

The Bino-trinomial Tree: A Simple Model for Efficient and Accurate Option Pricing

Tian-Shyr Dai

Associate Professor

Dept. Information &
Finance Management

Institute of Finance

National Chiao-Tung University,
1001 Ta Hsueh Road,
Hsinchu 300, Taiwan.

886-3-5712121x57054 (o)

886-3-5715544 (fax)

d88006@csie.ntu.edu.tw

Yuh-Dauh Lyuu

Professor

Dept. Computer Science &
Information Engineering

Dept. Finance

National Taiwan University
No. 1, Sec. 4, Roosevelt Rd.
Taipei, Taiwan

886-2-23625336 x429 (o)

886-2-23628167 (fax)

lyuu@csie.ntu.edu.tw

Abstract

Most derivatives do not have simple valuation formulas and must be priced by numerical methods such as tree models. Although the option prices computed by a tree model converge to the theoretical value as the number of time steps increases, the distribution error and the nonlinearity error may make the prices converge slowly or even oscillate significantly. This paper introduces a novel tree model, the bino-trinomial tree (BTT), that can price a wide range of derivatives efficiently and accurately. The BTT reduces the nonlinearity error sharply by adapting its structure to suit the derivative's specification; consequently, the pricing results converge smoothly and quickly. Moreover, the pricing of some European-style options on the BTT can be made extremely efficient by combinatorial tools, which are not available to most other tree models. Therefore, the BTT can efficiently reduce the distribution error by picking a large number of time steps. This paper uses a variety of options to demonstrate the effectiveness

of the BTT. Extensive numerical experiments show the superiority of the BTT to many other popular numerical models.

Keywords: bino-trinomial tree, nonlinearity error, distribution error, tree, option pricing

1 Introduction

With the rapid growth of financial markets, more sophisticated derivatives are constantly being structured by financial institutions to satisfy the needs of their clients. Financial innovations make markets more efficient, but they also give rise to pricing problems. Outside a small group of financial derivatives, simple yet exact analytical formulas — such as the celebrated Black-Scholes (1973) formula for vanilla options — do not exist. For such derivatives, even if approximation formulas are available, they may lead to large pricing errors. Take a continuously monitored (or simply “continuous” for short) double-barrier option as an example. Its payoff depends on whether the underlying asset’s price path ever touches either of two price levels (the barriers) before maturity. No simple, exact closed-form pricing formulas are available for this option. Analytical approximation formulas have been studied by Kunitomo and Ikeda (1992), Geman and Yor (1996), and Sidenius (1998), but they can lead to large pricing errors (see Luo (2001)). Moreover, the analytical formulas can not be easily extended to price American-style options or options with nonstandard payoff functions such as some power-type payoff functions. Developing efficient numerical methods to price those derivatives is obviously critical.

The tree model is a popular numerical pricing method (see Lyuu (2002)). It divides the time span from now to the option’s maturity date into n time steps and specifies the stock prices discretely at each time step. Their probability distribution must match the underlying process’s distribution asymptotically. The tree is flexible in that an option can be priced with only nominal changes when its payoff function is nonstandard; in contrast, certain power options do not have closed-form pricing formulas (see Zhang (1998)).

The option prices computed by tree models converge to the theoretical option value under the continuous-time model as n tends to infinity (see Duffie (1996)). However, the prices may converge slowly or even oscillate significantly, especially for the popular barrier-type options (see Boyle and Lau (1994)). Figlewski and Gao (1999) identify two types of errors introduced by a discrete-time discrete-state tree model: the distribution error and the nonlinearity error. The distribution error arises from approximating the continuous distribution of the stock price with a discrete distribution. Fortunately, it converges to zero

as $n \rightarrow \infty$. The nonlinearity error, on the other hand, is introduced by the nonlinearity of the option value function. The nonlinearity occurs at certain critical locations such as a certain point, a price level, or a time point (called a critical point, a critical price level, or a critical time point, respectively, in the paper). Critical locations are straightforward to identify. For example, the vanilla option has a critical point at maturity with the stock price equal to the strike price. For continuous barrier options, the critical price level occurs along the barrier price. For discrete-monitoring barrier options, the critical time points occur at monitoring dates.

Figlewski and Gao (1999) argue that the pricing results oscillate significantly due mainly to the nonlinearity error. They propose the adaptive mesh model (AMM) to suppress price oscillations. The AMM can be roughly viewed as a combination of two types of trinomial trees: the base tree and the finer ones. The resolution of the base tree is low for efficiency concerns. The finer trees, in contrast, have higher resolution and are built for only part of the base tree where the nonlinearity error is significant. The complicated structure of the AMM, however, makes it difficult to implement and still harder to tailor to different derivatives.

The nonlinearity error can alternatively be reduced by restructuring the tree to make the critical points, critical price levels, or critical time points coincide with the tree's nodes, layers, or discrete time steps, respectively. This idea is first suggested by Ritchken (1995). He reduces the nonlinearity error in pricing continuous barrier options by aligning a layer of nodes of the trinomial tree with each barrier. Similar ideas can also be found in Broadie and Detemple (1996), Cheuk and Vorst (1996,1997), Tian (1999), Widdicks et al. (2002), Ait-Sahlia et al. (2004), and Chung and Shih (2007). These models are simple and easy to implement. However, as more critical locations are added in complicated derivative contracts, the aforementioned models become too inflexible to deal with such instruments.

Andricopoulos *et al.* (2003) propose the quadrature method (QUAD for convenience), which has a multinomial tree structure. QUAD is flexible in that it can let nodes or discrete time steps coincide with critical points or critical time points, respectively. QUAD is especially efficient for pricing discretely monitored (or simply "discrete" for short) options as it deploys only one time step between two monitoring dates; traditional tree models need

more time steps to attain good results. QUAD applies Simpson’s rule and extrapolation to ensure fast convergence and accuracy. But QUAD is not as efficient as traditional tree models in handling such continuous sampling feature as the American exercise feature and the continuous-monitoring feature of continuous barrier options. This is because both QUAD and the binomial/trinomial tree models require large numbers of time steps to approximate the continuous sampling feature well. And it costs much more time to compute with the multinomial tree structure of QUAD than the binomial/trinomial tree when the number of time steps is large. Numerical results will be given later to support this claim.

This paper proposes a novel and simple tree model, the bino-trinomial tree (BTT), that can price a wide range of derivatives efficiently and accurately. The BTT is essentially a binomial tree with occasional trinomial structures for added flexibility. To reduce the nonlinearity error, the BTT can adapt its structure to deal with critical locations; consequently, the pricing results converge smoothly with only minor oscillations. For further improvement, the smoothing technique suggested by Heston and Zhou (2000) can be used. The binomial part of the BTT is the Cox-Ross-Rubinstein (CRR) binomial tree (see Cox *et al.* (1979)). Thus European options can be priced extremely efficient by the combinatorial tools designed for such trees (see Lyuu (1998) and Dai *et al.* (2008)). As a result, the distribution error can be drastically reduced as the BTT is very efficient, even with a large number of time steps. Numerical results given in this paper will show that the BTT achieves the same accuracy level with far less computational time than all other tree structures.

The BTT derives the flexibility from its ability to combine a more fundamental tree structure, the basic BTT (bBTT hereafter). The bBTT is essentially a binomial tree except for a trinomial structure at the first time step, which provides the needed flexibility to deal with critical locations. For example, Exhibit 1 outlines a bBTT to price a continuous double-barrier option with two barriers, L and H . The nonlinearity error is reduced by having two price layers of the bBTT coincide with the barriers L and H . The CRR tree, which, in gray and with the first two time steps truncated, comprises the bulk of the bBTT. Each node of the CRR tree lies on a grid point. The height of each cell of the grid equals the distance between two adjacent price levels of the CRR tree; the length of each cell equals the length of a time step (Δt). We first adjust the height of each cell so that the distance

between the barriers is an integer multiple of the height of a cell. Then we lay out the grid from the barrier L upward; barrier H will be on a layer of the grid automatically. The length of the first time step of the bBTT, $\Delta t'$, may be slightly larger than Δt to make the bBTT span the whole time span T . A method for selecting nodes A , B , and C from the light gray vertices of the grid at time $\Delta t'$ guarantees valid branching probabilities from the root node S (i.e., P_u , P_m , and P_d). The truncated CRR tree is finally laid on top of the grid by emanating from nodes A , B , and C . The bBTT can be efficiently calculated by the combinatorial tools designed for the CRR tree in pricing many European-style options.

The bBTT provides only two degrees of freedom in adapting the tree structure: (1) the height of a cell and (2) the position of the grid. For pricing more complex options, say a discrete moving-double-barrier option, more degrees of freedom are needed to make the tree align with all critical locations. This can be achieved by combining multiple bBTTs to form a single tree, called the BTT. Exhibit 2 illustrates how we combine four bBTTs, emanating from S , D , E , and F , into one BTT for pricing a discrete moving-double-barrier knock-out option with barriers H_0 and L_0 at time T_0 and H_1 and L_1 at time $T_0 + T_1$. Each bBTT is constructed as we did in the last paragraph, and the BTT aligns with all discrete barriers (marked by the black nodes) to reduce the nonlinearity error. Two truncated CRR trees, one starting at $\Delta t'_0$ and the other starting at $T_0 + \Delta t'_1$, laid on the top of the grids comprise the bulk of the BTT. Again, the BTT can be efficiently calculated by the combinatorial techniques designed for the CRR tree.

Our paper is organized as follows. The assumptions of the stock price process and the definitions for the options to be discussed in this paper are given in section 2. The methodology to construct the bBTT is detailed in section 3. To price an option that needs more degrees of freedom than the bBTT (such as moving barrier options), the BTT is needed and detailed in section 4. Numerical results are provided in section 5 to verify the superiority of our methods compared with many others. Section 6 concludes the paper.

2 Basic Terms and Preliminaries

Let S_t denote the stock price at time t , where $0 \leq t \leq T$. S_t follows the lognormal diffusion process:

$$S_{t+dt} = S_t \cdot \exp[(r - 0.5\sigma^2) dt + \sigma dW_t], \quad (1)$$

where W_t is the standard Wiener process, r is the risk-free interest rate per annum, and σ is the volatility of the stock price. We assume that the option initiates at time 0 (with stock price S_0) and matures at time T (with stock price S_T). The exercise price for this option is denoted by X .

A vanilla option gives its owner the right to buy or sell the underlying stock for the exercise price X and does not have other unusual features. The payoff of a European vanilla option at maturity date T is $\max(\theta(S_T - X), 0)$, where $\theta = 1$ for call options and -1 for put options. The exercise price X at maturity is the critical point as the payoff function is highly nonlinear at X . An American option allows the option holder to exercise the option early. The exercise value for an American option at time t ($0 \leq t \leq T$) is $\theta(S_t - X)$.

A barrier option is an option whose payoff depends on whether the underlying stock's price path ever touches certain price levels called the barriers. A knock-in barrier option comes into existence if the stock price touches the barrier(s) before the maturity date, whereas a knock-out one ceases to exist if the stock price touches the barrier(s) before maturity. Our paper focuses on knock-out barrier options since the value of a knock-in barrier option can be derived from the knock-out barrier option via the in-out parity. For a continuous barrier option, the underlying stock price is monitored continuously from time 0 to time T . For example, the payoff of a continuous down-and-out single-barrier option with a low barrier L is

$$\text{Payoff} = \begin{cases} \max(\theta(S_T - X), 0), & \text{if } S_{\text{inf}} > L, \\ 0, & \text{otherwise,} \end{cases}$$

where $S_{\text{inf}} = \inf_{0 \leq t \leq T} S_t$. On the other hand, the payoff of a continuous up-and-out single-

barrier option with a high barrier H is

$$\text{Payoff} = \begin{cases} \max(\theta(S_T - X), 0), & \text{if } S_{\text{sup}} < H, \\ 0, & \text{otherwise,} \end{cases}$$

where $S_{\text{sup}} = \sup_{0 \leq t \leq T} S_t$. The payoff of a continuous double-barrier option with a low barrier L and a high barrier H is

$$\text{Payoff} = \begin{cases} \max(\theta(S_T - X), 0), & \text{if } S_{\text{sup}} < H \text{ and } S_{\text{inf}} > L, \\ 0, & \text{otherwise.} \end{cases}$$

The prices L and H are the critical price levels as the option value freezes at zero once the stock price reaches L or H .

The payoff of a discrete barrier option depends on whether the stock price is above (or below) the barrier(s) at certain predetermined dates called the monitoring dates. Assume the barriers at times T_1, T_2, \dots, T_m are L_1, L_2, \dots, L_m , respectively. Then the payoff of a discrete moving-single-barrier down-and-out option is

$$\text{Payoff} = \begin{cases} \max(\theta(S_T - X), 0), & \text{if } S_{T_i} > L_i \text{ for } 1 \leq i \leq m, \\ 0, & \text{otherwise.} \end{cases}$$

Similarly, the payoff of a discrete moving-double-barrier knock-out option with high barrier H_i and low barrier L_i at time T_i ($1 \leq i \leq m$) is

$$\text{Payoff} = \begin{cases} \max(\theta(S_T - X), 0), & \text{if } H_i > S_{T_i} > L_i \text{ for } 1 \leq i \leq m, \\ 0, & \text{otherwise.} \end{cases}$$

The barrier prices L_1 and H_1 at time T_1 , L_2 and H_2 at time T_2 , and so on are critical points as the option value freezes at zero when the stock price is lower than L_i or higher than H_i at time T_i , where $1 \leq i \leq m$.

A tree model divides the time interval from time 0 to time T into n time steps and specifies the stock price at each time step. A tree converges to the stock price process mentioned in Eq. (1) if both the first and second moments of the stock price process are asymptotically matched at each node of the tree (see Duffie (1996)). Consider the CRR tree in Exhibit 3. From an arbitrary node with stock price S , the stock price after one time step equals Su (the up move) with probability p and Sd (the down move) with probability

$1 - p$, where $d < u$ and $ud = 1$. To match the first two moments of the stock price process, the CRR tree sets $u \equiv e^{\sigma\sqrt{T/n}}$, $d \equiv e^{-\sigma\sqrt{T/n}}$, and $p \equiv (e^{rT/n} - d)/(u - d)$. Note that the property $ud = 1$ is utilized to develop many efficient combinatorial algorithms for pricing a large variety of options on the CRR tree (see Lyuu (1998) and Dai *et al.* (2008)). The stock price S resulting from j down moves and $i - j$ up moves from time step 0 equals $S_0 u^{i-2j}$ with probability $\binom{i}{j} p^{i-j} (1-p)^j$.

3 Construction of the Basic Bino-Trinomial Tree (bBTT)

This section shows how to construct a bBTT for pricing options that need two or fewer degrees of freedom to align with all critical locations. We address continuous double-barrier options, continuous single-barrier options, and vanilla options, in that order as each is a special case of its predecessor.

Continuous Double-Barrier Options

The bBTT for pricing a continuous double-barrier option with two barriers L and H is depicted in Exhibit 4, which is simplified in Exhibit 1. The constituent CRR tree colored in gray is laid on a grid. The first two time steps of the CRR tree are truncated. As a result, this truncated CRR tree emanates from three nodes: A , B , and C at time $\Delta t'$. They are connected to node S at time 0 with branching probabilities P_u , P_m , and P_d . To price a continuous double-barrier option accurately and efficiently, the bBTT should possess the following two features: (1) Two layers of bBTT coincide with L and H so that the nonlinearity error is sharply reduced, and (2) the branching probabilities P_u , P_m , and P_d are valid (i.e., $0 \leq P_u, P_m, P_d \leq 1$). Given an integer m , we now proceed to construct a bBTT with approximately m time steps that has the aforementioned properties.

Define the stock price for node X as S_X and the V -log-price of stock price V' as $\ln(V'/V)$. Thus the V -log-price of z implies a stock price of Ve^z . The S_S -log-prices of the two barriers H and L are $h \equiv \ln(H/S_S)$ and $\ell \equiv \ln(L/S_S)$, respectively. The width of a cell of the grid equals Δt , which is the length of a time step of the CRR tree. Each node of the CRR tree is laid on the intersection of a vertical line and a horizontal line of the grid. The height of

a cell is $\sigma\sqrt{\Delta t}$ since the upward and the downward additive factors of the V -log-prices on the CRR tree are $\sigma\sqrt{\Delta t}$ and $-\sigma\sqrt{\Delta t}$, respectively. Note that the difference between the V -log-prices of two adjacent nodes such as nodes A and B in Exhibit 4 is $2\sigma\sqrt{\Delta t}$.

We now show how to choose the width of a cell, Δt , to make the grid hit both H and L . For the grid to have two layers coinciding with H and L , $\frac{h-\ell}{2\sigma\sqrt{\Delta t}}$ must be some integer k . For example, $k = 4$ in Exhibit 4. Although $\Delta\tau \equiv T/m$ is a natural choice for the length of each time step for an m -time-step bBTT, the problem is $\frac{h-\ell}{2\sigma\sqrt{\Delta\tau}}$ may not be an integer. So, instead, we pick a Δt that is close to, but does not exceed, $\Delta\tau$ and that makes $\frac{h-\ell}{2\sigma\sqrt{\Delta t}}$ an integer:

$$\Delta t = \left(\frac{h-\ell}{2\kappa\sigma} \right)^2, \quad (2)$$

where $\kappa = \left\lceil \frac{h-\ell}{2\sigma\sqrt{\Delta\tau}} \right\rceil$. Now, lay out the grid from barrier L upward. Automatically, a layer coincides with barrier H because of the integrality condition. The number of time steps of the bBTT is $\lfloor \frac{T}{\Delta t} \rfloor$ (which may be slightly larger than m) because the truncated CRR tree has $\lfloor \frac{T}{\Delta t} \rfloor - 1$ time steps. The length of the first time step of the bBTT, $\Delta t'$, is the remaining amount of time to make the whole tree span T years, i.e.,

$$\Delta t' = T - \left(\left\lfloor \frac{T}{\Delta t} \right\rfloor - 1 \right) \Delta t. \quad (3)$$

Clearly, $\Delta t \leq \Delta t' < 2\Delta t$. Note that primes are used to denote the first time step of the bBTT. The prime is dropped if the length of the first time step is equal to the length of each of the remaining time steps, like the bBTT for the continuous single-barrier option case (discussed later).

We still need to select nodes A , B , and C among the light gray vertices at time $\Delta t'$ in Exhibit 4 to make the branching probabilities from node S valid. These three nodes are connected to node S . Three branches are needed to match the first two moments of the logarithmic stock price process; a binomial branch simply does not have the needed degree of freedom. Define the mean function μ and the variance function Var as follows:

$$\begin{aligned} \mu(x) &\equiv (r - \sigma^2/2)x, \\ \text{Var}(x) &\equiv \sigma^2 x, \end{aligned}$$

The mean and the variance of the S_S -log-prices at nodes A , B , and C equal $\mu(\Delta t')$ and

$\text{Var}(\Delta t')$, respectively. The S_S -log-price for each light gray vertex at time $\Delta t'$ is

$$\begin{cases} \ell + 2j\sigma\sqrt{\Delta t}, & \text{if the truncated CRR tree has an even number of time steps,} \\ \ell + (2j + 1)\sigma\sqrt{\Delta t}, & \text{otherwise,} \end{cases} \quad (4)$$

for some integer j . For example, the truncated CRR tree in Exhibit 4 has 5 time steps, an odd number. As the difference between the S_S -log-prices of two adjacent light gray vertices is $2\sigma\sqrt{\Delta t}$, there must exist a unique vertex whose S_S -log-price lies in the interval $[\mu(\Delta t') - \sigma\sqrt{\Delta t}, \mu(\Delta t') + \sigma\sqrt{\Delta t})$, which has a length of $2\sigma\sqrt{\Delta t}$. Make this vertex node B . For example, the S_S -log-price of node B is $\ell + 3\sigma\sqrt{\Delta t}$ in Exhibit 4. Denote the S_S -log-price at node B as $\hat{\mu}$, which is closest to $\mu(\Delta t')$ among the S_S -log-prices of all the light gray vertices at time $\Delta t'$. We select nodes A and C from the light gray vertices that are adjacent to node B . Thus the S_S -log-prices of nodes A and C are $\hat{\mu} + 2\sigma\sqrt{\Delta t}$ and $\hat{\mu} - 2\sigma\sqrt{\Delta t}$, respectively. Define

$$\beta \equiv \hat{\mu} - \mu(\Delta t'), \quad (5)$$

$$\alpha \equiv \hat{\mu} + 2\sigma\sqrt{\Delta t} - \mu(\Delta t') = \beta + 2\sigma\sqrt{\Delta t},$$

$$\gamma \equiv \hat{\mu} - 2\sigma\sqrt{\Delta t} - \mu(\Delta t') = \beta - 2\sigma\sqrt{\Delta t}.$$

The first equation implies that $\beta \in [-\sigma\sqrt{\Delta t}, \sigma\sqrt{\Delta t})$. Note that $\alpha > \beta > \gamma$. The branching probabilities of node S (i.e., P_u, P_m, P_d) can be derived by solving the following three equalities:

$$P_u\alpha + P_m\beta + P_d\gamma = 0, \quad (6)$$

$$P_u\alpha^2 + P_m\beta^2 + P_d\gamma^2 = \text{Var}(\Delta t'), \quad (7)$$

$$P_u + P_m + P_d = 1. \quad (8)$$

Equations (6) and (7) match the first two moments of the logarithmic stock price, and Eq. (8) ensures that the probabilities P_u, P_m, P_d do sum to one. The above three equations indeed yield valid branching probabilities (see Appendix A).

The aforementioned model can be modified to align with the strike price by combining bBTTs with a procedure similar to the one in Section 4. However, mismatched strike price does not incur significant nonlinearity error (see Ritchken (1995)). Numerical experiments in

Section 5 validate this fact. Thus we omit the construction procedure to keep the discussion simple.

Continuous Single-Barrier Options

The bBTT for pricing continuous single-barrier options with barrier L follows a similar strategy and is illustrated in Exhibit 5. Again, the bBTT must have one layer coinciding with the barrier L to bring about reduction in nonlinearity error. This can be achieved by laying the underlying grid from barrier L upward as before. Note that there is no need to adjust the width of a cell of the grid, Δt , as we did for continuous double-barrier options before; we simply set $\Delta t = T/n$. This is because there is only one price level (L) for a layer of the bBTT to match. Note that the length of the first time step of the bBTT is also Δt since $T/\Delta t$ is already an integer n . Finally, select adjacent nodes A , B , and C (the successors of node S) among the light gray vertices at time Δt to make the branching probabilities P_u , P_m and P_d valid. Three nodes are needed for node S to match the first two moments of the logarithmic stock price process. By the lognormality of the stock price, the mean and the variance of the S_S -log-prices of A , B , and C equal $\mu(\Delta t)$ and $\text{Var}(\Delta t)$, respectively. The S_S -log-price of a light gray vertex at time Δt can be expressed as in Eq. (4). Again, we select the unique light gray vertex whose S_S -log-price lies in the interval $[\mu(\Delta t) - \sigma\sqrt{\Delta t}, \mu(\Delta t) + \sigma\sqrt{\Delta t}]$ as node B . Denote the S_S -log-price of B as $\hat{\mu}$. The S_S -log-prices of the two nodes A and C are again set to $\hat{\mu} + 2\sigma\sqrt{\Delta t}$ and $\hat{\mu} - 2\sigma\sqrt{\Delta t}$, respectively. Define α , β , and γ as in Eq. (5), where $\mu(\Delta t')$ is replaced by $\mu(\Delta t)$. The valid branching probabilities for node S can be solved by Eqs. (6)–(8), where $\text{Var}(\Delta t')$ is replaced by $\text{Var}(\Delta t)$.

Obviously, the aforementioned bBTT can also be made to match the strike price by adjusting the width of the cell Δt as we did in the last subsection. Numerical results in Section 5 shows that this adjustment improves the convergence.

Vanilla Options

To drastically reduce the nonlinearity error, the bBTT should have a layer coinciding with the exercise price X at the option's maturity date. This is because the payoff function of a

vanilla option is highly nonlinear at X at the maturity date (see Figlewski and Gao (1999)). Constructing a bBTT for pricing a vanilla option mimics that for a single-barrier option except that the underlying grid should now have a layer coinciding with X instead of L .

Combinatorial Algorithms

Pricing European-style barrier options and vanilla options on the bBTT can be made extremely efficient by the combinatorial pricing algorithms introduced in Lyuu (1998) and Dai *et al.* (2008) that are designed for the CRR tree. We will use the bBTT in Fig. 5 as an example to show how to efficiently evaluate a down-and-out single-barrier option by taking advantage of the combinatorial pricing algorithm of Lyuu (1998). Note that the outgoing branches from the root node S are connected to the three following nodes: A , B , and C . The trees emanating from these three nodes form three CRR trees. Lyuu's combinatorial algorithm can be used to efficiently evaluate the option values at these three nodes. Finally, the value at node S is calculated by the expected discounted value of these three values.

Lyyu's combinatorial algorithm first uses the reflection principle to count the number of paths that start from a node, say C , and then hit the barrier L before arriving at another node at maturity T , say D . For simplicity, we call a node at maturity T as a "terminal node". Assume that the starting node and the terminal node are a and b steps higher than the barrier, respectively, and there are \hat{n} time steps between these two nodes. By the reflection principle, the number of paths is

$$\binom{\hat{n}}{\frac{\hat{n}-a-b}{2}} \quad \text{for even, non-negative } \hat{n} - a - b \quad (9)$$

and zero otherwise. For example, the number of paths that start at C and hit the barrier L before arriving at D can be obtained by substituting $a = 1$, $b = 2$, and $\hat{n} = 3$ into Eq. (9) to obtain the count 1. Assume that the terminal node is reached from the starting node with i down movements and $\hat{n} - i$ up movements. Then the terminal node is $\hat{n} - 2i + a$ steps higher than the barrier. The option value contributed by the terminal node is $\binom{\hat{n}}{i-a} p^{\hat{n}-i} (1-p)^i e^{-r\hat{n}\Delta t} (\hat{S} u^{\hat{n}-i} d^i - X)^+$, where $\binom{\hat{n}}{i-a}$ is obtained by substituting $b = \hat{n} - 2i + a$ into Eq. (9), $p^{\hat{n}-i} (1-p)^i$ denotes the probability of each price path moving from the starting node to the terminal node, \hat{S} denotes the stock price of the starting node. Thus the option value

at the starting node is

$$\sum_{0 \leq i < \frac{\hat{n}+a}{2}} \binom{\hat{n}}{i-a} p^{\hat{n}-i} (1-p)^i e^{-r\hat{n}\Delta t} \left(\hat{S} u^{\hat{n}-i} d^i - X \right)^+, \quad (10)$$

where $\binom{\hat{n}}{i-a}$ is set to zero when $i-a < 0$. For example, the value at node B in Fig. (5) is obtained by substituting $\hat{n} = 3$, $a = 3$, and \hat{S} with the stock price at node B into Eq. (10). Note that the values at node A , B , and C can be efficiently calculated by Eq. (10). The option value at node S can then be evaluated by

$$V_S = e^{-r\Delta t} (P_u V_A + P_m V_B + P_d V_C),$$

where V_X denotes the option value at node X .

Similarly, double-barrier options and vanilla options can be efficiently computed by applying the combinatorial algorithms proposed in Lyuu (2002) and Dai *et al.* (2008) on the bBTTs.

4 Building a BTT from Multiple bBTTs

More degrees of freedom are required if a tree is to align with the critical locations of complex options. This is achieved by combining multiple bBTTs into one tree, called the BTT. We address discrete moving-double-barrier options and moving-single-barrier options, in that order as the latter is a special case of the former.

Discrete Moving-Double-Barrier Options

We now describe the BTT to price an option with two monitoring dates; extension to handle more monitoring dates is straightforward. The high barrier is H_0 at monitoring date T_0 and is H_1 at monitoring date $T_0 + T_1$. The low barrier is L_0 at monitoring date T_0 and is L_1 at monitoring date $T_0 + T_1$. See Exhibit 6 for illustration. The BTT rooted at S is constructed by combining 4 bBTTs, which emanate from nodes S , D , E , and F . Two truncated CRR trees comprise the bulk of the BTT. The first truncated CRR tree emanates from nodes A , B , and C at time $\Delta t'_0$. The width of a cell equals the length of a time step of this tree (Δt_0) and the height of a cell of the grid, c_0 , equals $\sigma\sqrt{\Delta t_0}$. The second truncated CRR

tree emanates from nodes $G, H, I, J,$ and K at time $T_0 + \Delta t'_1$. The width of a cell equals the length of a time step of this tree (Δt_1) and the height of a cell of the grid, c_1 , equals $\sigma\sqrt{\Delta t_1}$. To ensure that the option can be priced efficiently and accurately, the BTT should have the following properties: (1) It aligns with all discrete barriers (the black nodes), and (2) the probabilities for the trinomial branches at nodes $S, D, E,$ and F must be valid. Given an integer m , we proceed to construct a BTT with approximately m time steps that has the aforementioned properties. Ideally, the length of each time step for an m -time-step BTT is $\Delta\tau \equiv (T_0 + T_1)/m$; we will adjust the lengths for each bBTT ($\Delta t_0, \Delta t_1, \dots$) to make the BTT align with all critical locations as we did in Section 3.

We first focus on the part of the BTT that grows from time 0 to time T_0 , i.e., the bBTT emanating from node S . Constructing this bBTT mimics that for continuous double-barrier options in Section 3. To ensure the bBTT aligns with H_0 and L_0 at time T_0 , the distance (in S_S -log-price) between H_0 and L_0 , or $h_0 - \ell_0$, must be an integer multiple of $2c_0$, where $h_0 \equiv \ln(H_0/S_S)$ and $\ell_0 \equiv \ln(L_0/S_S)$. So we pick a Δt_0 such that $\frac{h_0 - \ell_0}{2\sigma\sqrt{\Delta t_0}}$ is an integer. Towards that end, we fix $\Delta t_0 = \left(\frac{h_0 - \ell_0}{2\kappa\sigma}\right)^2$, where $\kappa = \lceil \frac{h_0 - \ell_0}{2\sigma\sqrt{\Delta\tau}} \rceil$. Now, lay out a vertex of the grid from L_0 at time T_0 upward and another vertex of the grid will align with H_0 at time T_0 . The truncated CRR tree (growing from time $\Delta t'_0$ to T_0) has $\lfloor \frac{T_0}{\Delta t_0} \rfloor - 1$ time steps, and the length of the first time step $\Delta t'_0$ is the remaining amount of time to make the whole bBTT span T_0 years:

$$\Delta t'_0 \equiv T_0 - \left(\left\lfloor \frac{T_0}{\Delta t_0} \right\rfloor - 1 \right) \Delta t_0.$$

Select nodes $A, B,$ and C among the dark gray vertices at time $\Delta t'_0$ to make the branching probabilities from node S valid. The mean and the variance of the S_S -log-prices at nodes $A, B,$ and C are $\mu(\Delta t'_0)$ and $\text{Var}(\Delta t'_0)$, respectively. The S_S -log-price for a dark gray vertex in Exhibit 6 at time $\Delta t'_0$ can be expressed in Eq. (4), where ℓ and Δt are replaced by ℓ_0 and Δt_0 , respectively. Again, there must exist a unique vertex whose S_S -log-price in the interval $[\mu(\Delta t'_0) - \sigma\sqrt{\Delta t_0}, \mu(\Delta t'_0) + \sigma\sqrt{\Delta t_0}]$, which we choose for node B . Denote the S_S -log-price of B as $\hat{\mu}$. The S_S -log-prices of the flanking nodes A and C are set to $\hat{\mu} + 2\sigma\sqrt{\Delta t_0}$ and $\hat{\mu} - 2\sigma\sqrt{\Delta t_0}$, respectively. Define $\alpha, \beta,$ and γ as in Eq. (5), where $\mu(\Delta t')$ and Δt are replaced by $\mu(\Delta t'_0)$ and Δt_0 , respectively. Then the valid branching probabilities can be solved by Eqs. (6)–(8) with $\text{Var}(\Delta t')$ replaced by $\text{Var}(\Delta t'_0)$.

We proceed to construct the part of the BTT that grows from T_0 to T_0+T_1 . As before, to ensure that the BTT aligns with H_1 and L_1 at time T_0+T_1 , we pick $\Delta t_1 = \left(\frac{h_1-\ell_1}{2\kappa\sigma}\right)^2$, where $h_1 \equiv \ln(H_1/S_D)$ and $\ell_1 \equiv \ln(L_1/S_D)$ denote the S_D -log-prices of H_1 and L_1 , respectively, and $\kappa = \lceil \frac{h_1-\ell_1}{2\sigma\sqrt{\Delta\tau}} \rceil$. Lay out the grid from L_1 upward and a vertex of the grid will coincide with H_1 , automatically. The truncated CRR tree growing from time $T_0 + \Delta t'_1$ to $T_0 + T_1$ has $\lfloor \frac{T_1}{\Delta t'_1} \rfloor - 1$ time steps with $\Delta t'_1 \equiv T_1 - \left(\lfloor \frac{T_1}{\Delta t'_1} \rfloor - 1\right) \Delta t_1$.

At time step T_0 , the option is not knocked out at node D , E , or F . Thus we construct three bBTTs emanating from these three nodes. For each of these three nodes, we will select three successors from the light gray vertices at time $T_0 + \Delta t'_1$ to make the branching probabilities valid. Take node D for example. The S_D -log-price for a light gray vertex can be expressed in Eq. (4), where ℓ and Δt are replaced by ℓ_1 and Δt_1 , respectively. Again there exists a unique light gray vertex whose S_D -log-price lies in the interval $[\mu(\Delta t'_1) - \sigma\sqrt{\Delta t_1}, \mu(\Delta t'_1) + \sigma\sqrt{\Delta t_1})$. For example, this vertex is node H in Exhibit 6. Define the S_D -log-price for this node as $\hat{\mu}$. The S_D -log-prices of two other successors of D are set to $\hat{\mu} + 2\sigma\sqrt{\Delta t_1}$ and $\hat{\mu} - 2\sigma\sqrt{\Delta t_1}$, respectively. Define α , β , and γ as in Eq. (5), where $\mu(\Delta t')$ and Δt are replaced by $\mu(\Delta t'_1)$ and Δt_1 , respectively. Then the branching probabilities from node D can be solved by Eqs. (6)–(8) with $\text{Var}(\Delta t')$ replaced by $\text{Var}(\Delta t'_1)$.

The successors and the branching probabilities for nodes E and F can be derived in the same way. For example, the successors for node E are H , I , and J , and the successors for F are I , J , and K . The truncated CRR tree is then constructed by emanating from nodes G , H , I , J , and K at time $T_0 + \Delta t'_1$. To handle more monitoring dates, just apply the above procedure to each monitoring date.

Discrete Moving-Single-Barrier Options

Constructing a BTT to price discrete moving-single-barrier options follows a similar strategy. We focus on an option with barrier L_0 at time T_0 and barrier L_1 at time $T_0 + T_1$ in Exhibit 7. Again, the BTT must have two nodes coinciding with these two barriers to realize the reduction in the nonlinearity error. Assume there are m_0 time steps in the first part (from time 0 to time T_0) and m_1 time steps in the second part (from time T_0 to time $T_0 + T_1$) of the BTT. We first focus on the first part. There is no need to adjust Δt_0 , the

width of a cell of the thick dashed grid, since we only need one degree of freedom to align with barrier L_0 . So we simply set $\Delta t_0 \equiv T_0/m_0$. Lay out the grid from L_0 upward and select nodes A , B , and C as the successors of node S among the dark gray vertices at time Δt_0 to make the branching probabilities from node S valid. A truncated CRR tree then emanates from nodes A , B , and C .

Now we come to the second part of the BTT. The width of a cell of the thin dotted grid Δt_1 is set to T_1/m_1 . Lay out the grid from L_1 upward. Then we select the successors of nodes X , D , E , and F from the light gray vertices at time $T_0 + \Delta t_1$. These successors are Y , G , H , I , J , K . A truncated CRR tree then grows from these successor nodes.

5 Experimental Results

This section evaluates the performance of the BTT and other numerical methods in pricing vanilla options, single-barrier options, and double-barrier options. Running time measurements are obtained on a Pentium-4 2.8GHz computer.

Vanilla Options

We first compare the performance of the CRR model, the AMM, and the BTT in pricing vanilla call options. To improve the convergence rate, the BTT modifies the payoff at the strike price by incorporating the smooth option payoff function technique suggested by Heston and Zhou (2000) as follows:

$$\frac{1}{X(u-d)} \int_{Xd}^{Xu} (y-X)^+ dy.$$

In the setting of Exhibit 8, the theoretical option value given by the Black-Scholes formula is 17.7943. All models converge to the theoretical option value as $n \rightarrow \infty$; however, their behaviors differ. The CRR oscillates significantly. The AMM converges more smoothly as the AMM level increases. Finally, the BTT converges more smoothly and accurately than all the aforementioned methods.

Discrete Single- and Double-Barrier Options

Exhibit 9 demonstrates the convergence behavior of the BTT (with Heston and Zhou's (2000) idea for improving the convergence rate in place) in pricing a discrete single-barrier option. The extrapolated result 7.4517 accurately approximates the benchmark value 7.451115, which is computed by Monte Carlo simulation with 10,000,000 trials. Exhibit 10 compares the BTT, the AMM, and QUAD for pricing the same option. The sizes of these three models are carefully adjusted by varying K , m , and n so the computational times by these three models are roughly equal. QUAD converges more stably, to 7.452406, than the BTT and the AMM. But it seems to overvalue the option by $7.452406 - 7.451115 = 0.001291$. Both the maximum absolute error and the root-mean-squared error of the BTT are lower than the other two methods.

The BTT provides enough degrees of freedom to change the position of nodes at different time steps to reduce nonlinearity errors. Exhibit 11 demonstrates the convergence behavior of the BTT in pricing a discrete moving-double-barrier option, which is not considered by QUAD of Andricopoulos *et al.* (2003). The extrapolated result 0.2961 is very close to the benchmark value of 0.29616675, which is generated by Monte Carlo simulation with 1,000,000 trials.

Continuous Single- and Double-Barrier Options

Exhibit 12 compares the BTT and Ritchken's (1995) trinomial tree in pricing a European-style down-and-out single-barrier call option. The numerical settings and the true value are from Ritchken (1995). The x -axis and the y -axis denote the computational time and the option price, respectively. For example, it costs 0.014 second to compute with a 350-time-step Ritchken's trinomial tree (which can only be priced by the quadratic-time, standard backward induction algorithm) to obtain 5.998 (point A). It costs almost the same time to compute with a 4,500-time-step BTT (which can be done in linear time by applying combinatorial tools) to obtain 5.997 (point B, matching the barrier only) and 5.9968 (point C, matching both the strike price and the barrier), respectively. All models converge to the true value 5.9968, but the two BTTs converge more smoothly and faster than the Ritchken's model. Note that matching the strike price does improve convergence.

Andricopoulos *et al.* (2003) claim that QUAD can accurately price continuous barrier options by extrapolating the prices of discrete barrier options. But it seems their approach converges slowly: only at a rate of $O(m^{1/2})$, where m denotes the number of monitoring dates as illustrated in Exhibit 13. QUAD is not as efficient as either Ritchken’s model or the BTT. For example, it takes QUAD 14.593 seconds and 16.546 seconds to price a 950-monitoring-date barrier call and a 1000-monitoring-date barrier call, respectively. The prices for these two options are 6.370589 and 6.361381, respectively. Compared with Exhibit 12, the extrapolated result 6.006932035 of QUAD is not as accurate as either Ritchken’s model or the BTT.

The aforementioned observations raise a new issue: QUAD may not be efficient enough compared with traditional tree models in handling such continuous sampling feature as the continuous monitoring feature of the continuous barrier options and the American exercise feature, which is continuously exercisable. With that in mind, we compare Ritchken’s method, the BTT, and QUAD in pricing up-and-out American puts in Exhibit 14. Gao *et al.* (2000) provide an accurate quasi-analytical formula for pricing American barrier puts and we use it as the benchmark. The parameters for Ritchken’s model, QUAD and the BTT method are properly set to make the running times for all methods roughly equal. The results for QUAD are extrapolated by the prices of a 200-monitoring-date and a 400-monitoring-date barrier puts. The results for the BTT are extrapolated by the prices of a 1000-time-step and a 2000-time-step BTT. Note that both Ritchken’s model and the BTT can achieve 2-digit accuracy but not QUAD. The maximum absolute error (MAE) and the root-mean-squared error (RMSE) of QUAD are also larger than those of the other two methods.

Finally, we compare the performance of Ritchken’s trinomial model and the BTT in pricing a European-style double-barrier knock-out option in Exhibit 15. To our knowledge, no published papers discuss how to extend the AMM to accurately price continuous double-barrier options. The parameters and the accurate value of 1.4580 are from Ritchken (1995). All methods converge to the accurate value, but the two BTTs converge much more smoothly and faster than Ritchken’s model. Note that matching the strike price does not significantly improve convergence unlike what we observed in pricing single-barrier options.

The Barrier-Too-Close Problem

It is a well-known hard problem to price a barrier option efficiently when the barrier is very close to the initial stock price. This is the so-called barrier-too-close problem. Exhibit 16 compares Ritchken's model, the AMM, and the BTT for the barrier-too-close problem. Each row lists the number of time steps and the computational time required for each approach to achieve 3-digit accuracy. Both Ritchken's model and the BTT need a large number of time steps to ensure that the barrier is exactly hit. However, the BTT can be efficiently computed by combinatorial tools, whereas Ritchken's model can not. Thus the BTT can achieve 3-digit accuracy with much less computational time than Ritchken's model. Figlewski and Gao (1999) claim that the AMM can effectively solve the barrier-too-close problem since it requires fewer nodes to coincide with the barrier. The numerical results for the AMM in Exhibit 16 are computed by setting the AMM level to be 1. (The number of time steps of the AMM is determined by the AMM level.) To achieve 3-digit accuracy, the AMM again consumes more computation time than the BTT. This is because the efficiency improvement from algorithms (by replacing the slow quadratic-time backward induction algorithm with the efficient linear-time combinatorial algorithm) exceeds the efficiency improvement from reduction in the node count. We conclude that the BTT is superior to the AMM and Ritchken's trinomial tree model in addressing the barrier-too-close problem.

6 Conclusion

This paper proposes a novel, accurate, and efficient tree model for pricing a wide variety of derivatives: the bino-trinomial tree (BTT) model. The BTT is composed mostly of truncated CRR trees. The pricing results of the BTT converge smoothly and quickly since its structure can be adapted to suit the derivative's specification. Pricing on the BTT can furthermore be made extremely fast by applying combinatorial CRR tree pricing algorithms. Numerical results are given to confirm the superiority of the BTT over such methods as AMM, QUAD, and Ritchken's trinomial tree.

References

- [1] AIT-SAHLIA, F., IMHOF, H., AND LAI, T.-L. “Pricing and Hedging of American Knock-in Options.” *Journal of Derivatives*, 11 (2004), pp. 44–51.
- [2] ANDRICOPOULOS, A.D., WIDDICKS, M., DUCK, P.W., AND NEWTON, D.P. “Universal Option Valuation Using Quadrature Methods.” *Journal of Financial Economics*, 67 (2003), pp. 447–471.
- [3] BLACK, F., AND SCHOLES, M. “The Pricing of Options and Corporate Liabilities.” *Journal of Political Economy*, 81 (1973), pp. 637–659.
- [4] BOYLE, P., AND LAU, S. “Bumping against the Barrier with the Binomial Method.” *Journal of Derivatives*, 1 (1994), pp. 6–14.
- [5] BROADIE, M., AND DETEMPLE, J. “American Option Valuation: New Bounds, Approximations, and a Comparison of Existing Methods.” *Review of Financial Studies*, 9 (1996), pp. 1211–1250.
- [6] CHEUK, T.H.F., AND VORST, T.C.F. “Complex Barrier Options.” *Journal of Derivatives*, 4 (1996), pp. 8–22.
- [7] CHEUK, T.H.F., AND VORST, T.C.F. “Currency Lookback Options and Observation Frequency: A Binomial Approach.” *Journal of International Money and Finance*, 16 (1997), pp. 313–351.
- [8] CHUNG, S.L., AND SHIH, P.T. “Generalized Cox-Ross-Rubinstein Binomial Models.” *Management Science*, 53 (2007), pp. 508–520.
- [9] COX, J.C., ROSS, S., AND RUBINSTEIN, M. “Option Pricing: A Simplified Approach.” *Journal of Financial Economics*, 7 (1979), pp. 229–264.
- [10] DAI, T.-S., LIU, L.-M, AND LYUU, Y.-D. “Linear-Time Option Pricing Algorithms by Combinatorics.” *Computers and Mathematics with Applications*, 55 (2008), pp. 2142–2157.

- [11] DUFFIE, D. *Dynamic Asset Pricing Theory*. 2nd ed. Princeton, NJ: Princeton University Press, 1996.
- [12] FIGLEWSKI, S., AND GAO, B. “The Adaptive Mesh Model: A New Approach to Efficient Option Pricing.” *Journal of Financial Economics*, 53 (1999), pp. 313–351.
- [13] GAO, B., HUANG, J.Z., AND SUBRAHMANYAM, M. “The Value of American Barrier Options Using the Decomposition Technique.” *Journal of Economic Dynamics and Control*, 24 (2000), pp. 1783–1827.
- [14] GEMAN, H., AND YOR, M. “Pricing and Hedging Double-Barrier Options: A Probabilistic Approach.” *Mathematical Finance*, 6 (1996), pp. 365–378.
- [15] HESTON, S., AND ZHOU, G. “On the Rate of Convergence of Discrete-Time Contingent Claims.” *Mathematical Finance*, 10 (2000), pp. 53–75.
- [16] KUNITOMO, N., AND IKEDA, M. “Pricing Options with Curved Boundaries.” *Mathematical Finance*, 2 (1992), pp. 275–298.
- [17] LUO, L. “Various Types of Double Barrier Options.” *Journal of Computational Finance*, 4 (2001), pp. 125–138.
- [18] LYUU, Y.-D. “Very Fast Algorithms for Barrier Option Pricing and the Ballot Problem.” *Journal of Derivatives*, 5 (1998), pp. 68–79.
- [19] LYUU, Y.-D. *Financial Engineering & Computation: Principles, Mathematics, Algorithms*. Cambridge: Cambridge University Press, 2002.
- [20] RITCHKEN, P. “On Pricing Barrier Options.” *Journal of Derivatives*, 3 (1995), pp. 19–28.
- [21] SIDENIUS, J. “Double Barrier Options: Valuation by Path Counting.” *Journal of Computational Finance*, 1 (1998), pp. 63–79.
- [22] TIAN, Y. “A Flexible Binomial Option Pricing Model.” *Journal of Futures Markets*, 19 (1999), pp. 817–843.

- [23] WIDDICKS, M., ANDRICOPOULOS, A.D., NEWTON, D.P., AND DUCK, P.W. “On the Enhanced Convergence of Lattice Methods for Option Pricing.” *Journal of Futures Markets*, 22 (2002), pp. 315–338.
- [24] ZHANG, G. *Exotic Options: A Guide to Second Generation Options*. Singapore: World Scientific, 1998.

A Validity of Risk-Neutral Probabilities

To ensure that the branching probabilities in Eqs. (6)–(8) are valid, it suffices to show that $P_u, P_m, P_d \geq 0$. For convenience, define

$$\begin{aligned}\det &= (\beta - \alpha)(\gamma - \beta)(\gamma - \alpha), \\ \det(u) &= (\beta\gamma + \text{Var}(\Delta t'))(\gamma - \beta), \\ \det(m) &= (\alpha\gamma + \text{Var}(\Delta t'))(\alpha - \gamma), \\ \det(d) &= (\alpha\beta + \text{Var}(\Delta t'))(\beta - \alpha).\end{aligned}$$

Then the branching probabilities can be represented as $P_u = \det(u)/\det$, $P_m = \det(m)/\det$, and $P_d = \det(d)/\det$ by Cramer's rule. Note that $\det < 0$ because $\alpha > \beta > \gamma$. To prove that the branching probabilities are nonnegative, it is sufficient to show $\det(u)$, $\det(m)$, and $\det(d) \leq 0$ instead. In addition, since $\alpha > \beta > \gamma$, it suffices to show that $\beta\gamma + \text{Var}(\Delta t') \geq 0$, $\alpha\gamma + \text{Var}(\Delta t') \leq 0$, and $\alpha\beta + \text{Var}(\Delta t') \geq 0$ under the premises $\Delta t \leq \Delta t' < 2\Delta t$ and $\beta \in [-\sigma\sqrt{\Delta t}, \sigma\sqrt{\Delta t}]$. These three inequalities are proved as follows:

$$\begin{aligned}\beta\gamma + \text{Var}(\Delta t') &= \beta^2 - 2\beta\sigma\sqrt{\Delta t} + \sigma^2\Delta t' \geq \beta^2 - 2\beta\sigma\sqrt{\Delta t} + \sigma^2\Delta t = (\beta - \sigma\sqrt{\Delta t})^2 \geq 0, \\ \alpha\gamma + \text{Var}(\Delta t') &= \beta^2 - 4\sigma^2\Delta t + \sigma^2\Delta t' \leq \beta^2 - 4\sigma^2\Delta t + 2\sigma^2\Delta t = \beta^2 - 2\sigma^2\Delta t \leq 0, \\ \alpha\beta + \text{Var}(\Delta t') &= \beta^2 + 2\beta\sigma\sqrt{\Delta t} + \sigma^2\Delta t' \geq \beta^2 + 2\beta\sigma\sqrt{\Delta t} + \sigma^2\Delta t = (\beta + \sigma\sqrt{\Delta t})^2 \geq 0.\end{aligned}$$

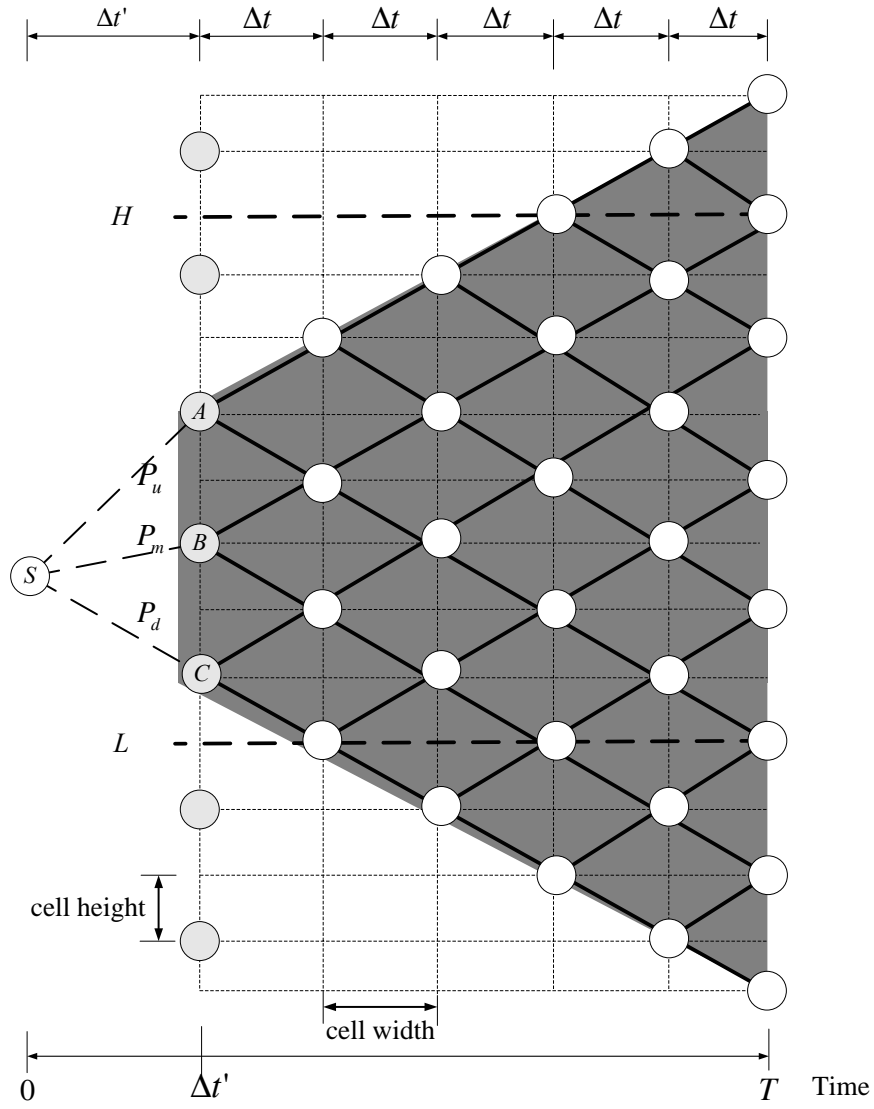


Exhibit 1: **The bBTT for Pricing Continuous Double-Barrier Options.** Two barriers, L and H , are in thick dashed lines. The root of the bBTT is the node S . The CRR tree (with the first two time steps truncated) that comprises the bulk of the bBTT is shaded. This CRR tree is placed on a grid (in thin dotted lines) that has two layers of nodes coinciding with the two barriers.

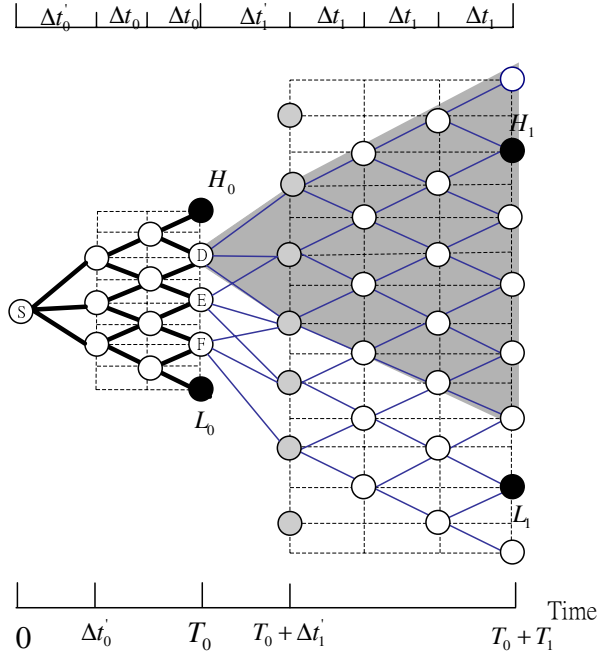


Exhibit 2: **Pricing Discrete Moving-Double-Barrier Knock-Out Options by the BTT.** The barriers are H_0 and L_0 at time T_0 and H_1 and L_1 at $T_0 + T_1$. Four critical points (the black nodes) are on some nodes of the BTT. The BTT is constructed by combining four bBTTs emanating from nodes S , D , E , and F . The bBTT emanating from node S is in thick edges, whereas the bBTT emanating from node D is in shadow.

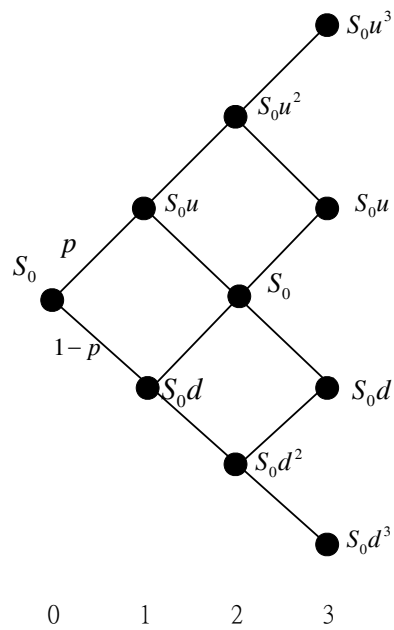


Exhibit 3: **The CRR Tree.** The initial stock price is S_0 . The upward and downward multiplicative factors for the stock price are u and d , respectively. The upward and downward branching probabilities are p and $1 - p$, respectively.

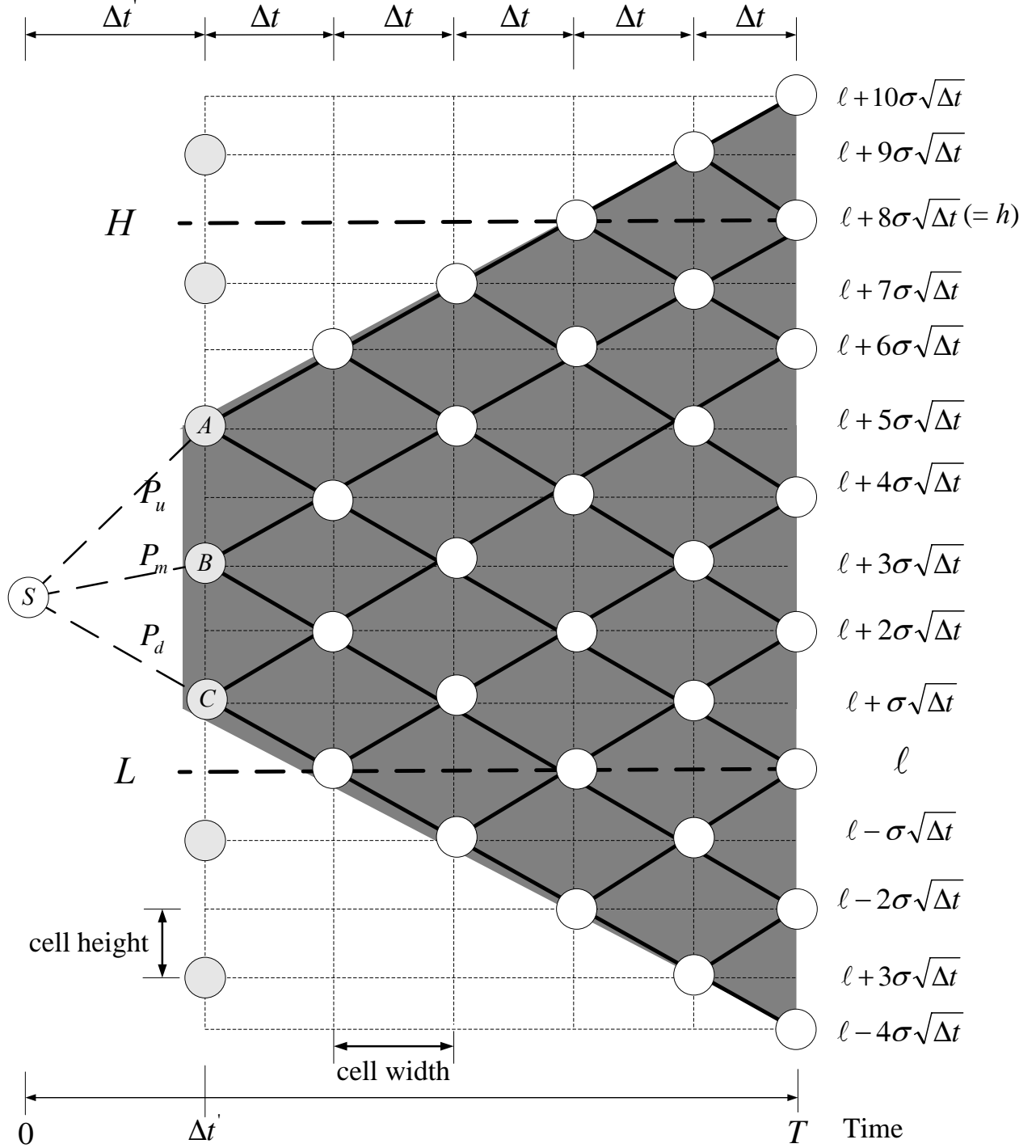


Exhibit 4: **The bBTT for Pricing a Continuous Double-Barrier Option.** Two barriers, L and H , are denoted by thick dashed lines. The root of the bBTT is denoted by node S . The CRR tree (with the first two time steps truncated) that comprises the bulk of the bBTT is shadowed. This CRR tree is put on a grid (thin dotted lines). The S_S -log-prices of the nodes at maturity are next to these nodes.

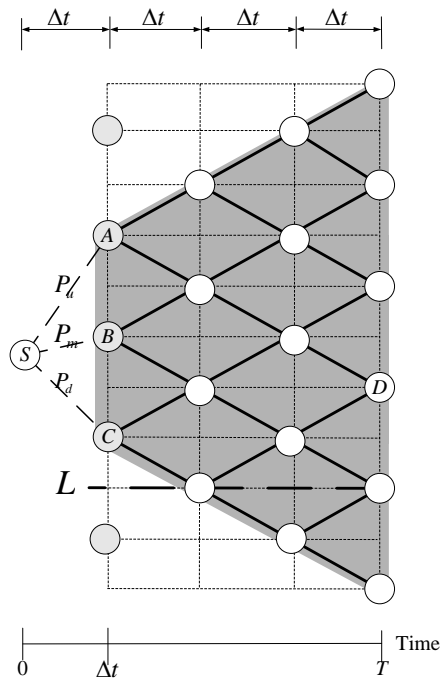


Exhibit 5: **The bBTT for Pricing a Single-Barrier Option.** Barrier L is denoted by thick dashed lines. The root of the bBTT is denoted by node S . The CRR tree (with the first two time steps truncated) that comprises the bulk of the bBTT is shaded. This CRR tree is placed on a grid.

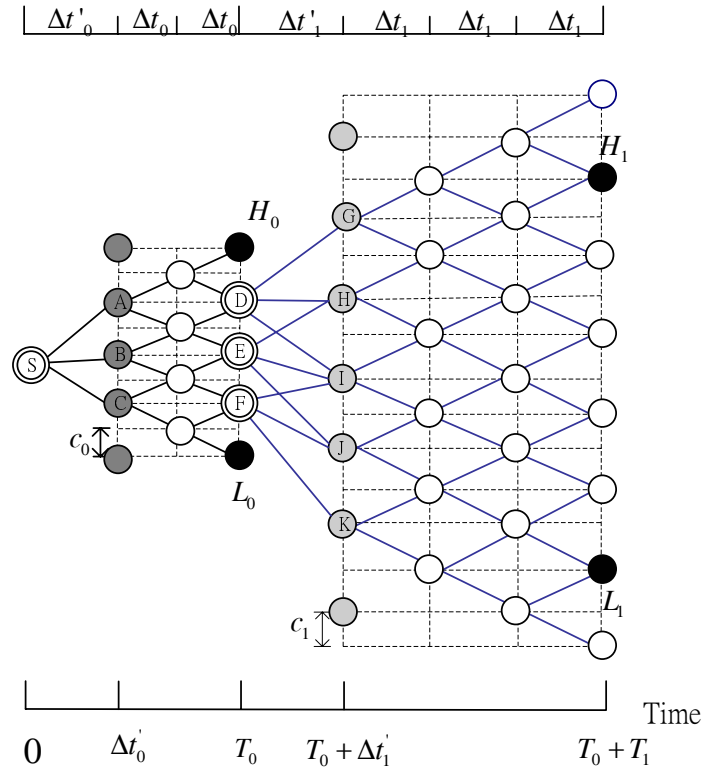


Exhibit 6: **Pricing a Moving-Double-Barrier Knock-Out Option by the BTT.** The discrete barriers (the black nodes) are H_0 and L_0 at time T_0 and H_1 and L_1 at time $T_0 + T_1$. The BTT is composed of four bBTTs, and the root nodes of these bBTTs are marked in double circles. Two truncated CRR trees, one emanating from nodes A , B , and C at time $\Delta t'_0$ and the other one emanating from nodes G , H , I , J , and K at time $T_0 + \Delta t'_1$, comprise the bulk of the BTT. The numbers c_0 and c_1 denote the height of a cell of the former grid and the latter grid, respectively.

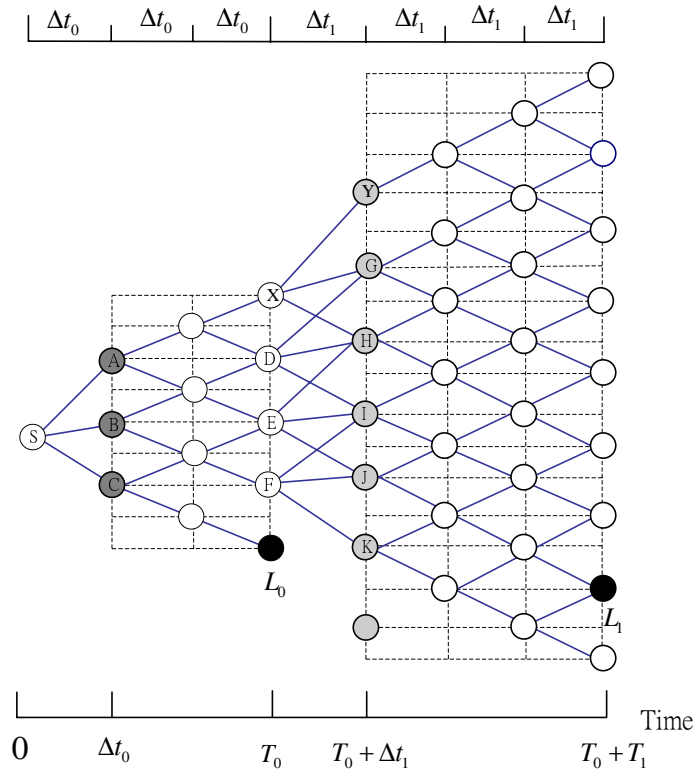


Exhibit 7: **Pricing a Moving-Single-Barrier Knock-Out Option by the BTT.** The barriers (marked by black nodes) are L_0 at time T_0 and L_1 at $T_0 + T_1$. Two truncated CRR trees, one growing from nodes A , B , and C at time Δt_0 and the other one growing from Y , G , H , I , J , and K at time $T_0 + \Delta t_1$, comprise the bulk of the BTT.

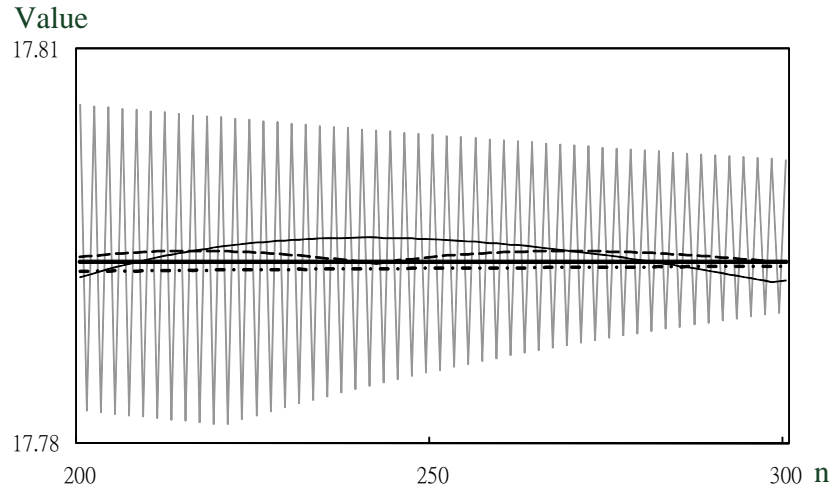


Exhibit 8: **Convergence of the CRR, the AMM, and the BTT in Valuing Vanilla Call Options.** The x -axis denotes the number of time steps. The y -axis denotes the option value. The initial stock price is 100, the exercise price is 98, the risk-free rate is 10% per annum, the volatility of the stock price is 30%, and the time to maturity is 1 year. The line that oscillates most significantly is the CRR tree model. The thin solid line denotes the AMM with AMM level 1. The dashed line denotes the AMM with AMM level 2. The dot-dashed line denotes the BTT. The thick solid line denotes the analytical option value 17.7943.

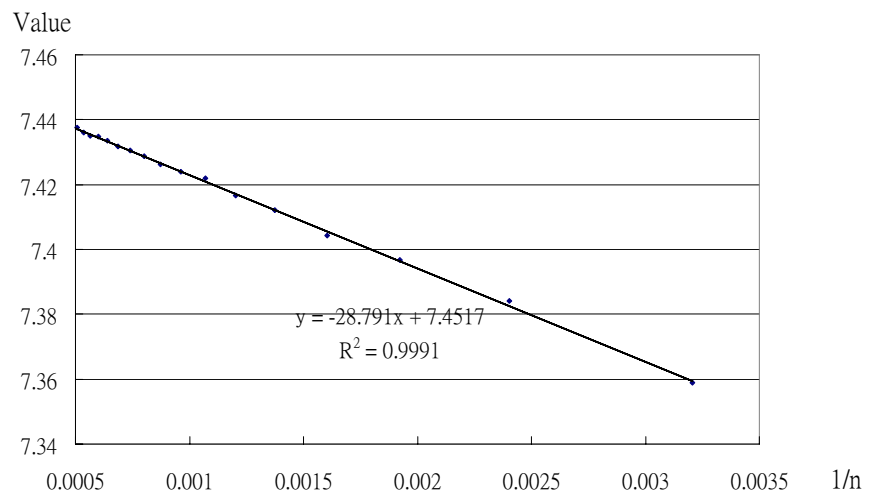


Exhibit 9: **Convergence of the BTT in Valuing a Discrete Down-and-Out Single-Barrier Call Option.** The initial stock price is 95, the volatility of stock price is 25%, the risk-free rate is 10%, the strike price is 100, the time to maturity is 1 year, the barrier is 90, and the stock price is monitored 52 times during the option life. The x -axis denotes $1/n$, where n denotes the number of time steps of the BTT. The y -axis denotes the option value.

QUAD			AMM			BTT		
K	Time	Error	m	Time	Error	n	Time	Error
4	.092	.001310	50	.093	.005324	936	.082	.000446
5	.124	.001295	60	.141	.006919	1144	.132	.000690
6	.171	.001292	70	.172	.000424	1248	.187	.000301
7	.217	.001291	80	.234	.002178	1456	.233	.000360
8	.280	.001291	90	.297	.003500	1664	.283	.000209
MAE	.001310		.006919			.000689		
RMSE	.001296		.004322			.000433		

Exhibit 10: **AMM, the BTT and QUAD for Pricing a Discrete Single-Barrier Call Option.** The numerical settings are the same as the settings in Exhibit 9. The benchmark value, 7.451115, is computed by Monte Carlo simulation with 10,000,000 trails. The parameter K tunes the fineness of QUAD by determining the distance between two nodes at the same monitoring date (see Andricopoulos *et al.* (2003)). The parameter m denotes the number of time steps between two monitoring dates in the AMM. Note that the AMM level equals 1 in this case (see Gao *et al.* (1999)). The parameter n denotes the number of time steps of the BTT. Time and Error denote the computational time and the absolute pricing error, respectively. We apply the Richardson-type extrapolation to both QUAD and the BTT method. MAE and RMSE denote the maximum absolute error and root-mean-squared error, respectively.

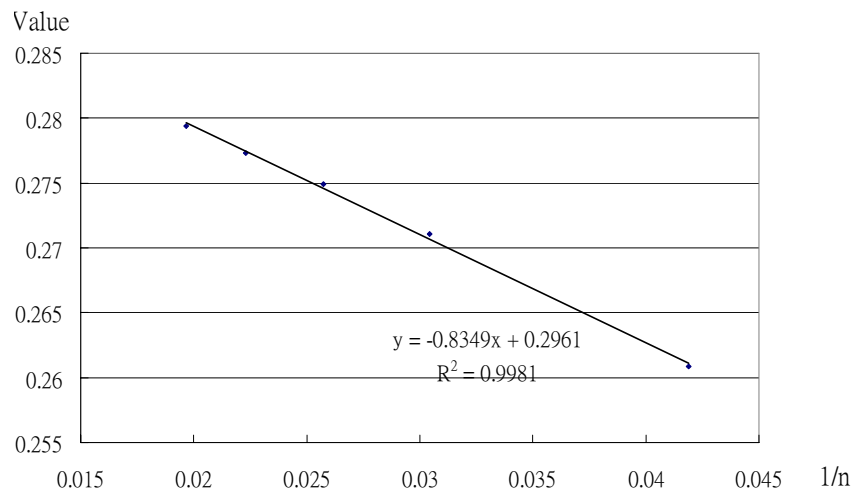


Exhibit 11: **Pricing a Discrete, Moving-Double-Barrier Call Option.** The initial stock price is 100, the volatility of stock price is 30%, the risk-free rate is 10%, the strike price is 100, and the time to maturity is 1 year. The stock price is monitored at years 1/3, 2/3, and 1. The low barriers are 90, 90.05, 90.1, respectively. The high barriers are 110, 110.05, 110.1, respectively. The benchmark value as computed by the Monte Carlo simulation with 1,000,000 trials is 0.29616675.

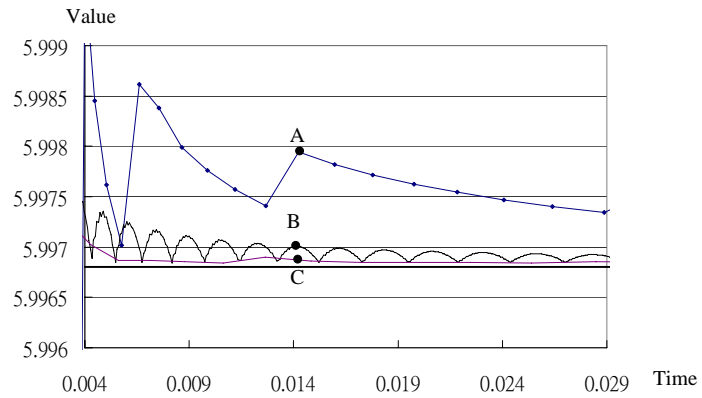


Exhibit 12: **Convergence of the BTT and Ritchken's Trinomial Tree Model in Pricing a Continuous Down-and-Out Single-Barrier Call Option.** The initial stock price is 95, the exercise price is 100, the risk-free rate is 10% per annum, the volatility of the stock price is 25%, the time to maturity is 1 year, and the barrier is 90. The horizontal line $y = 5.9968$ denotes the true value, the lines passing through point A, B and C denote the prices generated by the Ritchken's model, the BTT (matching the barrier only), and the BTT (matching both the barrier and the strike price), respectively.

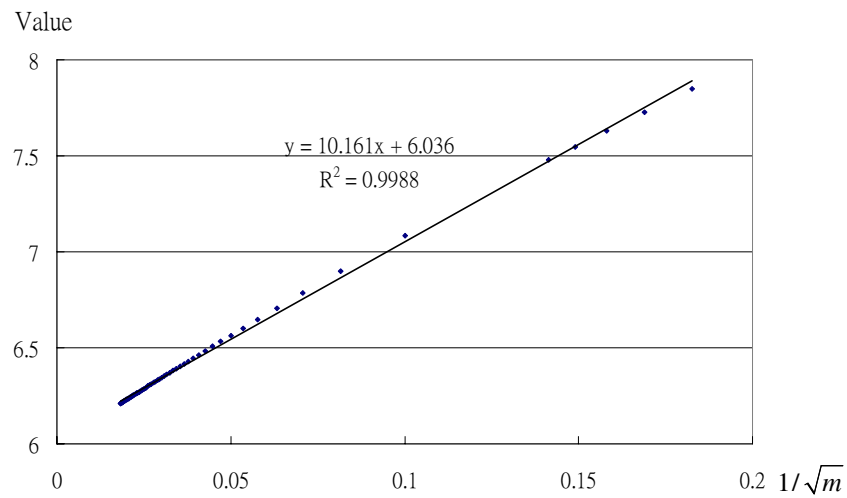


Exhibit 13: **Convergence of QUAD for Pricing a Continuous Down-and-Out Single-Barrier Call Option by Extrapolation.** The numerical settings are the same as the settings in Exhibit 12. The x -axis denotes $1/\sqrt{m}$, where m denotes the number of monitoring dates of an otherwise identical discrete barrier call option. The y -axis denotes the option value of the discrete barrier call computed by QUAD. The value of the continuous barrier call option estimated by regression is about 6.036.

S	T	Gao	Ritchken	BTT	QUAD
40.0	0.5	5.1869	5.1828	5.1881	5.1902
40.0	1.0	5.3843	5.3797	5.3859	5.3907
42.5	0.5	3.3456	3.3443	3.3475	3.3508
42.5	1.0	3.6423	3.6403	3.6443	3.6435
45.0	0.5	1.9357	1.9331	1.9374	1.9347
45.0	1.0	2.2132	2.2132	2.2148	2.2213
47.5	0.5	0.8614	0.8609	0.8624	0.8622
47.5	1.0	1.0213	1.0215	1.0221	1.0310
RMSE			0.0026	0.0016	0.0055
MAE			0.0046	0.0020	0.0097

Exhibit 14: **Pricing American Up-and-Out Barrier Puts.** The initial stock prices are listed in the first column, the times to maturity are in the second column, the volatility is 0.2, the strike price is 45, the barrier level is 50, and the risk-free rate is 4.88%. Gao denotes the quasi-analytical formula in Gao *et al.* (1996). Ritchken denotes the Ritchken's trinomial tree based on 5,000 time steps. We apply the Richardson-type extrapolation to both QUAD and the BTT model. MAE and RMSE denote the maximum absolute error and root-mean-squared error, respectively.

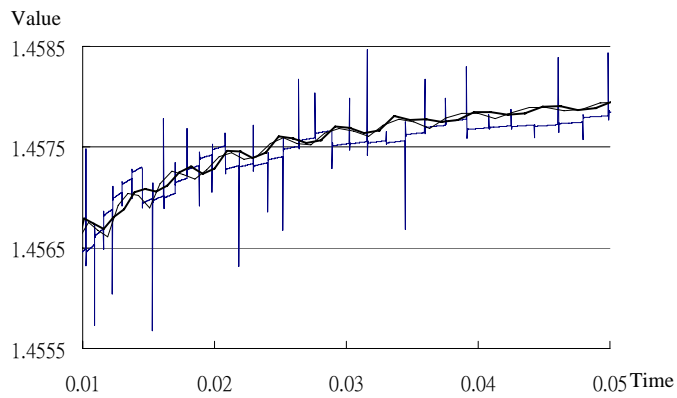


Exhibit 15: **Pricing a Continuous Double-Barrier Knock-Out Call.** The initial stock price is 95, the exercise price is 100, the risk-free rate is 10% per annum, the volatility of the stock price is 25%, the time to maturity is 1 year, and the two barriers are 140 and 90, respectively. The x -axis and the y -axis denote the computational time and the option value, respectively. The thin curve (with large oscillations), the thin curve (with small oscillations), and the thick curve denote Ritchken's trinomial tree model, the BTT (matching the two barriers only), and the BTT (matching the two barriers and the strike price), respectively.

Stock Price	Ritchken			AMM		BTT			True
	n	Time	Value	Time	Value	n	Time	Value	Value
91	1000	0.110	1.274	0.032	1.274	2000	0.005	1.274	1.274
90.5	4000	1.891	0.642	0.235	0.642	8000	0.023	0.642	0.642
90.4	5000	3.062	0.515	0.562	0.515	11000	0.031	0.515	0.515

Exhibit 16: **Ritchken’s Model, the BTT, and the AMM When the Barrier Is Close to the Initial Stock Price.** A down-and-out single-barrier call option is priced above. The initial stock prices are listed in the first column, the barrier is 90, the exercise price is 100, the risk-free rate is 10% per annum, the volatility of the stock price is 25%, and the time to maturity is 1 year. **Ritchken** denotes Ritchken’s trinomial tree model. **AMM** denotes the adaptive mesh model. **BTT** denotes the BTT. The variable n , Time, and Value denote the number of time steps, the computational time, and the price of each tree model, respectively. The true values with 3-digit accuracy are from Ritchken (1995).