie et al. followed their continuity correction concept [5] and proposed a barrier-shifted lattice mechanism for discrete barrier options called *enhanced trinomial method* in 1999[11]. They use the same trinomial approach of Ritchken's method described just above but shift the discrete barrier at level H to $H' = He^{\pm 0.5\lambda^* \sigma\sqrt{\Delta t}}$(with + for an up barrier and − for a down barrier). The $\lambda^* = \sqrt{3/2}$ is recommended by Boyle [18] and Omberg [19], and $0.5\lambda^*\sigma\sqrt{\Delta t}$ is the average overshoot over a boundary for the random walk process. Broadie et al. suggest a procedure producing an n (time step number) which is divisible by m (barrier monitoring frequency), a λ (stretch parameter) which is close to λ^*, and a layer of nodes which coincides with the shifted barrier, and then use those parameters to construct the enhanced trinomial tree.

Nevertheless, for the convenience of comparing with other mechanisms, the time step number n should be free for input. Hence, we use a different procedure against Broadie et al.. Let $\lambda_k = |log(H/S)|/(k\sigma\sqrt{\Delta t})$, for $k = 1$, 2, ...,k', where k' corresponds to the first time a layer of nodes crosses the shifted barrier without stretch of price step size (i.e. $\lambda = 1$). Then we choose the λ from λ_k which minimizes $|\lambda_k - \lambda^*|$ for $k = 1$, 2, ...,k', no matter what kind of n is input. But the λ we choose here can only make one barrier be matched by a price level of enhanced trinomial tree so we apply the Ritchken's second stretch parameter technique described above to the enhanced trinomial tree making the second barrier be hit.

Finally, there is a noteworthy point. Broadie et al. remark by themselves that the enhanced trinomial method preforms better with less frequent monitoring of the barrier.

4.1.3 Numerical Comparisons

Since the competing mechanisms have been shortly introduced, now we can turn our focus onto the numerical comparisons of these methods.

Table. 4.1 shows numerical comparisons of AMM with its competitors in a down-and-out option under different barriers and different condition monitoring frequencies. We choose the benchmark as the AMM with AMM level of 8 and time step number $n = 1,000,000$. It's because we can find from our research data that AMM contributes the best convergence rate, and result prices of all other methods are getting closer to AMM-8's value while time step number is increasing. We see an example of this phenomenon in Table. 4.2 by numerical data, and also there are some figures of convergence rate in Fig. 4.3.

In Table. 4.1 AMM-8 generally dominates over other methods in accuracy. The enhanced trinomial lattice takes the second place followed by the Ritchken's method and standard trinomial tree. No matter what monitoring frequency it is, all methods invoke worse outcomes while the down barrier gets closer to the initial price. Moreover, we can see from the table

Barrier	Benchmark[a]	Trinomial[b]		Ritchken[c]		Enhanced Trinomial[d]		AMM-8	
		value[e]	error(%)[f]	value	error(%)	value	error(%)	value	error(%)
				monitoring frequency=2					
80	8.2566	8.2559	−0.0093	8.2561	−0.0063	8.2561	−0.0068	8.2566	−0.0001
90	8.1273	8.1393	0.1473	8.1160	−0.1390	8.1272	−0.0015	8.1272	−0.0020
95	7.8092	7.8208	0.1486	7.7752	−0.4346	7.8092	0.0017	7.8091	−0.0006
99	7.3019	7.3348	0.4516	7.2202	−1.1186	7.3097	0.1065	7.3022	0.0047
				monitoring frequency=5					
80	8.2535	8.2526	−0.0099	8.2525	−0.0115	8.2529	−0.0063	8.2535	0.0000
90	7.9118	7.9463	0.4365	7.8799	−0.4027	7.9123	0.0060	7.9117	−0.0008
95	7.0217	7.0542	0.4633	6.9293	−1.3158	7.0226	0.0131	7.0214	−0.0047
99	5.7210	5.8010	1.3978	5.5310	−3.3207	5.7398	0.3277	5.7219	0.0158
				monitoring frequency=25					
80	8.2435	8.2426	−0.0116	8.2414	−0.0256	8.2431	−0.0050	8.2435	0.0001
90	7.5882	7.6530	0.8527	7.5305	−0.7614	7.5904	0.0285	7.5881	−0.0020
95	5.9302	5.9946	1.0871	5.7524	−2.9982	5.9335	0.0563	5.9297	−0.0080
99	3.4393	3.5885	4.3374	3.1095	−9.5892	3.4748	1.0329	3.4382	−0.0335
				monitoring frequency=125					
80	8.2350	8.2340	−0.0120	8.2322	−0.0346	8.2347	−0.0032	8.2350	0.0001
90	7.3683	7.4530	1.1482	7.2996	−0.9335	7.3729	0.0618	7.3684	0.0004
95	5.3370	5.4211	1.5745	5.1280	−3.9177	5.3470	0.1866	5.3370	−0.0007
99	2.1829	2.3881	9.4019	1.7717	−18.8369	2.2230	1.8374	2.1801	−0.1284

It is an down-and-out call with $T = 0.5$ year, $r = 5\%$, $q = 0\%$, $\sigma = 25\%$, $S = 100$, and $X = 100$.

All methods are calculated with time steps $n = 750$.

[a] The Benchmark comes from the AMM-8 lattice with $1,000,000$ steps.

[b] The Trinomial is the standard trinomial tree proposed by Kamrad and Ritchken[17] with $\lambda = 1.2533136$[20].

[c] The Ritchken is the Ritchken Trinomial Tree Mechanism[9] with modification described above.

[d] The Enhanced Trinomial is proposed by Broadie et al.[11] with modification described above.

[e] All the values are rounded off to the forth decimal place.

[f] The error(%) field is the percentage pricing error = [approximation/(benchmark)−1]100% rounded to the forth decimal place with all the values computed before rounding.

Table 4.1: **Numerical comparisons of AMM with other tree lattice methods in single discrete barrier options.**

n	$1,000$	$10,000$	$100,000$	$1,000,000$
Trinomial	8.25230781	8.25328042	8.25345179	8.25347439
Ritchken	8.25262447	8.25321817	8.25338130	8.25343194
Enhanced Trinomial	8.25302744	8.25341562	8.25345090	8.25345481
AMM-8	8.25344311	8.25345525	8.25345551	8.25345628

Table 4.2: **An numerical data of convergence of tree methods in a down-and-out European Call.**
$S = 100$, $X = 100$, $H = 80$, $\sigma = 25\%$, $r = 5\%$, $q = 0\%$, $T = 0.5$ year and monitoring frequency $F = 5$.

that the results of Ritchken's method are even worse than those of the standard trinomial tree with a down barrier at the price level of 95 or 99. This kind of error arises from the option value drop around the barrier price at monitored date. Fig. 4.2 is a call value plotting related to the asset price around barrier at $4T/5$, which is just like the option value curve of a discrete barrier option with monitoring frequency $F = 5$ before the barrier condition at $4T/5$ is checked. As the down barrier is shifted upper (i.e. 80, 90, 95, and 99) and getting closer to initial asset price $S(= 100)$, the drop of option value after the barrier condition is checked is also increasing. The gap in option value curve gives rise to some kind of error similar to the nonlinearity error introduced before. Although the Ritchken's method makes the lattice hit the down-barrier price level, this kind of error still occurs in those nodes whose down branch or middle branch hits the barrier at time $4T/5 - \Delta t$ and makes the result go awry. On the other, the enhanced trinomial tree with the continuity correction and AMM-8 applying higher mesh resolution to the critical area can both restrain this kind of error.

We can also observe from Table. 4.1 that the higher the monitoring frequency is, the more erroneous the approximated option price values are. It is very intuitive because the erroneousness contributed by option value drop at each barrier monitored time is cumulative. Hence, the option with higher monitoring frequency will be priced with greater fallaciousness. Furthermore, just like Broadie et al. suggests in [11] that the enhanced trinomial method is more accurate with low monitoring frequency, it gets a larger error rate than others in the case of barrier price 80 and monitoring frequency 125.

Fig. 4.3 shows the convergence behaviors of discrete barrier option with monitoring frequencies $F = 5$ and $F = 25$. It is very clear that AMM-8 has the best convergence rate of result, values of the enhanced trinomial method converge worse with higher monitoring frequency, outcomes of the Ritchken's

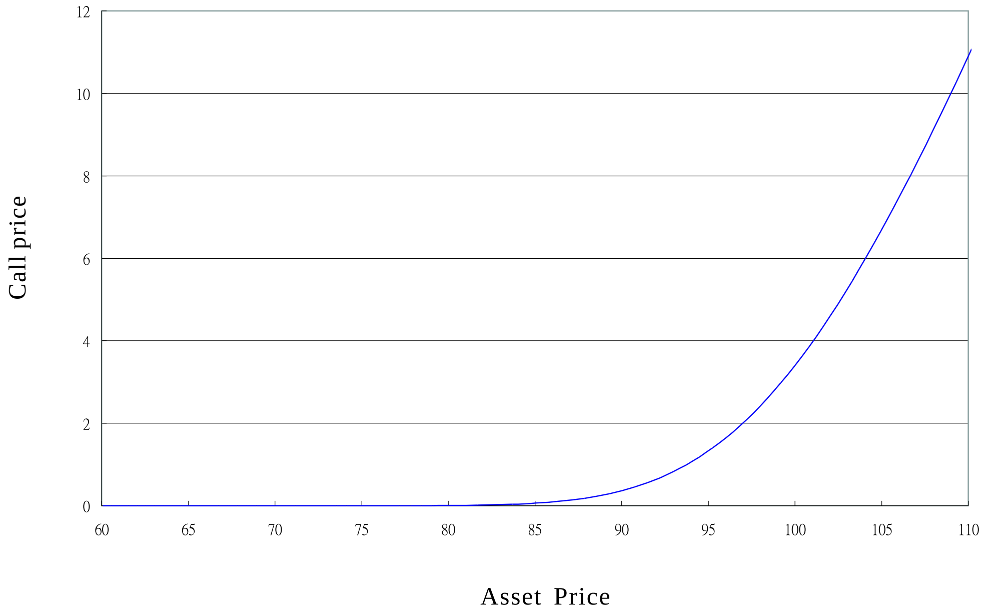
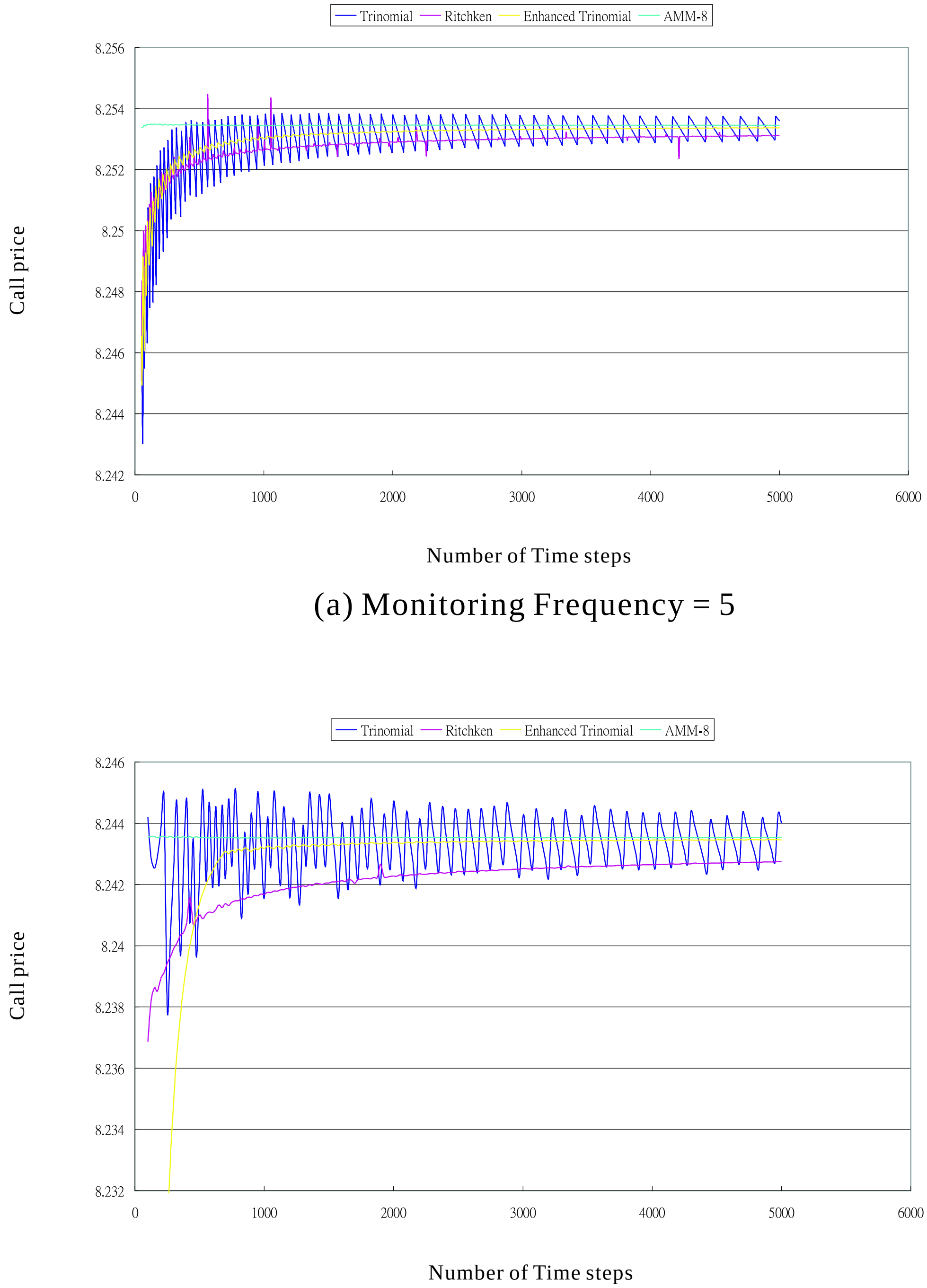


Figure 4.2: **A call option value in relation to asset price at $4T/5$.** $S = 100$, $X = 100$, $\sigma = 25\%$, $r = 5\%$, $q = 0\%$, and $T = 0.5$ year.

lattice are usually under-estimated, and the results of all other methods are going to converge to the one of AMM-8 with the increase of the increasing time step number.

We are also interested in the pricing behavior of those methods under barrier-too-close situation. Table. 4.3 is a table with numerical comparisons with different too-close barriers. There is an answer come out from traditional trinomial method in every close barrier. But a greater time step number n doesn't promise a smaller percentage error because of obvious zig-zag curve shape and slow convergence rate as we can see in Fig. 4.3. The convergence curve of AMM-8 is also sawtooth-like. However, because of AMM's fast convergence rate, the erroneousness of AMM-8 in Table. 4.3 decreases while the time step number n increases except for the case between $n = 20,000$ and $n = 35,000$ with a barrier 99.5. Also, we can clearly see the influence of the barrier-too-close problem in Ritchken's lattice and enhanced trinomial method. With the barrier getting closer to the initial asset price an extremely large number of time steps should be used to price options. However, the option value calculated by Ritchken's lattice and the enhanced trinomial method with $n = 35,000$ can't even compete against AMM-8 with only 500 time steps. There are some cases that the enhanced trinomial method



(a) Monitoring Frequency = 5

(b) Monitoring Frequency = 25

Figure 4.3: **The convergence behaviors for discrete down-and-out European calls with different monitored frequencies in tree methods.**

$S = 100$, $X = 100$, $H = 80$, $\sigma = 25\%$, $r = 5\%$, $T = 0.5$ year and $q = 0\%$.

Barrier	Benchmark[a]		Trinomial[b]		Ritchken[c]		Enhanced Trinomial[d]		AMM-8	
		n	value[e]	error(%)[f]	value	error(%)	value	error(%)	value	error(%)
99.5	3.0962	500	3.1041	0.2563	NA	NA	3.1022	0.1924	3.0957	−0.0177
		1,500	2.9548	−4.5657	2.9302	−5.3617	3.0286	−2.1847	3.0953	−0.0296
		5,000	3.0762	−0.6457	3.0071	−2.8795	3.0904	−0.1877	3.0965	0.0091
		20,000	3.1284	1.0409	3.0539	−1.3650	3.0994	0.1027	3.0963	0.0019
		35,000	3.1179	0.6991	3.0624	−1.0931	3.0977	0.0475	3.0963	0.0023
99.75	2.9267	500	3.1041	6.0625	NA	NA	NA	NA	2.9274	0.0248
		1,500	2.9548	0.9612	NA	NA	2.9401	0.4582	2.9271	0.0143
		5,000	2.8651	−2.1044	2.8606	−2.2602	2.8960	−1.0492	2.9266	−0.0039
		20,000	2.9167	−0.3411	2.8846	−1.4400	2.9237	−0.1072	2.9268	0.0039
		35,000	2.9577	1.0582	2.8849	−1.4290	2.9312	0.1542	2.9267	−0.0022
99.875	2.8428	500	3.1041	9.1950	NA	NA	NA	NA	2.8452	0.0861
		1,500	2.9548	3.9430	NA	NA	NA	NA	2.8434	0.0230
		5,000	2.8651	0.7869	NA	NA	2.8531	0.3654	2.8430	0.0078
		20,000	2.8118	−1.0878	2.8040	−1.3636	2.8270	−0.5542	2.8426	−0.0048
		35,000	2.8781	1.2436	2.8012	−1.4625	2.8207	−0.7772	2.8427	−0.0015
99.9	2.8260	500	3.1041	9.8415	NA	NA	NA	NA	2.8273	0.0460
		1,500	2.9548	4.5585	NA	NA	NA	NA	2.8268	0.0286
		5,000	2.8651	1.3836	NA	NA	2.8446	0.6592	2.8263	0.0109
		20,000	2.8118	−0.5022	NA	NA	2.8186	−0.2641	2.8259	−0.0018
		35,000	2.7989	−0.9581	2.7928	−1.1757	2.8122	−0.4879	2.8260	−0.0013

It is an down-and-out call with $T = 0.5$ year, $r = 5\%$, $q = 0\%$, $\sigma = 25\%$, $S = 100$, $X = 100$, and $F = 25$.

[a] The Benchmark comes from the AMM-8 lattice with $1,000,000$ steps.

[b] The Trinomial is the standard trinomial tree proposed by Kamrad and Ritchken[17] with $\lambda = 1.2533136$[20].

[c] The Ritchken is the Ritchken Trinomial Tree Mechanism[9] with modification described above.

[d] The Enhanced Trinomial is proposed by Broadie et al.[11].

[e] All the values are rounded off to the forth decimal place.

[f] The error(%) field is the percentage pricing error = [approximation/(benchmark)−1]100% rounded off to the forth decimal place with all the values computed before rounding.

Table 4.3: **Numerical comparisons of AMM with other tree lattice methods under barrier-too-close situation in single discrete barrier options.**

can come up with a result but Ritchken's lattice. It is because the barrier shift of the enhanced trinomial method increase the price distance between the initial asset price and the adjusted barrier. The result data in Table. 4.3 also represents that Ritchken's lattice is overwhelmed by the enhanced trinomial method under too-close barrier situation, and AMM-8 is still the method with the best convergence rate when barrier is too close.

Table. 4.4 and Fig. 4.4 depict comparing numerical results and pricing convergence behaviors of discrete double knock-out barrier options. We can see from those numerical data that all the observations just found from single barrier options can also be adapted to double barrier case. Comparing the percentage errors in Table. 4.4 with those of relative fields in Table. 4.1 shows that the fallaciousness of double barrier option pricing is generally greater than that of pricing in single barrier options. It is because the pricing error does not only originate from down barrier but also from up barrier, besides the option value drop arising from condition check of up-barrier is also larger than that from down-barrier. We can see the convergence behavior of those methods in Fig. 4.4. AMM-8 performs well enough even with small numbers of time steps. The convergence of the enhanced trinomial method is also good enough with monitoring frequency $F = 5$. However, when it comes to the case with monitoring frequency $F = 25$ (i.e. a higher monitoring frequency), we can find results of the enhanced trinomial method with unacceptable errors when the time step number $n \leq 500$. Like the convergence behaviors in single barrier options, the standard trinomial method converges with an obvious zip-zap phenomenon ,and the results of Ritchken's lattice are always under-estimated and converge along the lower edge of sawtooth-like curve of the standard trinomial method.

Fig. 4.5 is the time-error plotting with x-axis denoting execution time of programs and y-axis denoting percentage error of results. All the programs are optimized by omitting to calculate those nodes whose backward induction nodes are going to hit the barrier. The marks in Fig. 4.5 are generated by programs with input parameter n starting from 25 and incrementing by 25 under other input parameters being fixed. Although there are still some uncontrolled variables which would influence the execution time of programs such as the temperature of CPU, the loading of operating system and memory size etc., the plotting in Fig. 4.5 can still give us a rough sketch of the time-error relationship of those competing methods. We can see from Fig. 4.5 that spending extra node calculation for finer mesh nodes makes AMM comes up with a more accurate result in a smaller number of time steps than Ritchken's lattice and the enhanced trinomial method. We can also tell from Fig. 4.5 that AMM-8 is generally more accurate and faster than other three methods.

Barrier		Benchmark[a]	Trinomial[b]		Ritchken[c]		Enhanced Trinomial[d]		AMM-8	
H	L		value[e]	error(%)[f]	value	error(%)	value	error(%)	value	error(%)
				monitoring frequency=2						
120	80	2.8384	2.8302	−0.2880	2.7442	−3.3190	2.8392	0.0305	2.8386	0.0074
	90	2.7287	2.7294	0.0278	2.6256	−3.7784	2.7291	0.0158	2.7287	0.0028
	95	2.4855	2.4851	−0.0157	2.3566	−5.1857	2.4842	−0.0524	2.4857	0.0062
	99	2.1376	2.1507	0.6117	2.0044	−6.2303	2.1085	−1.3592	2.1380	0.0178
				monitoring frequency=5						
120	80	2.4499	2.4439	−0.2455	2.3562	−3.8234	2.4510	0.0453	2.4498	−0.0033
	90	2.2028	2.2173	0.6586	2.0900	−5.1194	2.2028	0.0014	2.2026	−0.0066
	95	1.6831	1.6926	0.5642	1.5302	−9.0865	1.6799	−0.1938	1.6829	−0.0126
	99	1.0812	1.1078	2.4606	0.9499	−12.1364	1.0402	−3.7926	1.0813	0.0104
				monitoring frequency=25						
120	80	1.9420	1.9490	0.3613	1.8545	−4.5065	1.9438	0.0924	1.9421	0.0041
	90	1.5354	1.5630	1.7998	1.4248	−7.2006	1.5362	0.0502	1.5353	−0.0036
	95	0.8668	0.8823	1.7828	0.7343	−15.2866	0.8644	−0.2825	0.8667	−0.0127
	99	0.2931	0.3153	7.5488	0.2224	−24.1209	0.2703	−7.8077	0.2931	−0.0242
				monitoring frequency=125						
120	80	1.6808	1.7477	3.9800	1.6044	−4.5466	1.6846	0.2230	1.6810	0.0082
	90	1.2029	1.2370	2.8394	1.1057	−8.0785	1.2064	0.2924	1.2029	0.0038
	95	0.5532	0.5699	3.0274	0.4568	−17.4136	0.5543	0.2123	0.5531	−0.0017
	99	0.1042	0.1201	15.2063	0.0688	−33.9769	0.0969	-7.0129	0.1041	−0.1119

It is an down-and-out call with $T = 0.5$ year, $r = 5\%$, $q = 0\%$, $\sigma = 25\%$, $S = 100$, and $X = 100$.

All methods are calculated with time steps $n = 750$.

[a] The Benchmark comes from the AMM-8 lattice with $1,000,000$ steps.

[b] The Trinomial is the standard trinomial tree proposed by Kamrad and Ritchken[17] with $\lambda = 1.2533136$[20].

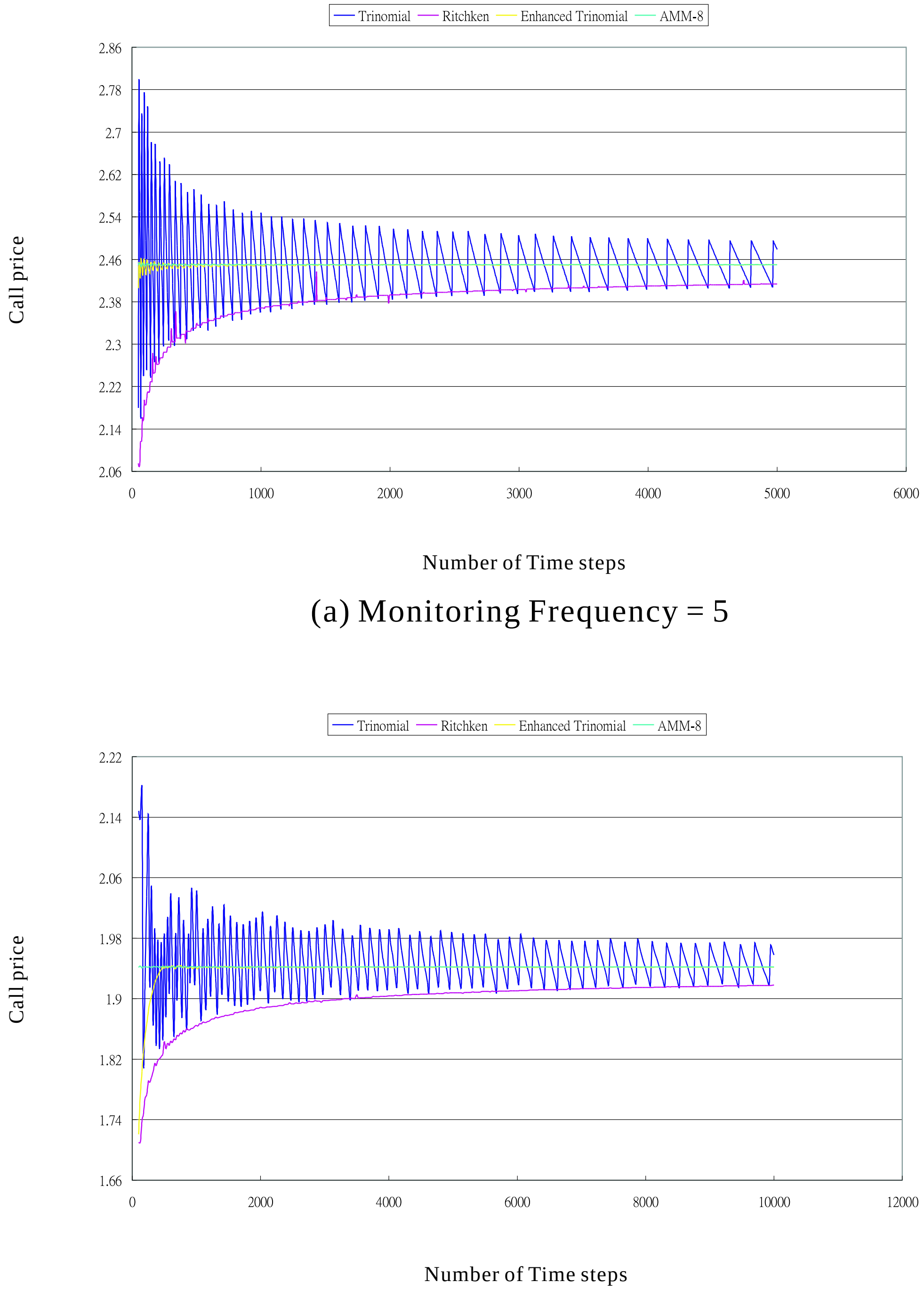
[c] The Ritchken is the Ritchken Trinomial Tree Mechanism[9] with modification described above. A λ is chose to hit the up barrier and a γ is chose to hit the down barrier.

[d] The Enhanced Trinomial is proposed by Broadie et al.[11] with modification described above. A λ is chose to hit the shifted up barrier and a γ is chose to hit the shifted down barrier.

[e] All the values are rounded off to the forth decimal place.

[f] The error(%) field is the percentage pricing error = [approximation/(benchmark)−1]100% rounded to the fourth decimal place with all the values computed before rounding.

Table 4.4: **Numerical comparisons of AMM with other tree lattice methods in double discrete barrier options.**

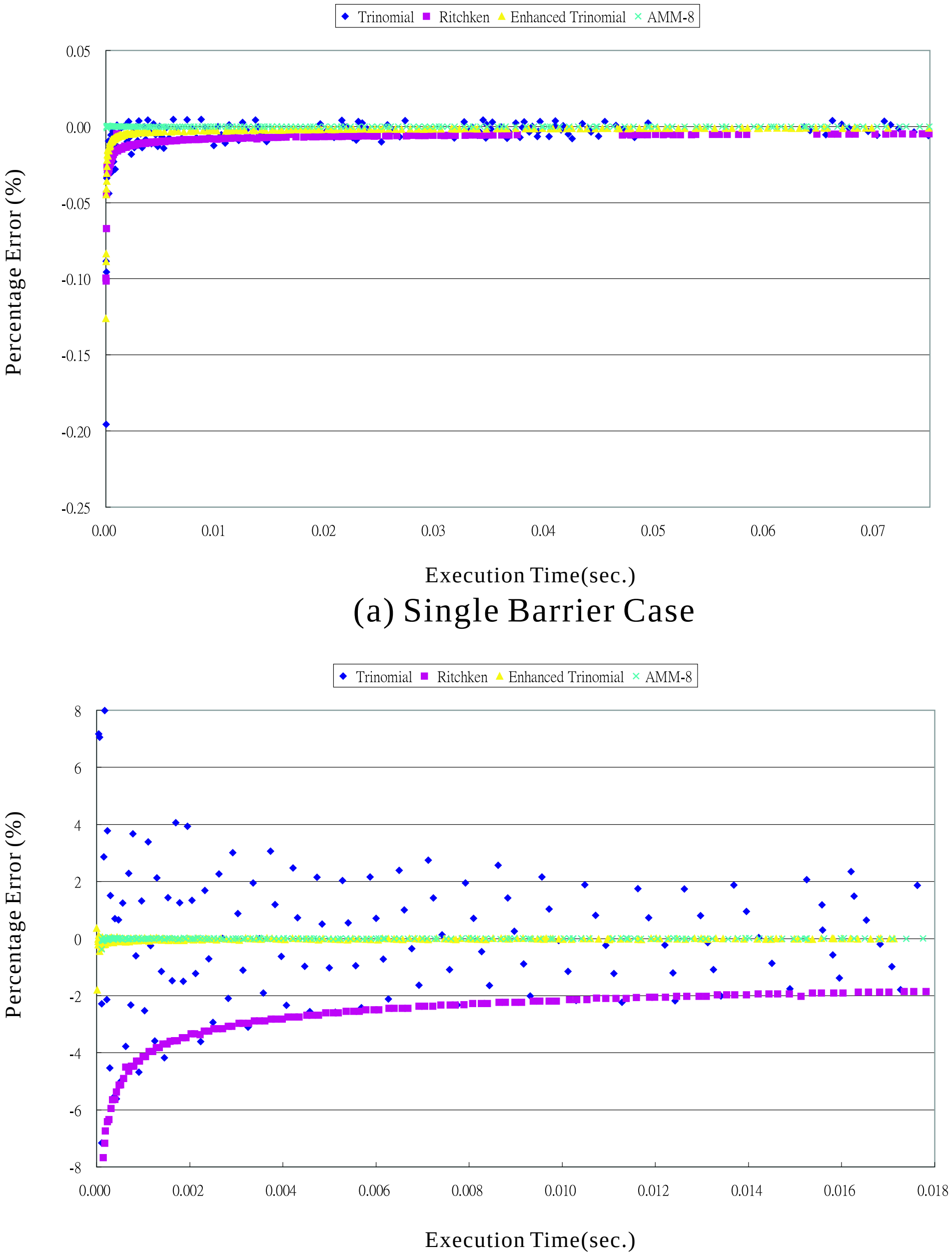


(a) Monitoring Frequency = 5

(b) Monitoring Frequency = 25

Figure 4.4: **The convergence behaviors for discrete down-and-out up-and-out European calls with different monitored frequencies in tree methods.**

$S = X = 100$, $L = 80$, $H = 120$, $\sigma = 25\%$, $r = 5\%$, $T = 0.5$ year and $q = 0\%$.



(a) Single Barrier Case

(b) Double Barrier Case

Figure 4.5: **The time-error plotting for discrete barrier options in tree methods.**

(a)Single down barrier case with $L = 80$. (b)Double barrier case with $H = 120$ and $L = 80$. Both cases are generated from setting the number of time steps n starting from 25 by an increment amount 25 with parameters $S = 100$, $X = 100$, $\sigma = 25\%$, $r = 5\%$, $T = 0.5$ year, $q = 0\%$, and $F = 5$. The percentage error is relative to the benchmark from AMM-8 with $n = 1,000,000$.

4.2 The BGK Formula Approach

The BGK formula approach is the method proposed by Broadie, Glasserman, and Kuo in [5] to price a discrete barrier option using the continuous monitored formula with an adjustment to the barrier away from the initial asset price. The continuity correction formula for discrete barrier options is as follows:

$$V_m(H) = V(He^{\pm\beta\sigma\sqrt{T/m}}) + o(\frac{1}{\sqrt{m}}).$$

where H is the barrier price, $V_m(H)$ is the price of a discretely monitored down-barrier call and up-barrier put with monitoring frequency $F = m$, $V(H)$ is the price of the corresponding continuously monitored barrier option, $+$ is for the case of up barrier, $-$ is for down-barrier, and $\beta = -\zeta(\frac{1}{2})/\sqrt{2\pi} \approx 0.5826$, with ζ the Riemann zeta function.

When the stock price follows a continuous path and is monitored continuously, the stock price is always equal to the barrier level at the moment the barrier is breached, but when the barrier is monitored discretely, there will almost always be an overshooting whenever the barrier is breached (i.e. the stock price is almost always below the barrier when the barrier is reached from above, and vice versa). It turns out that the value of a discrete barrier option can be Taylor-expanded in terms of the size of the overshooting, and it can then be approximated by using the closed-form formula for the continuous monitoring case with the barrier adjusted by the overshooting amount.

4.2.1 Numerical Comparisons

There is no close-form solution for continuous double barrier options such that the BGK method can only be used to price options with a single discrete barrier. Fig. 4.5 shows the numerical comparisons of AMM-8 with the BGK method in different monitoring frequencies and barrier prices. As Broadie et al. have mentioned in their paper, we can see from the table that the BGK model works with high accuracy when the barrier is not too close to the initial asset price. Also, the higher the monitoring frequency is, the smaller the pricing error of the BGK method is. Comparing AMM-8 with the BGK method, AMM-8 dominates the BGK method in monitoring frequencies $F = 5$, 25, and 125. However, the gap of pricing errors between these two methods is shrinking as the monitoring frequency is rising. When it comes to the case of monitoring frequency $F = 250$, the BGK method can beat AMM-8 under those cases of $L = 80$, 90, and 95.

Barrier	**Benchmark**[a]	**BGK**[b]		**AMM-8**[c]	
		value[d]	error(%)[e]	value	error(%)
		monitoring frequency=5			
80	8.2535	8.2540	0.0062	8.2535	0.0000
90	7.9118	7.9206	0.1119	7.9117	−0.0008
95	7.0217	6.9208	−1.4373	7.0214	−0.0047
99	5.7210	5.0769	−11.2581	5.7219	0.0158
99.9	5.3700	4.4853	−16.4756	5.3701	0.0009
		monitoring frequency=25			
80	8.2435	8.2436	0.0012	8.2435	0.0001
90	7.5882	7.5891	0.0115	7.5881	−0.0020
95	5.9302	5.9341	0.0664	5.9297	−0.0080
99	3.4393	3.2039	−6.8438	3.4382	−0.0335
99.9	2.8260	2.3728	−16.0357	2.8279	0.0675
		monitoring frequency=125			
80	8.2350	8.2350	0.0001	8.2350	0.0001
90	7.3683	7.3684	0.0012	7.3684	0.0004
95	5.3370	5.3371	0.0011	5.3370	−0.0007
99	2.1829	2.1431	−1.8222	2.1801	−0.1284
99.9	1.3928	1.1924	−14.3883	1.3946	0.1266
		monitoring frequency=250			
80	8.2324	8.2324	−0.0000	8.2324	−0.0001
90	7.3080	7.3080	0.0004	7.3081	0.0009
95	5.1795	5.1795	0.0004	5.1795	0.0018
99	1.8797	1.8699	−0.5238	1.8752	−0.2407
99.9	1.0256	0.8899	−13.2265	1.0249	−0.0672

It is an down-and-out call with $T = 0.5$ year, $r = 5\%$, $q = 0\%$, $\sigma = 25\%$, $S = 100$, and $X = 100$.

[a] The Benchmark comes from the AMM-8 lattice with $1,000,000$ steps.

[b] The BGK model is continuity correction to the formula proposed by Broadie et al.[5].

[c] The values of AMM-8 is calculated by AMM level 8 with the number of time steps $n = 750$.

[d] All the values are rounded off after the forth decimal place.

[e] The error(%) field is the percentage pricing error = [approximation/(benchmark)−1]100% rounded off to the forth decimal place with all the values computed before rounding.

Table 4.5: **Numerical comparisons of AMM with BGK model in single discrete barrier options.**

4.3 The Quadrature Method

The quadrature method is first proposed by Andricopoulos et al.[15] for single barrier options in 2003 and enhanced and extended for double barrier options by Tsai[21] in 2005. Here we use the quadrature method of Tsai which is going to be introduced below to compare with AMM.

The concept of quadrature method for single discrete barrier options is depicted by Fig. 4.6 and Fig. 4.7 that we first calculate all option values of nodes at maturity date (i.e. $m = 4$), and then induct backward to the initial date (i.e. $m = 0$) by integrating each node value at each barrier monitoring date. Define S_0 as the initial asset price, X as the exercise price, M as the monitoring frequency, T as the option lifetime, and $\Delta t = T/M$ as the time interval between barrier observations. The option value at time t with asset price S_t denoted by $V(S_t, t)$ can be expressed as follows.

$$V(S_t, t) = e^{-r\Delta t} \int_0^\infty V(S_{t+\Delta t}, t + \Delta t) f(S_{t+\Delta t}) dS_{t+\Delta t}.$$

where $f(.)$ is the probability density function of the lognormal discribution which is assumed to be followed by underlying asset price. With the standard transformation by $x = log(S_t/X)$ and $y = log(S_{t+\Delta t}/X)$, the equation becomes

$$\begin{aligned} V(x,t) &= e^{r\Delta t} \int_{-\infty}^{\infty} V(y, t + \Delta t) f(y) dy, \\ f(y) &: N(x + [(r-q)\frac{1}{2}\sigma^2]\Delta t, \sigma\sqrt{\Delta t}). \end{aligned}$$

It can be simplified thus:

$$V(x,t) = A(x) \int_{-\infty}^{\infty} B(x,y) V(y, t + \Delta t) dy, \tag{4.4}$$

where

$$\begin{aligned} A(x) &= \frac{1}{\sqrt{2\sigma^2 \pi \Delta t}} e^{-\frac{1}{2}kx - \frac{1}{8}\sigma^2 k^2 \Delta t - r\Delta t}, \\ B(x,y) &= e^{-\frac{(x-y)^2}{2\sigma^2 \Delta t} + \frac{1}{2}ky}, \\ k &= \frac{2(r-q)}{\sigma^2} - 1. \end{aligned}$$

Eq. 4.4 contains an integral that has to be evaluated by numerical techniques. The Simpson's rule is requested to approximate the integral.

The idea of Simpson's rule is simple. For a function of y, divide the desired range of y, $[a_1, a_2]$ into n intervals of a fixed length δy such that $n\delta y = a_2 - a_1$. Then approximate the integral by summing the area of the individual regions. This yields the following expression:

$$\int_{a_1}^{a_2} f(y)dy \approx \frac{\delta y}{6}\{f(a_1) + 4f(a_1 + \frac{1}{2}\delta y) + 2f(a_1 + \delta y) + 4f(a_1 + \frac{3}{2}\delta y) + 2f(a_1 + 2\delta y) + \cdots + 2f(a_2 - \delta y) + 4f(a_2 - \frac{1}{2}\delta y) + f(a_2)\}. \quad (4.5)$$

Because there are a desired range $[a_1, a_2]$ and a fixed interval δy needed by Simpson's rule, the integral bounds in Eq. 4.4 must be truncated and then an interval length δy has to be decided. Now we can have a more clear picture of the scenario of pricing discrete barrier options by the quadrature method which is listed below.

1. Decide the desired integral range $[a_1, a_2]$ and interval δy for each barrier monitoring date. It is where the price level of those nodes are decided in Fig. 4.6 and Fig. 4.7 at each barrier monitoring date.
2. Calculate option values of all nodes at maturity date.
3. Backward induct node's value at time t, $V(x, t)$ by integrating with Simpson's rule to those option values of nodes at time $t + \Delta t$, $V(y, t + \Delta t)$.

4.3.1 Pricing Discrete Down-and-Out Barrier Options

There is an upper bound $Ymax_m$ depicted by Ko[22] that can be used for Simpson's rule in single barrier's case.

$$Ymax_m = log(S_0/X) + R\sigma\sqrt{m\Delta t}$$

where R can be any number greater than 7.5 and $m = 1, 2, 3, \cdots, M$ depending on which monitoring date it is. We set the lower bound of y at maturity to be zero because values of $V(y, t + \Delta t)$ in Eq. 4.4 is extinguished with $y = log(S_T/X) < 0$. As to the lower bound at other barrier monitoring dates, it will be $l = log(L/X)$ because any $y < l$ is going to be knocked out.

Define $x_{i,m} = log(S_{i,m\Delta t}/X)$ as a variable transformation of the asset price at the i-th node from the bottom at the m-th barrier monitoring date in Fig. 4.6. Next we can calculate option values of nodes at maturity date as

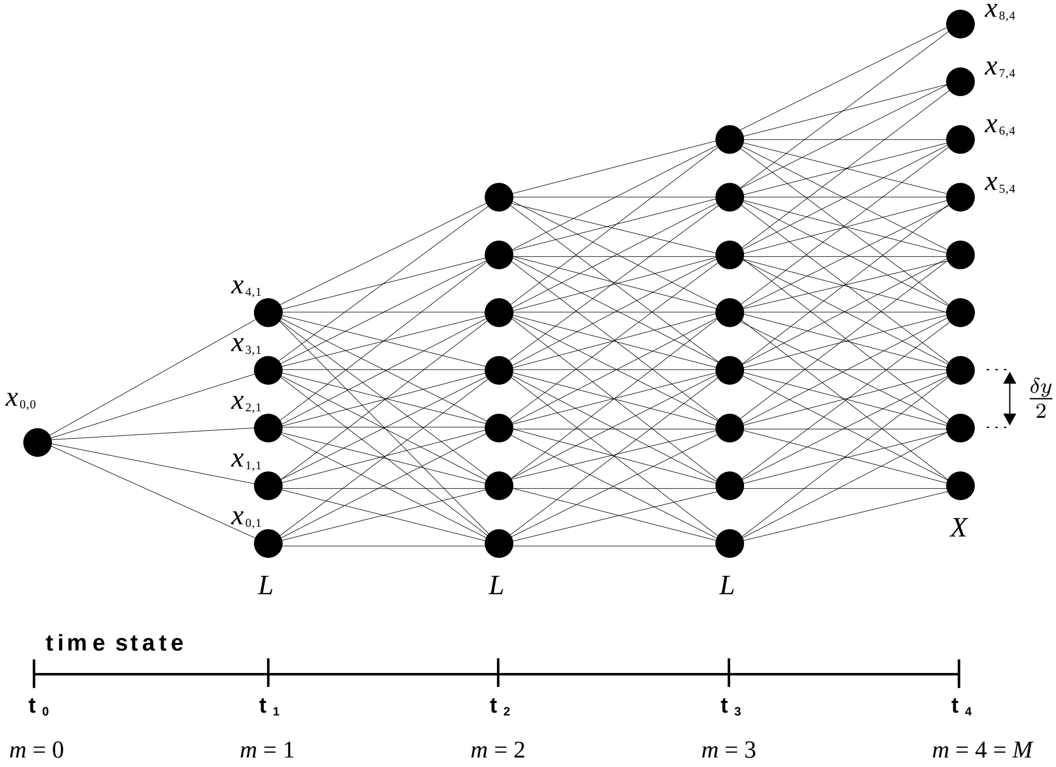


Figure 4.6: **The multinomial tree structure of quadrature method for single barrier options.**

$$V(x_{i,M}, T) = X(e^{x_{i,M}} - 1),$$

where

$$\begin{aligned} x_{i,M} &= \frac{i}{2}\delta y, \qquad i = 0, 1, 2, 3, \cdots, 2N_M \\ N_M &= [\frac{Ymax_M}{\delta y}], \end{aligned}$$

and δy can be any number smaller that $\sqrt{\delta t}/4$.

After constructing the last column of the multinomial tree as in Fig. 4.6, we can work on the option values of nodes with price level $x_{i,M-1}$ as

$$V(x_{i,M-1}, T - \Delta t) = A(x_{i,M-1}) \int_0^{Ymax_M} B(x_{i,M-1}, y)V(y, T)dy, \qquad (4.6)$$

where

$$\begin{aligned} x_{i,M-1} &= l + \frac{i}{2}\delta y, \qquad i = 0, 1, 2, 3, \cdots, 2N_{M-1} \\ N_{M-1} &= [\frac{Ymax_{M-1}}{\delta y}]. \end{aligned}$$

However, the bounds in Eq. 4.6 can be further truncated to $[x_{i,M-1} - 10\sigma\sqrt{\Delta t}, x_{i,M-1} + 10\sigma\sqrt{\Delta t}]$ because an asset price walk following the geometric Brownian motion is unlikely to move more than 10 standard deviations within one time period. According to the range, all the node values at maturity that should be used by Simpson's rule for integration of $V(x_{i,M-1}, T - \Delta t)$ are

$$V(y = x_{i,M}, T), \qquad i = 2v, 2v + 1, 2v + 2, \cdots, 2u$$

where

$$\begin{aligned} v &= ([\frac{x_{i,M-1} - 10\sigma\sqrt{\Delta t}}{\delta y}], 0)^+, \\ u &= ([\frac{x_{i,M-1} + 10\sigma\sqrt{\Delta t}}{\delta y}], N_M)^-. \end{aligned}$$

Then we can iteratively use the same procedure mentioned above to perform backward induction as in Fig. 4.6 until the result option price comes out. Note that the node price level at initial date $x_{0,0} = log(S_0/X)$ and the bounds for monitoring dates except for maturity should be $[l, Ymax_m]$ which could be truncated into $[x_{i,m} - 10\sigma\sqrt{\Delta t} - l, x_{i,m} + 10\sigma\sqrt{\Delta t} - l]$ where $m = 1, 2, \cdots, M - 1$.

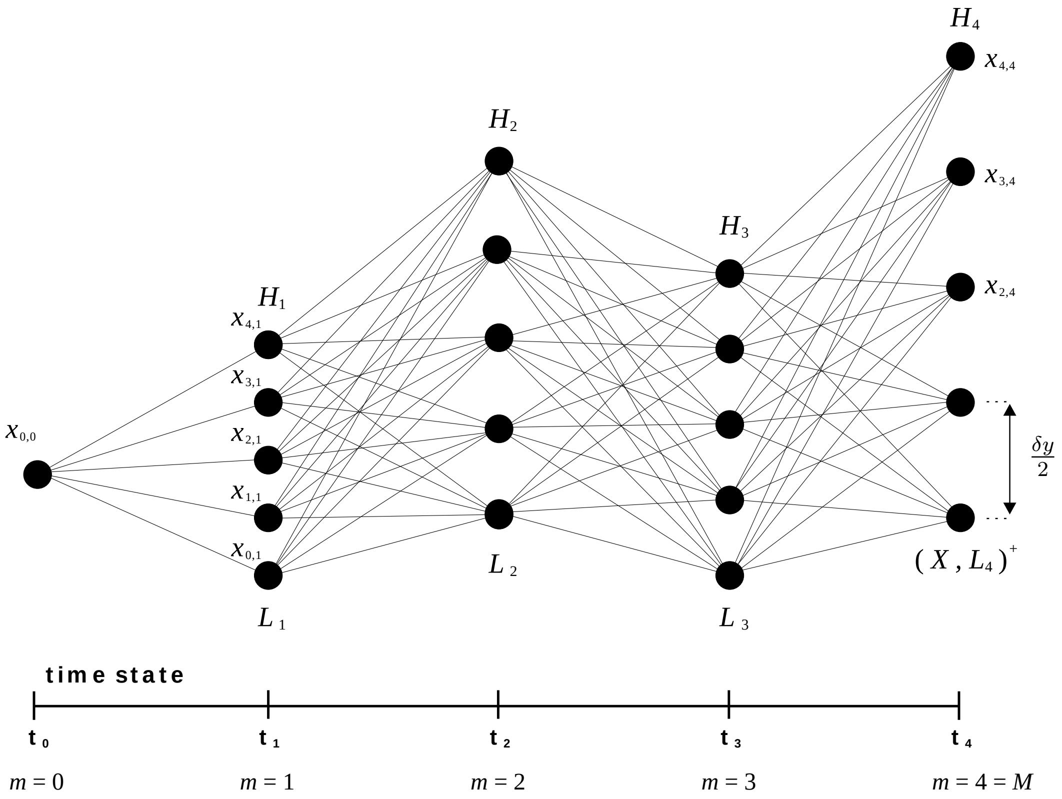


Figure 4.7: **The multinomial tree structure of quadrature method for double barrier options.**

Moreover, because the Simpson's rule converges at a rate of order $(\delta y)^4$, the outcomes of quadrature method can be further enhanced by the Richardson's extrapolation as

$$V_{ext} = \frac{(\delta y_1)^4 V_2 - (\delta y_2)^4 V_1}{(\delta y_1)^4 - (\delta y_2^4)},$$

where V_1 and V_2 are option prices calculated by quadrature method with δy_1 and δy_2.

4.3.2 Pricing Discrete Double Moving Knock-out Options

The quadrature method for moving discrete double barrier is sketched in Fig. 4.7. It is completely the same as the down-and-out case except for the way bounds being chose and the interval size δy being decided. The bounds for the m-th barrier monitoring date is

$$[l_m, h_m] \qquad m = 1, 2, \cdots, M$$

where

$$\begin{aligned} l_M &= log(max(X, L_M)/X), \\ l_m &= log(L_m/X), \qquad m = 1, 2, \cdots, M-1 \\ h_m &= log(H_m/X). \qquad m = 1, 2, \cdots, M \end{aligned}$$

To make the barriers are hit by the price levels of nodes, we must set the step size δy as

$$\delta y_m = \frac{h_m - l_m}{K_m}$$

where K_m represents the number of steps between up barrier and down barrier which must be an even integer. Besides, for the condition $\delta y_m < \sqrt{\Delta t_m}/4$, K_m can be any integer greater than $4(h_m - l_m)/\sqrt{\Delta t_m}$.

Hence, the m-th column has nodes with option values $V(x_{i,m}, T_m)$, where

$$\begin{aligned} &x_{0,0} = log(S_0/X), \\ &x_{i,m} = l_m + \frac{i}{2}\delta y_m. \\ &i = 0, 1, \cdots, 2K_m, m = 1, 2, \cdots, M \end{aligned}$$

The nodes needed for valuing $V(x_{i,m}, T_m)$, $m = 0, 1, 2, \cdots, M-1$, are

$$V(x_{j,m+1}, T_{m+1}), \qquad j = 0, 1, 2, \cdots, 2K_{m+1}$$

By adopting backward induction and the quadrature method, we can come up with the result by valuing every node $V(x_{i,m}, T_m)$ for $m = 0, 1, \cdots, M-1$ as:

$$\begin{aligned} V(x_{i,m}, T_m) \approx &\frac{A(x_{i,m})\delta y_{m+1}}{6}\{B(x_{i,m}, x_{0,m+1})V(x_{0,m+1}, T_{m+1}) \\ &+ 4B(x_{i,m}, x_{1,m+1})V(x_{1,m+1}, T_{m+1}) \\ &+ 2B(x_{i,m}, x_{2,m+1})V(x_{2,m+1}, T_{m+1}) + \cdots\cdots \\ &+ 2B(x_{i,m}, x_{2K_{m+1}-2,m+1})V(x_{2K_{m+1}-2,m+1}, T_{m+1}) \\ &+ 4B(x_{i,m}, x_{2K_{m+1}-1,m+1})V(x_{2K_{m+1}-1,m+1}, T_{m+1}) \\ &+ B(x_{i,m}, x_{2K_{m+1},m+1})V(x_{2K_{m+1},m+1}, T_{m+1})\}, \end{aligned}$$

where

$$\begin{aligned}
A(X_{i,m}) &= \frac{1}{\sqrt{2\sigma^2\pi\Delta t_{m+1}}} e^{-\frac{1}{2}kx-\frac{1}{8}\sigma^2k^2\Delta t_{m+1}-r\Delta t_{m+1}}, \\
B(x_{i,m}, x_{j,m+1}) &= e^{-\frac{(x_{i,m}-x_{j,m+1})^2}{2\sigma^2\Delta t_{m+1}}+\frac{1}{2}kx_{j,m+1}}, \\
k &= \frac{2(r-q)}{\sigma^2} - 1.
\end{aligned}$$

4.3.3 Numerical Comparisons

Before we compare AMM with the quadrature method, we must first decide what is the benchmark. There are two candidates. One is AMM-8 with $n = 1,000,000$ as before, and the other is come out from the quadrature method with small δy (extrapolation by results of $\delta y_1 = \sqrt{\Delta t}/150$ and $\delta y_2 = \sqrt{\Delta t}/300$ in single barrier case; $K_m = 200$ where $m = 1, 2, \cdots, M$ in double barrier case). Both two candidates output fairly precise outcomes which are generally the same to the 6-th decimal place and at least to the 4-th decimal place. Here we choose the result of quadrature to be our benchmark.

In Table 4.6 there are numerical results of QUAD, QUAD$_{\text{ext}}$, and AMM-8 in single barrier options with different barriers and different monitoring frequency. QUAD is the quadrature method and QUAD$_{\text{ext}}$ is the quadrature method with extrapolation applied. We can see from the percentage error fields that AMM-8 is usually more accurate than QUAD, but is overwhelmed by QUAD$_{\text{ext}}$ in every case. However, when we look at the value fields, all values including barrier-too-close cases are accurate to the second decimal place and generally precise to the third or forth decimal place which are well enough to be used in real market. Hence we can turn our focus onto the efficiency of these methods. Fig. 4.6 shows the relationship between monitoring frequency and execution time in single barrier options. Under other parameters are fixed, both AMM and the quadrature method require extra computation amount with the increase of monitoring frequency. The extra computation amount comes from finer mesh node calculation of additional barrier monitoring times in AMM. In quadrature method, it arises from the increase of time state number. Yet Fig. 4.8(a) shows that the execution time of AMM-8 is not always rising with the increase of monitoring frequency. It is because the optimization of our program omitting calculation of those nodes destined for being knocked out between two monitoring date offsets the computation amount gain by increasing monitoring time. Fig. 4.8(b) compare the frequency-time curve of AMM-8 with QUAD and QUAD$_{\text{ext}}$. We can get a clear picture that the quadrature method is less efficient than AMM with higher monitoring frequency. In this figure the execution time of

Barrier	Benchmark[a]	QUAD[b]		QUAD$_{\text{ext}}$[c]		AMM-8[d]	
		value[e]	error(%)[f]	value	error(%)	value	error(%)
			monitoring frequency=2				
80	8.2566	8.2573	0.0080	8.2566	−0.0000	8.2566	−0.0001
90	8.1273	8.1277	0.0049	8.1273	0.0000	8.1272	−0.0020
95	7.8092	7.8101	0.0114	7.8092	0.0000	7.8091	−0.0007
99	7.3019	7.3033	0.0199	7.3018	−0.0001	7.3022	0.0050
99.9	7.1514	7.1529	0.0215	7.1514	−0.0001	7.1517	0.0045
			monitoring frequency=5				
80	8.2535	8.2535	0.0007	8.2535	−0.0000	8.2534	0.0000
90	7.9118	7.9114	−0.0046	7.9118	0.0000	7.9117	−0.0004
95	7.0216	7.0219	0.0039	7.0216	0.0000	7.0214	−0.0036
99	5.7208	5.7219	0.0191	5.7208	−0.0001	5.7219	0.0187
99.9	5.3699	5.3711	0.0223	5.3699	−0.0001	5.3701	0.0030
			monitoring frequency=25				
80	8.2435	8.2435	−0.0002	8.2435	0.0000	8.2435	0.0001
90	7.5882	7.5879	−0.0043	7.5882	0.0000	7.5881	−0.0020
95	5.9302	5.9297	−0.0086	5.9302	0.0001	5.9297	−0.0080
99	3.4393	3.4398	0.0151	3.4393	0.0000	3.4382	−0.0327
99.9	2.8260	2.8266	0.0221	2.8260	−0.0001	2.8279	0.0676
			monitoring frequency=125				
80	8.2350	8.2350	−0.0001	8.2350	0.0000	8.2350	0.0001
90	7.3683	7.3682	−0.0020	7.3683	0.0000	7.3684	0.0004
95	5.3367	5.3367	−0.0071	5.3370	0.0000	5.3370	−0.0007
99	2.1829	2.1829	0.0019	2.1829	0.0000	2.1801	−0.1272
99.9	1.3928	1.3931	0.0185	1.3928	−0.0001	1.3946	0.1269
			monitoring frequency=250				
80	8.2324	8.2324	−0.0001	8.2324	0.0000	8.2324	−0.0001
90	7.3080	7.3079	−0.0015	7.3080	0.0000	7.3081	0.0009
95	5.1795	5.1792	−0.0053	5.1795	0.0000	5.1795	0.0018
99	1.8797	1.8796	−0.0056	1.8797	0.0001	1.8752	−0.2402
99.9	1.0256	1.0257	0.0170	1.0256	−0.0001	1.0249	−0.0682

It is an down-and-out call with $T = 0.5$ year, $r = 5\%$, $q = 0\%$, $\sigma = 25\%$, $S = 100$, and $X = 100$.

[a] The Benchmark comes from the quadrature method with $\delta y = \sqrt{\Delta t}/150$ and the Richardson extrapolation applied.

[b] The QUAD is the quadrature method with $\delta y = \sqrt{\Delta t}/5$.

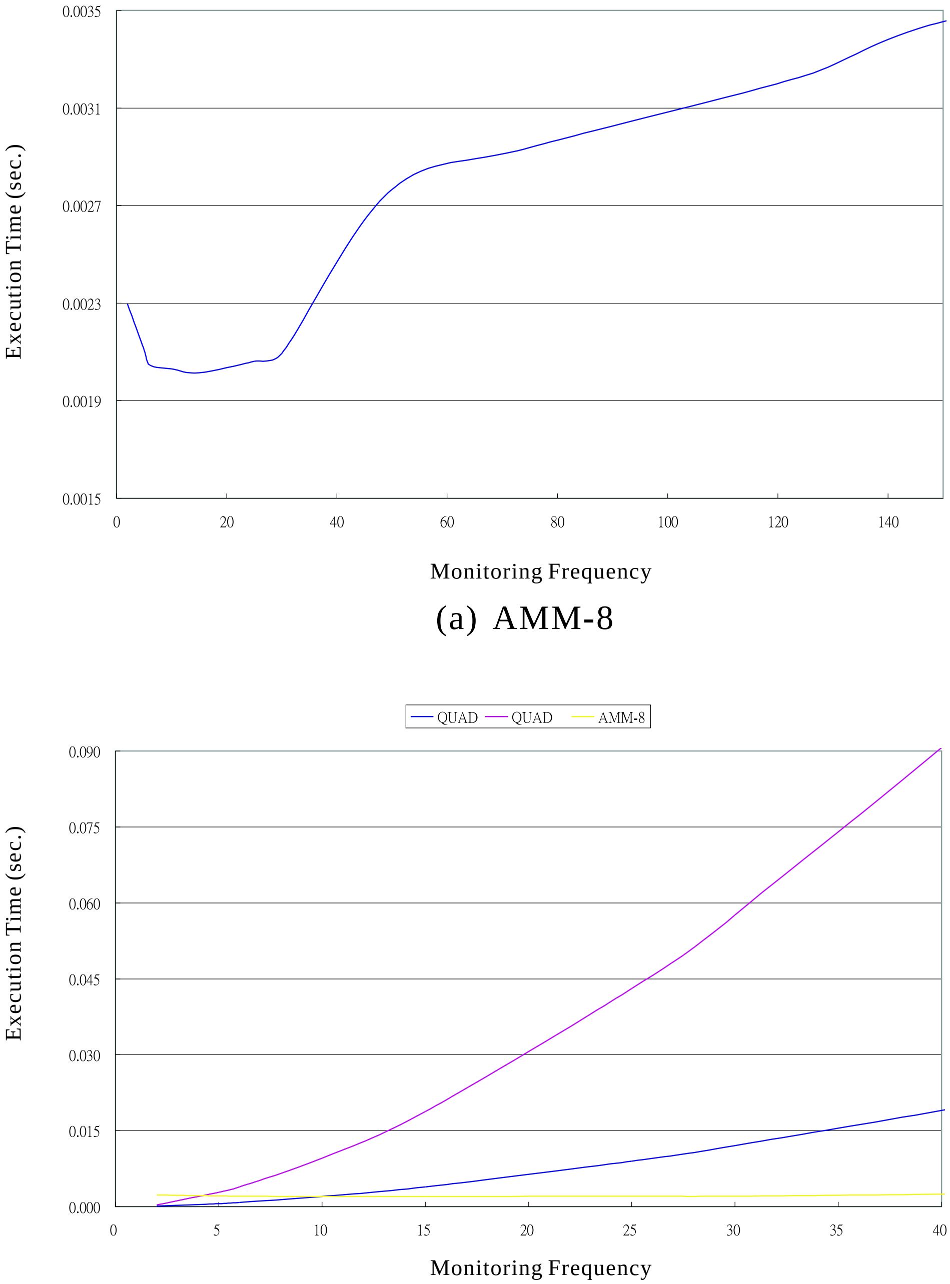
[c] The QUAD$_{\text{ext}}$ is the Richardson extrapolation of the quadrature method results in $\delta y_1 = \sqrt{\Delta t}/5$ and $\delta y_2 = \sqrt{\Delta t}/10$.

[d] The values of AMM-8 is calculated by AMM level 8 with the number of time steps $n = 750$.

[e] All the values are rounded off to the forth decimal place.

[f] The error(%) field is the percentage pricing error = [approximation/(benchmark)−1]100% rounded to the forth decimal place with all the values computed before rounding.

Table 4.6: **Numerical comparisons of AMM with the quadrature method in single discrete barrier options.**



(a) AMM-8

(b) QUAD, QUAD$_{\text{ext}}$, and AMM-8

Figure 4.8: **The frequency-time chart for single barrier options in QUAD, QUAD$_{\text{ext}}$, and AMM-8.**
(a)AMM-8. (b)QUAD, QUAD$_{ext}$, and AMM-8. Plotting with parameters $S = 100$, $X = 100$, $L = 80$, $\sigma = 25\%$, $r = 5\%$, $T = 0.5$ year $q = 0\%$, and $n = 750$ for AMM-8.

QUAD exceed AMM-8 around $F = 10$ and QUAD$_{\text{ext}}$ is explicit slower than AMM-8 when $F > 5$.

Table 4.7 and Fig. 4.9 are comparisons in the case of double barrier options. The AMM-8 are calculated with $n = 1,500$. QUAD-K20 and QUAD-K30 are the quadrature method with $K = 20$ and $K = 30$ in $\delta y = (h-l)/K$. The benchmark in Table 4.7 comes from quadrature method with $K = 200$. We can see from the percentage error fields in Table 4.7 that QUAD-K30 is more precise than AMM-8 except for the case of $F = 500$ and $L = 80$, and QUAD-20 is also more accurate than AMM-8 in most of the cases. There is one thing should be noteworthy that we release the constraint of $K > (h-l)/\sqrt{\Delta t}$ in this table to see what would happen when the region of $h - l$ is not divided fine enough. All of the released outcomes are marked a star. We can find that the stared answers which would be generally more erroneous show up in those cases with higher monitoring frequency and larger price gap between H and L. The larger price gap between H and L enlarges $h - l$ and higher monitoring frequency decreases $\Delta t(= T/M)$. Both two factors confine K to be a larger number which would reduce the efficiency of quadrature method. The situation is even worse in the increase of monitoring frequency. As we can see in Fig. 4.9(b), higher monitoring frequency largely raises execution time of quadrature method against AMM-8. However, as the increase of monitoring times, an larger K is required to come up with an accurate enough result, which would make efficiency of the quadrature method worse.

Fig. 4.9(a) is a frequency-time curve of AMM-8. As in the single barrier case, the optimization increase efficiency of AMM-8 at first, but the computation amount of extra fine mesh nodes offsets the efficiency gain while the monitoring frequency increases. Fig. 4.9(b) compares AMM-8 with QUAD-K20 and QUAD-K30. We can see from this figure that the efficiency of all methods reduces while barrier monitoring times increase. However, the increase rates of execution time in the three methods are different that AMM-8 is the smaller than QUAD-K20 which has a slower increase rate than QUAD-K30.

Barrier		**Benchmark**[a]	**QUAD-K20**[b]		**QUAD-K30**[c]		**AMM-8**[d]	
H	L		value[e]	error(%)[f]	value	error(%)	value	error(%)
			monitoring frequency=5					
	80	2.4499	2.4499	0.0000	2.4499	0.0000	2.4499	−0.0026
	90	2.2028	2.2028	0.0000	2.2028	0.0000	2.2027	−0.0020
120	95	1.6831	1.6831	0.0000	1.6831	0.0000	1.6830	−0.0099
	99	1.0811	1.0811	0.0000	1.0811	0.0000	1.0811	−0.0019
	99.9	0.9432	0.9432	0.0000	0.9432	0.0000	0.9433	0.0057
			monitoring frequency=25					
	80	1.9420	1.9420	−0.0014	1.9420	−0.0001	1.9419	−0.0063
	90	1.5354	1.5354	−0.0009	1.5354	−0.0001	1.5353	−0.0071
120	95	0.8668	0.8668	−0.0005	0.8668	0.0000	0.8668	−0.0063
	99	0.2931	0.2931	0.0002	0.2931	0.0000	0.2932	0.0091
	99.9	0.2023	0.2023	0.0002	0.2023	0.0000	0.2024	0.0261
			monitoring frequency=125					
	80	1.6808	*1.6803	*−0.0281	1.6808	−0.0025	1.6807	−0.0081
	90	1.2029	1.2026	−0.0212	1.2029	−0.0012	1.2028	−0.0076
120	95	0.5532	0.5531	−0.0173	0.5532	−0.0010	0.5531	−0.0039
	99	0.1042	0.1042	−0.0075	0.1042	−0.0004	0.1043	0.0816
	99.9	0.0513	0.0513	−0.0027	0.0513	−0.0002	0.0513	0.1074
			monitoring frequency=250					
	80	1.6165	*1.8581	*14.9440	*1.6164	*−0.0089	1.6163	−0.0141
	90	1.1237	*1.1234	*−0.0328	1.1237	−0.0041	1.1236	−0.0144
120	95	0.4867	*0.4864	*−0.0602	0.4867	−0.0035	0.4867	−0.0049
	99	0.0758	0.0758	−0.0396	0.0758	−0.0022	0.0759	0.1153
	99.9	0.0311	0.0311	0.0103	0.0311	−0.0006	0.0311	0.1575
			monitoring frequency=500					
	80	1.5706	*651.1038	*41355.3859	*1.5712	*0.0394	1.5701	−0.0335
	90	1.0680	*1.4357	*34.4207	*1.0679	*−0.0134	1.0676	−0.0410
120	95	0.4420	*0.4475	*1.2413	0.4420	−0.0111	0.4419	−0.0418
	99	0.0593	*0.0592	*−0.1755	0.0593	−0.0101	0.0600	1.2477
	99.9	0.0198	*0.0198	*−0.0188	0.0198	−0.0018	0.0198	0.0786

It is an double knock-out call with $T = 0.5$ year, $r = 5\%$, $q = 0\%$, $\sigma = 25\%$, $S = 100$, and $X = 100$.

[a] The Benchmark comes from the quadrature method with $K_m = 200$, where $m = 1, 2, \cdots, M$.

[b] The QUAD-K20 is the quadrature method with $K_m = 20$, where $m = 1, 2, \cdots, M$.

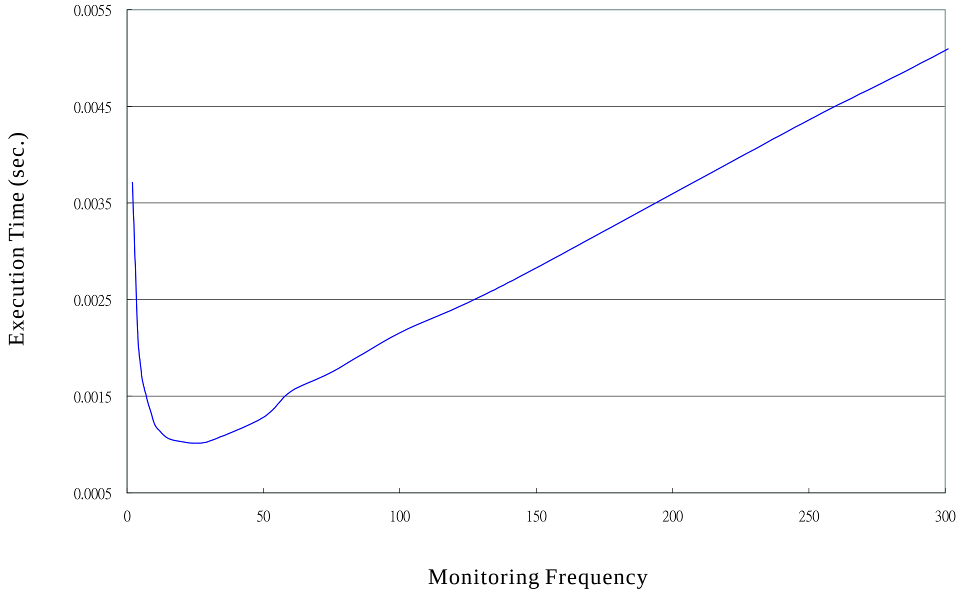
[c] The QUAD-K30 is the quadrature method with $K_m = 30$, where $m = 1, 2, \cdots, M$.

[d] The values of AMM-8 is calculated by AMM level 8 with the number of time steps $n = 1500$.

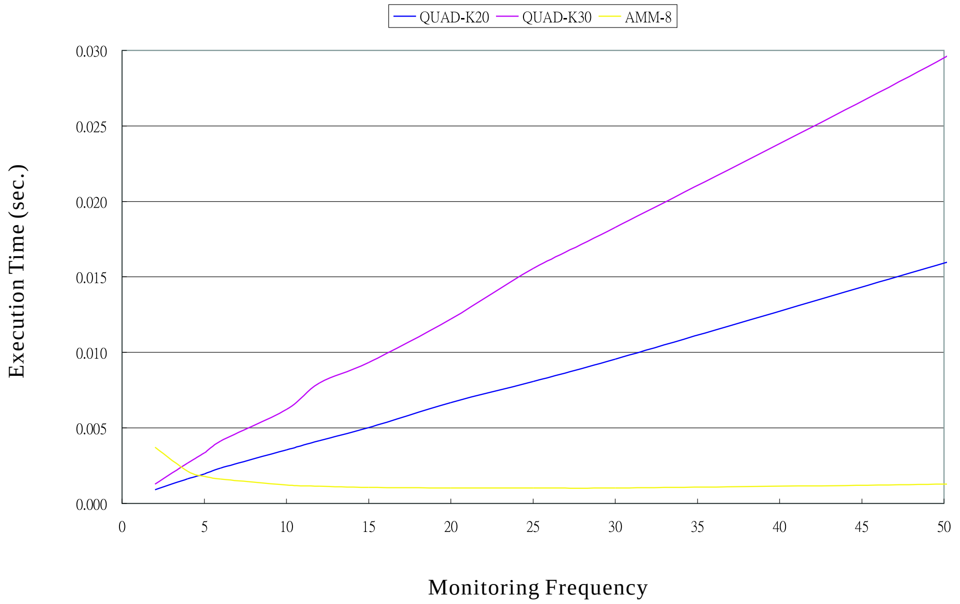
[e] All the values are rounded off to the forth decimal place.

[f] The error(%) field is the percentage pricing error = [approximation/(benchmark)−1]100% rounded to the forth decimal place with all the values computed before rounding.

Table 4.7: **Numerical comparisons of AMM with the quadrature method in double barrier options.**



(a) AMM-8



(b) QUAD-K20, QUAD-K30, and AMM-8

Figure 4.9: **The frequency-time chart for double discrete barrier options in QUAD-K20, QUAD-K30, and AMM-8.**
(a)AMM-8. (b)QUAD-K20, QUAD-K30, and AMM-8. Plotting with parameters $S = 100$, $X = 100$, $H = 120$, $L = 95$, $\sigma = 25\%$, $r = 5\%$, $T = 0.5$ year $q = 0\%$, and $n = 1500$ for AMM-8.

Chapter 5

Conclusions

This thesis does not only implement the Adaptive Mesh Model for single discrete barrier options but also extend the Adaptive Mesh Model to price double discrete barrier options. Besides, we also comprehensively implement other competitive methods with detailed numerical results to compare with AMM such as the standard trinomial tree, Ritchken's trinomial lattice, the enhanced trinomial tree, the BGK formula approach, and the quadrature method. Reducing nonlinearity error by applying higher resolution lattices in critical area makes AMM converge to accurate option price more efficiently than other tree lattice methods. Comparing with the BGK formula approach, AMM is more precise under the barrier-too-close situation and lower barrier monitoring frequency. Although the quadrature method is generally more exact than AMM, AMM can beat the quadrature method in efficiency under higher barrier monitoring frequency with accurate enough outcomes. From our research data, we numerically prove the accuracy and the efficiency of the Adaptive Mesh Model no matter where the barrier price is or what barrier monitoring frequency it is.

Bibliography

[1] Merton, R. (1973) Theory of Rational Option Pricing. *Bell Journal of Economics & Management*

[2] Reiner, E., and Rubinstein, M. (Sep 1991) Breaking Down the Barriers. *Risk 4*, 8, pp. 28-35.

[3] Reiner, E. (2000) *Convolution Methods for Path-Dependent Options.* Preprint, UBS Warburg Dillon Read.

[4] Heynen, R.C., and Kat, H. (1996) Discrete Partial Barrier Options with a Moving Barrier. *Journal of Financial Engineering*, 5, pp. 199-209.

[5] Broadie, M., Glasserman, P., and Kou, S.G. (1997) A Continuity Correction For Discrete Barrier Options. *Mathematical Finance* , 7, pp. 325-49.

[6] Boyle, P., and Lau, S.H. (1994) Bumping Against the Barrier with the Binomial Method. *Journal of Derivatives*, 1, pp. 6-14.

[7] Hull, J. (2002) *Options, Futures and Other Derivatives*, 5th Edition, Prentice-Hull.

[8] Derman, E., Kani, I., Ergener, D., and Bardhan, I. (1995) Enhanced numerical methods for options with barriers. *Goldman Sachs Quantitative Strategies Research Notes.*

[9] Ritchken, P. (1995) On Pricing Barrier Options. *Journal of Derivatives*, 3, pp. 19-28.

[10] Cheuk, T.H.F., and Vorst, T.C.F. (1995) Complex Barrier Options. *Journal of Derivatives*, 4, pp. 8-22.

[11] Broadie, M., Glasserman, P., and Kou, S.G. (1999) Connecting discrete and continuous path-dependent options. *Finance Stochastics*, 3, pp. 55-82.

[12] Wei, J. (1998) Valuation of Discrete Barrier Options by Interpolations. *Journal of Derivatives*, 6, pp. 51-73.

[13] Figlewski, S., and Gao, B. (1999) The Adaptive Mesh Model: A New Approach to Efficient Option Pricing. *Journal of Financial Economics*, 53, pp. 313-351.

[14] Ahn, D-G., Figlewski, S., and Gao, B. (1999) Pricing Discrete Barrier Options with an Adaptive Mesh Model. *Journal of Derivatives*, 6(4), pp. 33-44.

[15] Andricopoulos, A.D., Widdicks, M., Duck, P.W., and Newton, D.P. (2003) Universal option valuation using quadrature methods. *Journal of Financial Economics*, 67, pp. 447-471.

[16] Tavella, D., and Randall, C. (2000) *Pricing Financial Instruments*, 1th Edition, John Wiley & Sons.

[17] Kamrad, B., and Ritchken, P. (1991) Multinomial Approximating Models for Options with k-State Variables. *Management Science*, 37, 12, pp. 1640-1652.

[18] Boyle, P. (1988) A lattice framwork for option pricing with two state varaibles. *Journal of Financial and Quantitative Analysis*, 35, pp. 1-12.

[19] Omberg, E. (1988) Efficient discrete time jump process models in option pricing. *Journal of Financial and Quantitative Analysis*, 23, pp. 161-174.

[20] Lyuu, Y. (2002) *Financial Engineering and Computation*, 1th Edition, Cambridge University Press.

[21] Tsai, Shao-Hwai. (2004) Pricing discrete double-barrier options with the quadrature method. Master's Thesis. Department of Finance, National Taiwan University, Taiwan.

[22] Ko, Kun-Yi. (2003) Fast accurate option valuation using Gaussian quadrature. Master's Thesis. Department of Finance, National Central University, Taiwan.

[23] Moler, C. (2004) *Numerical Computing with MATLAB*, Mathworkd, New York.