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

Abstract

Most derivatives do not have simple valuation formulas and must be priced by numerical methods. However, the distribution error and the nonlinearity error introduced by many numerical methods make the pricing results converge slowly or even oscillate significantly. This paper introduces a novel tree model, the bino-trinomial tree (BTT) model, for pricing a wide range of derivatives. The BTT reduces the nonlinearity error sharply by adjusting its structure to suit the derivative's specification; consequently, the pricing results converge smoothly and quickly. As an added benefit, pricing of some European-style options on the BTT can be made extremely efficient by combinatorial tools, which is not possible with most traditional tree models. Therefore, the BTT can efficiently reduce the distribution error by picking a large number of time steps. Besides, our approach can be easily adapted for nonstandard payoff functions unlike analytical formulas because the tree construction is independent of the payoff function. This paper uses vanilla options, single-barrier options, and double-barrier options to demonstrate the effectiveness of the BTT. Extensive numerical experiments are given to show the superiority of the BTT to other popular models, including the adaptive mesh model. For example, the BTT outperforms other methods in solving the notorious "barrier-too-close" problem.

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

1 Introduction

A financial derivative is a financial instrument whose payoff is based on other more elementary financial "assets," such as stocks, indexes, currencies, commodities, bonds, mortgages, other derivatives, temperatures, and countless others. The elementary financial instrument in this paper is assumed to be stock for convenience. With the rapid growth of financial markets, more sophisticated derivatives are constantly being structured by financial institutions to satisfy the needs of their clients.

Although financial innovations make the markets more efficient, they also give rise to problems in pricing because only a small subset of financial derivatives can be priced by exact yet simple analytical formulas. Although approximation formulas may exist for other derivatives, these formulas may potentially lead to large pricing errors. Those derivatives that can not be valued accurately by analytical formulas must be priced by numerical methods such as tree models. Take the barrier option for example. A barrier option is an option whose payoff depends on whether the stock's price path ever touches certain price levels called the barriers. Exact yet simple analytical formulas for single-barrier options exist only when their payoff functions take certain forms (see Merton (1973) and Reiner and Rubinstein (1991)). For double-barrier options, no simple, exact closed-form pricing formula is available. The value of the double-barrier option can be expressed as an infinite series of cumulative normal distributions. Analytical approximation formulas for the infinite series have been studied by Kunitomo and Ikeda (1992), German and Yor (1996), Bhagavatula and Carr (1997), Sidenius (1998), and Luo (2001). Truncation of this infinite series is necessary in practice, but it can lead to large pricing errors (see Luo (2001)). Besides, these analytical formulas can not be easily adapted for nonstandard payoff functions.

A tree model is a popular numerical method for pricing options. This approach is flexible in that an option can be priced with only nominal changes even when its payoff function is nonstandard, such as power options, for which there may not be closed-form formulas. A tree divides the time span from now to the option's maturity date into n time steps and specifies the stock price discretely at each time step. The well-known CRR tree model of Cox, Ross and Rubinstein (1979) is a good example. A simple 3-time-step CRR tree is illustrated in Fig. 1. The stock price of the root node of the CRR tree is S_0. 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 $ud=1$. The option values priced by tree models converge to the theoretical option value under the continuous-time model as $n\to\infty$ (see Duffie (1996)). Unfortunately, the convergence may not be smooth or fast. Worse, for some options like barrier options, the convergence can oscillate significantly (see Boyle and Lau (1994)). As a result, large amounts of computational time may be required to achieve acceptable accuracy.

Figlewski and Gao (1999) identify two types of errors, the distribution error and the nonlinearity error, when pricing with discrete-time, discrete-state tree models. The distribution error refers to the error introduced by approximating the continuous lognormal distribution of the stock price with a discrete probability distribution. The nonlinearity error is introduced by the nonlinearity of the option value function. Look at the CRR tree in Fig. 1 again. This 3-time step tree approximates the stock price at the maturity date with the following discrete random variable $\boldsymbol{S}$:

$$\boldsymbol{S}=\begin{cases} S_0u^id^{3-i}, & \text{with probability } \binom{3}{i}p^i(1-p)^{3-i}, \quad i=0,1,2,3,\\ \text{otherwise}, & \text{with probability } 0. \end{cases}$$

By construction, the random variable $\boldsymbol{S}$ and the stock price at the maturity date have the same mean and variance, but the discrepancy between these two distributions produces distribution error in the option value. The distribution error converges to zero as $n\to\infty$, however.

Recall that the nonlinearity error is introduced by the nonlinearity of the option value function. Consider the pricing of a single-barrier option with barrier L. The payoff of a barrier option depends on whether the stock price touches the barrier before the maturity date. Thus the option value function is highly nonlinear along the barrier. We demonstrate why price oscillations occur with the CRR trees with Fig. 2. In the 2-time-step tree of panel (a), the stock price does not hit barrier L exactly. Instead, a close price on the tree like L_1 acts as the barrier. We call L_1 the effective barrier. Similarly, in the 3-time-step tree of panel (b), the effective barrier is changed to, say, L_2. Price oscillations occur because the effective barrier fluctuates with the number of time steps, n.

The oscillation phenomenon for pricing vanilla options by the tree model has been studied by Omberg (1987). To suppress price oscillations, the tree model can be structured differently to reduce the nonlinearity error. Klassen (2001) reduces the nonlinearity error by tuning the structure of the tree so that the middle node of his tree hits the exercise price at the maturity date. Figlewski and Gao (1999) propose the adaptive mesh model (AMM) to solve this problem. This tree model can be roughly viewed as a combination of two types of trinomial trees: the base tree and the finer one(s). The resolution of the base tree is low for efficiency concerns. The finer trees have higher resolution and are built on only part of the base tree where the nonlinearity error can be significantly reduced. Take Fig. 3 for example. The resolution of the AMM is high near the exercise price at the maturity date. This is because the payoff function of a vanilla option is highly nonlinear near the exercise price. Although the AMM approach is efficient and fast convergent, it has a complicated structure, which makes it hard to implement and still harder to tailor to different derivatives.

Barrier option pricing on a CRR tree oscillates significantly because of the nonlinearity error. To handle this problem, Boyle and Lau (1994) propose using only those n for which the CRR tree has a layer that is very close to the barrier. This method reduces the oscillations dramatically, but raising n according to the regime sanctioned by Boyle and Lau (1994) may not necessarily produce more accurate results according to Ritchken (1995). Their method can not be easily adapted to handle multiple barriers, either. This is because it is impossible to pick an n that will tailor to more than one barrier. Derman et al. (1995) calculate the value for each hallow node (in panel (b) of Fig. 2) by adopting L_2 (equivalent to moving the barrier L outward) and L' (equivalent to moving the barrier L inward) as effective barriers, respectively. The value for each hollow node is then obtained by interpolating the two values mentioned above. Hull (2000) calls L' and L_2 the inner barrier and the outer barrier.

Ritchken (1995) proposes a trinomial tree model to alleviate the oscillation problem. With a stretch parameter to tune the structure of the tree, one layer of the tree can be made to coincide with the barrier. When pricing double-barrier options, two separate stretch parameters are required to make two different layers of the tree coincide with the two barriers. The branching probabilities of Ritchken's trinomial tree model can be made valid (i.e., between 0 and 1) by raising n if necessary. The value of n can be large when interest rates are high, volatilities are low, or, most seriously, a barrier is very close to the initial stock price (resulting in the so-called "barrier-too-close" problem). Ritchken's algorithm is therefore potentially very costly. Figlewski and Gao (1999) use the AMM to solve this "barrier-too-close" problem. But no efforts have been made to extend the AMM to price double-barrier options.

To reduce the nonlinearity error, this paper proposes a simple tree model, the bino-trinomial tree (BTT), that combines a CRR binomial tree structure and a root node with a trinomial structure. The construction steps are (1) finding a suitable length for the time step (Δt) of the CRR tree, (2) properly laying out the CRR tree, and (3) locating the position of the root node and calculating its branching probabilities. Let us use the BTT in Fig. 4 to illustrate the ideas. This BTT prices a double-barrier option with two barriers, L and H, and with time to maturity T. It is composed of a CRR tree (in shadow) but with the first two time steps removed. To position this truncated CRR tree properly, we need to fix the underlying grid first. Note that the width of each cell of the grid equals the length of a time step of the CRR tree. The ratio of the height and the width of each cell of the grid is a fixed constant as determined by the CRR structure (it will be detailed later). To reduce the nonlinearity error, the grid must have two layers that coincide with the barriers L and H. To realize it, we adjust the width of the cells so the distance between the two barriers is an integer multiple of the height of a cell. Then we lay out the grid from the barrier L upward so that barrier H is also on a layer of the grid. The length of the first time step of the BTT, $\Delta t'$, is the remaining amount of time to make the whole time span T. Nodes A, B, and C are selected from the light gray nodes of the grid at time $\Delta t'$ to make the branching probabilities from the root node S (i.e.,

P_u, P_m, and P_d) valid. The truncated CRR tree is finally laid on top of the grid by growing from nodes A, B, and C.

That a CRR tree constitutes the bulk of the BTT has important consequences. Pricing of options like barrier options, lookback options, Parisian options, and vanilla options on the CRR tree can all be sped up by combinatorial techniques (see Lyuu (1998) and Dai et al. (2006)). These very combinatorial algorithms therefore speed up pricing on the BTT as well. The running times of all BTT pricing algorithms described in this paper turn out to be proportional to n, the number of time steps of the tree. Thus these algorithms are called $O(n)$-time algorithms. On the other hand, the running times of other tree approaches are at least proportional to $n^2/2$. (These tree approaches are called $\Omega(n^2)$-time algorithms.) This is because they must evaluate the option value on each node of a tree during backward induction and the number of nodes of the tree alone is at least $n^2/2$. Thus, it costs much less time to use the BTT than other tree models with the same n, which means the BTT can afford to reduce the distribution error by picking large n. Finally, the trinomial structure at the root node allows us to calculate both delta and gamma on the BTT essentially without extra work (see Pelsser and Vorst (1994)).

In this paper, we will use vanilla options, single- and double-barrier options as examples to show how to sharply reduce the nonlinearity error with the BTT. Specifically for vanilla options, we make a layer of the BTT coincide with the exercise price. For both single- and double-barrier options, we make each barrier coincide with a layer of the BTT. Numerical results show that our approach converges more smoothly and faster than other tree approaches. Most interestingly, the BTT outperforms others in solving the notorious barrier-too-close problem.

Our paper is organized as follows. The assumptions of the stock price process and the definitions of vanilla options and barrier options are given in section 2. The methodology to construct the BTT is given in section 3. We will construct BTTs for double-barrier options, single-barrier options, and vanilla options. Section 4 details an efficient $O(n)$-time algorithm that prices double-barrier options on the CRR tree. This algorithm is then used to build an $O(n)$-time algorithm on BTT. 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:

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

where W_t is the standard Wiener process, r is the risk-free interest rate per annum, and σ denotes 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 stock for the exercise price and does not have other unusual features. The payoffs of a vanilla call option and a vanilla put option at time T are $\max(S_T - X, 0)$ and $\max(X - S_T, 0)$, respectively.

A barrier option is an option whose payoff depends on whether the 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. On the other hand, a knock-out one ceases to exist if the stock price touches the barrier(s) before the maturity. A single-barrier option is an option with only one barrier, whereas a double-barrier option is an option with two barriers.

Consider a single-barrier option with barrier L. Assume that $L < S_0$ for convenience. Define $S_{\text{inf}} = \inf_{0 \leq t \leq T} S_t$. The payoff of a knock-out single-barrier call option is

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

Consider a double-barrier option with two barriers L and H. Assume $L < S_0 < H$ for convenience. Define $S_{\text{sup}} = \sup_{0 \leq t \leq T} S_t$. The payoff of a knock-in double-barrier call option is

$$\text{Payoff} = \begin{cases} \max(S_T - X, 0), & \text{if } S_{\sup} \geq H \text{ or } S_{\inf} \leq L, \\ 0, & \text{otherwise.} \end{cases}$$

Finally, the payoff of a knock-out double-barrier call option is

$$\text{Payoff} = \begin{cases} 0, & \text{if } S_{\sup} \geq H \text{ or } S_{\inf} \leq L, \\ \max(S_T - X, 0), & \text{otherwise.} \end{cases}$$

A tree model divides a certain time interval from time 0 to time T into n time steps and specifies the stock price discretely at each time step. A tree converges to the stock price process mentioned in Eq. (1) if 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 illustrated in Fig. 1. To match the first two moments of the stock price process, the CRR tree sets $u \equiv e^{\sigma\sqrt{T/n}}$ and $d \equiv e^{-\sigma\sqrt{T/n}}$. The probability p is set to $(e^{rT/n} - d)/(u - d)$. Note that the stock price S resulting from j down moves and $i - j$ up moves from time step 0 equals $S_0 u^{i-j} d^j$ with probability $\binom{i}{j} p^{i-j}(1-p)^j$. This node is at time step i and is denoted as $N(i, j)$ for simplicity.

3 Construction of the BTT

This section shows how to construct BTTs for pricing double-barrier options, single-barrier options, and vanilla options, in that order as each is a special case of the predecessor.

Double-Barrier Options

We will be constantly referring to the BTT in Fig. 5. This BTT prices a double-barrier option with two barriers L and H. The constituent CRR tree is in gray. This CRR tree is laid on a grid. The first two time steps of the CRR tree are truncated, and this truncated CRR tree grows from three nodes: A, B, and C at time $\Delta t'$. These three nodes are connected to node S with stock price S_0 at time 0 with branching probabilities P_u, P_m, and P_d.

Define the log-price of stock price V' as $\ln(V'/S_0)$. A log-price of z therefore implies a stock price of $S_0 e^z$. The upward and the downward additive factors of the log-prices on the CRR tree are $\sigma\sqrt{\Delta t}$ and $-\sigma\sqrt{\Delta t}$, respectively. The intersection of a vertical line and a horizontal line of the grid defines a vertex. Thus each node of the CRR tree is on a vertex. The width of a cell (of the grid) equals Δt, the length of a time step in the CRR tree. The height of a cell is $\sigma\sqrt{\Delta t}$. Note that the difference between the log-prices of two adjacent nodes like nodes A and B is $2\sigma\sqrt{\Delta t}$. The log-prices of the two barriers H and L are $h \equiv \ln(H/S_0)$ and $\ell \equiv \ln(L/S_0)$, respectively.

To price a double-barrier option accurately and efficiently, the BTT should have the following two features: (1) Two layers of BTT 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., $0 \leq P_u, P_m, P_d \leq 1$). We proceed to show how to achieve both goals.

First, we show how to choose the width of a cell to make the grid hit both H and L. If the grid has two layers that coincide with H and L, $\frac{h-\ell}{2\sigma\sqrt{\Delta t}}$ should be some integer k. For example, $k = 4$ in Fig. 5. Suppose that we want to construct an (approximately) m-time-step BTT. Ideally, the length of each time step is $\Delta\tau \equiv T/m$, but in really $\frac{h-\ell}{2\sigma\sqrt{\Delta\tau}}$ 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. It is hence natural to pick $\Delta t = \left(\frac{h-\ell}{2\kappa\sigma}\right)^2$, where $\kappa = \left\lceil \frac{h-\ell}{2\sigma\sqrt{\Delta\tau}} \right\rceil$. Lay out the grid from barrier L upward. Automatically, a layer coincides with barrier H. Note that the number of time steps of the BTT is $\lfloor \frac{T}{\Delta t} \rfloor$ (instead of m) as the truncated CRR tree has $\lfloor \frac{T}{\Delta t} \rfloor - 1$ time steps. The length of the first time step of the BTT, $\Delta t'$, is the remaining amount of time to make the whole tree span T years:

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

It is easy to verify that $\Delta t \leq \Delta t' < 2\Delta t$.

We still need to select nodes A, B, and C among the light gray vertices to make the branching probabilities from node S valid. These three nodes are then 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 enough degrees of freedom. The mean and the variance of the log-prices at A, B, and C equal

$$\begin{aligned} \mu &\equiv (r - \sigma^2/2)\,\Delta t', \\ \mathrm{Var} &\equiv \sigma^2 \Delta t', \end{aligned}$$

respectively. The log-price for each light gray vertex at time $\Delta t'$ is:

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

for some integer j. As the difference between the log-prices of two adjacent light gray vertices is $2\sigma\sqrt{\Delta t}$, there must exist a unique vertex whose log-price lies in the interval $[\,\mu - \sigma\sqrt{\Delta t}, \mu + \sigma\sqrt{\Delta t})$. We select this vertex for node B. For example, the log-price at node B is $\ell + 3\sigma\sqrt{\Delta t}$ in Fig. 5. Denote the log-price at B as $\hat{\mu}$, which is closest to μ among the log-prices at time $\Delta t'$. The log-prices at nodes A and C are therefore $\hat{\mu} + 2\sigma\sqrt{\Delta t}$ and $\hat{\mu} - 2\sigma\sqrt{\Delta t}$, respectively. Define

$$\begin{aligned} \beta &\equiv \hat{\mu} - \mu, \\ \alpha &\equiv \hat{\mu} + 2\sigma\sqrt{\Delta t} - \mu = \beta + 2\sigma\sqrt{\Delta t}\,, \\ \gamma &\equiv \hat{\mu} - 2\sigma\sqrt{\Delta t} - \mu = \beta - 2\sigma\sqrt{\Delta t}\,. \end{aligned}$$

Note that the first equation implies that $\beta \in [-\sigma\sqrt{\Delta t}, \sigma\sqrt{\Delta t})$. Note also that $\alpha > \beta > \gamma$. The branching probabilities can be derived by solving the following three equalities

$$P_u\alpha + P_m\beta + P_d\gamma = 0, \tag{3}$$

$$P_u\alpha^2 + P_m\beta^2 + P_d\gamma^2 = \mathrm{Var}, \tag{4}$$

$$P_u + P_m + P_d = 1. \tag{5}$$

Equations (3) and (4) match the first two moments of the logarithmic stock price, and Eq. (5) ensures that P_u, P_m, P_d as probabilities sum to one. Appendix A verifies that the above three equations yield valid branching probabilities.

Pricing European-style double-barrier options on the BTT can be sped up by a combinatorial algorithm given by Lyuu (2002) and Dai and Lyuu (2005) in section 4. The combinatorial algorithm is used to evaluate the option values of nodes A, B, and C in $O(n)$ time. The final pricing result of our BTT at node S is evaluated by standard backward induction:

$$V_S = e^{-r\Delta t'} \left(P_u V_A + P_m V_B + P_d V_C \right), \tag{6}$$

where V_X denotes the double-barrier option value at node X.

Single-Barrier Options

The BTT for pricing a single-barrier option with barrier L follows similar lines and is illustrated in Fig. 6. Again, barrier L must be hit by the BTT to realize reduction in the nonlinearity error. This can be achieved by laying the underlying grid from barrier L upward as before. Note that we no longer adjust the width of a cell of the grid as we did in the case of double-barrier options. This is because only one barrier needs to be hit now instead of two. Thus the length of a cell (Δt) is simply set to be T/n for an n-time step BTT. Note also that the length of the first time step of the BTT is now simply Δt. Finally, select adjacent nodes A, B, and C among the light gray vertices at time Δt to make the branching probabilities P_u, P_m and P_d valid. These three nodes are required 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 log-prices of A, B, and C equal

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

respectively. The log-price of a light gray vertex at time Δt is expressed in Eq. (2). Again, there exists a unique light gray vertex whose log-price lies in the interval $[\mu - \sigma\sqrt{\Delta t}, \mu + \sigma\sqrt{\Delta t})$, and this vertex is selected as node B. Denote the log-price of B as $\hat{\mu}$. The log-prices of the two nodes A and C are selected as $\hat{\mu} + 2\sigma\sqrt{\Delta t}$ and

$\hat{\mu} - 2\sigma\sqrt{\Delta t}$, respectively. Define

$$\begin{aligned}
\beta &\equiv \hat{\mu} - \mu, \\
\alpha &\equiv \hat{\mu} + 2\sigma\sqrt{\Delta t} - \mu = \beta + 2\sigma\sqrt{\Delta t}\,, \\
\gamma &\equiv \hat{\mu} - 2\sigma\sqrt{\Delta t} - \mu = \beta - 2\sigma\sqrt{\Delta t}\,.
\end{aligned}$$

The valid branching probabilities can be derived by solving Eqs. (3)–(5) again.

Pricing European-style single-barrier options on the BTT can be sped up by the efficient combinatorial algorithm of Lyuu (1998), which evaluates the option values at nodes A, B, and C in $O(n)$ time. The final pricing result is obtained by applying Eq. (6), where $\Delta t'$ is replaced by Δt.

Vanilla Options

To drastically reduce the nonlinearity error, the BTT must hit the exercise price X at the 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 BTT for pricing a vanilla option mimics that for pricing a single-barrier option except that the underlying grid should have a layer coinciding with X instead of L. The term "terminal node" refers to a node at the maturity date of the CRR tree. The value of a call option priced by an n-time-step CRR tree can be efficiently calculated by summing the value contributed by each terminal node:

$$e^{-rT} \sum_{i=0}^{n} \binom{n}{i} p^i (1-p)^{n-i} \max\left(S_0 u^i d^{n-i} - X, 0\right). \tag{7}$$

This formula can be evaluated in $O(n)$ time (see Lyuu (2002)).

4 An $O(n)$-Time Algorithm for Double-Barrier Options

An efficient combinatorial algorithm for pricing double-barrier options on the CRR tree is presented in this section. First, a useful formula derived in Lyuu (2002) and

Dai and Lyuu (2005) is given. This combinatorial formula is then used to build up the BTT pricing algorithm. We focus on the pricing of knock-in double-barrier call options. Extension to other types of double-barrier options is straightforward.

4.1 A Combinatorial Formula

Refer to Fig. 7 in the following. This grid reflects the structure of the CRR tree model. The x-coordinate denotes the time step of the CRR tree, and the y-coordinate denotes the logarithmic price. To mimic stock price movements on the CRR tree model, each path in the grid can move from node (i, j) to node $(i+1, j+1)$ (the up move) or node $(i+1, j-1)$ (the down move). To price a double-barrier option, we need to count the number of price paths form $(0, -a)$ (node A) to $(n, -b)$ (node B) that reach either barrier H or barrier L at least one. Without loss of generality, we assume that $a, b \geq 0$. The count turns out to be

$$\mathbf{N}(a, b, s) = \sum_{i=1}^{\lceil \frac{n}{s} \rceil} (-1)^{i+1} (|\alpha_i| + |\beta_i|), \tag{8}$$

where s denotes the distance between barrier L and barrier H and

$$|\,\alpha_i\,| = \begin{cases} \binom{n}{\frac{n+a+b+(i-1)\,s}{2}} & \text{for odd } i \\ \binom{n}{\frac{n+a-b+is}{2}} & \text{for even } i \end{cases} \qquad |\,\beta_i\,| = \begin{cases} \binom{n}{\frac{n-a-b+(i+1)\,s}{2}} & \text{for odd } i \\ \binom{n}{\frac{n-a+b+is}{2}} & \text{for even } i \end{cases} \tag{9}$$

The above formula is derived by applying the reflection principle and the inclusion-exclusion principle. Appendix B sketches the ideas.

4.2 The Pricing Algorithms

The pricing algorithm varies slightly under different scenarios. We first consider the degenerate case $X \geq H$. In this case, if the knock-in double-barrier call option is in the money at maturity, this barrier option must also come into existence. Thus the value of this double-barrier call option equals the value of an otherwise identical vanilla call option. From now on, we focus on $X < H$.

The price is obtained by summing the value contributed by each terminal node of the CRR tree much like direct integration. Combinatorial techniques are required to implement this idea efficiently. To explain how we develop the pricing algorithms clearly, we place the CRR tree on a grid as displayed in Fig. 8. The barriers H and L equal $S_0 u^{n-h} d^h (= S_0 u^{n-2h})$ and $S_0 u^{n-l} d^l (= S_0 u^{n-2l})$, respectively. The exercise price X satisfies the following equalities $S_0 u^{n-a} d^a \leq X < S_0 u^{n-a+1} d^{a-1}$ for some integer a.

Now we analyze the option value contributed by a price path that reaches terminal node $N(n, j)$. The probability for this price path is $p^{n-j}(1-p)^j$. The payoff at node $N(n, j)$ is $\max(S_0 u^{n-j} d^j - X, 0)$. Thus the value contributed by this price path is

$$p(j) \equiv e^{-rT} p^{n-j} (1-p)^j \max(S_0 u^{n-j} d^j - X, 0) \tag{10}$$

if this price path hits the barrier (i.e., the barrier option comes into existence). Furthermore, the value contributed by terminal node $N(n, j)$ is

$$V(j) \equiv \binom{n}{j} p(j) \tag{11}$$

if node $N(n, j)$ is above the barrier H (inclusive) or below the barrier L (inclusive). This is because the number of price paths reaching node $N(n, j)$ is $\binom{n}{j}$, and any price path reaching this node must also hit a barrier.

The algorithms for pricing the knock-in double-barrier call option are now within reach. Two non-degenerate cases are discussed as follows.

Case 1. $L < X < H$:

The option value can be decomposed into two parts: (1) the value contributed by the terminal nodes between X and H (exclusive), and (2) the value contributed by the terminal nodes above H (inclusive).

The first part of the option value is by accumulating the values contributed by the terminal nodes $N(n, j)$, $h < j < a$, between X and H. The number of paths that reach one of the barriers before reaching $N(n, j)$ is $\mathbf{N}(n-2h, 2j-2h, 2l-2h)$ as defined in Eq. (8). The value contributed by such a path is $p(j)$ from Eq. (10). Therefore, the value contributed by node $N(n, j)$ is $\mathbf{N}(n-2h, 2j-2h, 2l-2h) p(j)$. The sum of the values contributed by the terminal nodes between X and H is therefore

$$\sum_{j=h+1}^{a-1} \mathbf{N}(n-2h, 2j-2h, 2l-2h)p(j). \tag{12}$$

The second part of the option value is by accumulating the values contributed by the terminal nodes $N(n,i)$, $0 \leq i \leq h$, above the barrier H (inclusive). The value contributed by $N(n,i)$ is $V(i)$ (see Eq. (11)). Therefore, the second part of the option value is

$$\sum_{i=0}^{h} V(i). \tag{13}$$

We conclude that the value of a knock-in double-barrier call option is (12)+(13).

Case 2. $X \leq L$:

The option value can be decomposed into three parts: (1) the option value contributed by the terminal nodes between L (exclusive) and H (exclusive), (2) the option value contributed by the terminal nodes above H (inclusive), and (3) the option value contributed by the terminal nodes between L (inclusive) and X.

The first part of the option value is by accumulating the values contributed by the terminal nodes between L and H as follows:

$$\sum_{j=h+1}^{l-1} \mathbf{N}(n-2h, 2j-2h, 2l-2h)p(j). \tag{14}$$

The second part of the option value is by accumulating the values contributed by the terminal nodes above the barrier H (inclusive) and equals Eq. (13). The third part of the option value is by accumulating the option value contributed by the terminal nodes $N(n,k)$, $l \leq k < a$, between L (inclusive) and X. As the value contributed by $N(n,k)$ is $V(k)$ (see Eq. (11)), the third part is

$$P_1' = \sum_{k=l}^{a} V(k). \tag{15}$$

Finally, the value of a knock-in double-barrier call option is (13)+(14)+(15).

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. All the running time measurements are obtained by running programs on a Pentium-4 2.8GHz computer.

Vanilla Options

We first compare the performance of the CRR, Klassen's binomial tree model, the AMM, and the BTT on pricing vanilla options. In the setting of Fig. 9, the theoretical option value given by the Black-Scholes formula is 17.7943. All models converge to the theoretical option value as $n \to \infty$; however, their behaviors differ. The CRR oscillates significantly. Klassen's binomial model converges more smoothly but slowly. The AMM seems to converge more smoothly than the above two methods. Finally, the BTT converges most smoothly among all methods.

Single-Barrier Options

We next compare the BTT and Ritchken's (1995) trinomial tree model in pricing a European-style down-and-out single-barrier call option (see Fig. 10). The numerical settings and the true value are from Ritchken (1995). The x-axis and the y-axis denote the computation time and the option value, respectively. For example, it costs 0.014 second to compute a 350-time-step Ritchken's trinomial tree to obtain 5.998 (point A). It costs almost the same time to compute a 4,500-time-step BTT to obtain 5.997 (point B). Both Ritchken's model and the BTT converge well to the true value 5.9968, but the BTT converges more smoothly and faster.

It is a well-known hard problem to price a barrier option when the barrier is very close to the initial stock price: the so-called barrier-too-close problem. Table 1 reveals this fact. The numerical results of Ritchken's model in the second column and the true values listed in the last column are from Ritchken (1995). Each row contains information about the number of time steps and the computational time required by each approach to achieve 3-digit accuracy. Indeed, a large number of time steps is required for Ritchken's trinomial tree model to ensure that the barrier is exactly hit. Ritchken's approach is thus very inefficient in this case. On the other hand, although a much larger number of time steps is required for the BTT to achieve 3-digit accuracy, the BTT consumes much less time than Ritchken's model. This is because the BTT costs only $O(n)$ time instead of Ritchken's $\Omega(n^2)$ time. The AMM is another algorithm that can efficiently solve the barrier-too-close problem. All the numerical results for the AMM in Table 1 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. We conclude that the BTT is superior to the AMM and Ritchken's trinomial tree model when addressing the barrier-too-close problem.

Double-Barrier Options

We first compare Ritchken's model and the BTT in pricing a European-style knock-out double-barrier call. The numerical results are illustrated in Fig. 11. The x-axis and the y-axis denote the computational time and the option value, respectively. The parameters and the accurate value 1.4580 are from Ritchken (1995). Note that both methods converge well to the accurate value. But the BTT converges much more smoothly and faster.

Second, the BTT solves the barrier-too-close problem efficiently. Numerical evidence for pricing a knock-out double-barrier call is given in Fig. 12. The x-axis and the y-axis denote the number of time steps and the option value, respectively. The initial stock price 90.05 is very close to the lower barrier 90. It can be observed that the BTT converges smoothly and quickly to 10.55.

The AMM efficiently solves the barrier-too-close problem when pricing single-barrier options. When the AMM is applied to pricing double-barrier options without modifications, the pricing results are inaccurate, and the experimental results in Table 2 reveal this fact. The first column ("Level") in Table 2 denotes the level of the AMM, and the second column (n) denotes the number of time steps of the AMM. Note that

n is determined by the AMM level. The third column ("Inner Barrier") and the fourth column ("Outer Barrier") denote the pricing results of the AMM computed by moving the upper barrier (120) to the inner barrier or the outer barrier, respectively. When pricing a knock-out barrier option, the AMM underestimates the option value if the upper barrier moves down to the inner barrier. On the other hands, the AMM overestimates the option value if the upper barrier moves up to the outer barrier. Both phenomena are as expected. Each pricing result of the BTT in the same table is selected so the number of time steps of the BTT is close to, but does not exceed, that of the AMM. Clearly, the BTT converges more accurately and smoothly than the AMM as it generates results between the lower and the upper bounds of the AMM.

6 Conclusion

This paper proposes a novel, accurate, and efficient tree model, the bino-trinomial tree (BTT) model, for pricing a wide variety of derivatives. The BTT combines a CRR binomial tree structure and a root node with a trinomial structure. To reduce the nonlinearity error, the BTT structure can be adjusted by (1) finding a suitable length for the time step of the CRR tree, (2) properly laying out the CRR tree, and (3) locating the position of the root node and calculating its branching probabilities. We use vanilla options, single-barrier options, and double-barrier options to show how to construct the BTT for them. Numerical results are given to confirm the superiority of the BTT. Interestingly, our method outperforms other methods in solving the notorious barrier-too-close problem.

References

[1] Bhagavatula, R.S., and Carr, P.P. (1997) "Valuing Double Barrier Options with Fourier Series." Manuscript, New York University.

[2] Boyle, P., and Lau, S. (1994) "Bumping against the Barrier with the Binomial Method." *Journal of Derivatives*, 1 , pp. 6–14.

[3] Cox, J., Ross, S., and Rubinstein, M. (1979) "Option Pricing: A Simplified Approach." *Journal of Financial Economics*, 7 , pp. 229–264.

[4] Dai, T.S., Lyuu, Y.D., and Liu, L.M. (2006) "Developing Efficient Option Pricing Algorithms by Combinatorial Techniques." In *Proceedings of the 2006 International Conference on Scientific Computing*, Las Vegas, USA. pp. 104–110.

[5] Dai, T.S., and Lyuu, Y.D. (2005) "Pricing Double Barrier Options by Combinatorial Approaches." *Advances in Soft Computing — Soft Computing as Transdisciplinary Science and Technology*, Singapore: Springer-Verlag, pp. 1131–1140.

[6] Derman, E., Kani, I., Ergener, D., and Bardhan, I. (1995) "Enhanced Numerical Methods for Options with Barriers." Working Paper, Goldman Sachs.

[7] Duffie, D. (1996) *Dynamic Asset Pricing Theory.* 2nd ed. Princeton, NJ: Princeton Univ. Press.

[8] 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.

[9] Geman, H., and Yor, M. (1996) "Pricing and Hedging Double-Barrier Options: A Probabilistic Approach." *Mathematical Finance,* 6, pp. 365–378.

[10] Klassen,T.R. (2001) "Simple, Fast, and Flexible Pricing of Asian Options." *Journal of Computational Finance*, 4, pp. 89–124.

[11] Kunitomo, N., and Ikeda, M. (1992) "Pricing Options with Curved Boundaries." *Mathematical Finance,* 2, pp. 275–298.

[12] Hull, J. (2000) *Options, Futures, and Other Derivatives.* 4th ed. Englewood Cliffs, NJ: Prentice-Hall.

[13] Luo, L. (2001) "Various Types of Double Barrier Options." *Journal of Computational Finance*, 4, pp. 125–138.

[14] Lyuu, Y.D. (1998) "Very Fast Algorithms for Barrier Option Pricing and the Ballot Problem." *Journal of Derivatives*, 5, pp. 68–79.

[15] Lyuu, Y.D. (2002) *Financial Engineering & Computation: Principles, Mathematics, Algorithms.* Cambridge, U.K.: Cambridge University Press.

[16] Merton, R. (1973) "Theory of Rational Option Pricing." *Bell Journal of Economics and Management,* 4, pp. 141–183.

[17] Omberg, E. (1987) "A Note on the Convergence of Binomial-Pricing and Compound-Option Models." *Journal of Finance,* 42, pp. 463–469.

[18] Pelsser, A. and Vorst, C. (1994) "The Binomial Model and the Greeks." *Journal of Derivatives,* 1, pp. 45–49.

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

[20] Reiner, E. and Rubinstein, M. (1991) "Breaking Down the Barriers." *Risk,* 4, pp. 28–35.

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

[22] Sidenius, J. (1998) "Double Barrier Options: Valuation by Path Counting." *Journal of Computational Finance,* 1, pp. 63–79.

A Validity of Risk-Neutral Probabilities

Define

$$
\begin{aligned}
\det &= (\beta-\alpha)(\gamma-\alpha)(\gamma-\beta),\\
\det_u &= (\beta\gamma+\mathrm{Var})(\gamma-\beta),\\
\det_m &= (\alpha\gamma+\mathrm{Var})(\alpha-\gamma),\\
\det_d &= (\alpha\beta+\mathrm{Var})(\beta-\alpha).
\end{aligned}
$$

Then Cramer's rule applied to Eqs. (3)–(5) gives $P_u = \det_u/\det$, $P_m = \det_m/\det$, and $P_d = \det_d/\det$. Note that $\det < 0$ because $\alpha > \beta > \gamma$. To ensure that the branching probabilities are valid, it suffices to show that P_u, P_m, $P_d \geq 0$. As $\det < 0$, it is sufficient to show $\det_u, \det_m, \det_d \leq 0$ instead. Finally, as $\alpha > \beta > \gamma$, it suffices to show that $\beta\gamma + \mathrm{Var} \geq 0$, $\alpha\gamma + \mathrm{Var} \leq 0$, and $\alpha\beta + \mathrm{Var} \geq 0$ under the premise $\beta \in [-\sigma\sqrt{\Delta t}, \sigma\sqrt{\Delta t})$. Indeed,

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

as desired.

B A Useful Combinatorial Formula

The following analysis is based on the grid in Fig. 7. Before counting the number of price paths from node A to node B and reaching either barrier L or H at least once, we consider a simplified problem first: How many price paths from node A to node B will hit barrier H before one hit of barrier L? One such path may hit barrier H first at J and barrier L later at K. We can reflect the path $\widehat{AJ}$ marked by the solid curve with respect to the H-axis to obtain path $\widehat{A_1 J}$ marked by the dash curve. Each path from node A to node J maps to a unique path from node A_1 to node J, and vice versa. Thus the number of paths from node A to node J equals the number of paths from node A_1 to node J. This is the celebrated reflection principle. So the number of paths from node A to node B and hitting barrier H equals the number of paths from node A_1 to node B.

The reflection principle can be applied more than once. The curve $\widehat{A_1 K}$ can be reflected with respect to the L-axis to obtain $\widehat{A_2 K}$. Therefore, the number of paths from node A_1 to node B and reaching barrier L at least once equals the number of paths from node A_2 to node B. In conclusion, the number of paths from node A to node B and reaching barrier H at least once before reaching barrier L equals the number of paths from A_2 to B.

Assume that x up moves and y down moves are required to move from node A_2 (with coordinate $(0, -(a + 2s))$) to node B (with coordinate $(n, -b)$). They must satisfy

$$
\begin{aligned}
x + y &= n, \\
x - y &= a - b + 2s,
\end{aligned}
$$

which yield $x = (n+a-b+2s)/2$. So the answer to this simplified problem mentioned above is

$$
\binom{n}{\frac{n+a-b+2s}{2}} \quad \text{for even, non-negative } n + a - b. \tag{16}
$$

Note that a path counted by Eq. (16) may hit L first before hitting H. The point is that among the hits, one hit of barrier H must appear before one hit of barrier L.

The answer to the number of paths that will hit either barrier L or H before arriving at node B is now within reach. It is useful to consider a function f that maps paths to strings. The strings contain the information about the barrier hitting sequences. For example, $f(\widehat{AB}) = HHL$ since the path $\widehat{AB}$ hits the barrier H twice before hitting the barrier L. Next, we define α_i as the set of paths whose f value *contains* $\overbrace{H^+ L^+ H^+ \cdots}^{i}$ with $i \geq 1$. Here, L^+ and H^+ denote a sequence of Ls and Hs, respectively. For example, the path $\widehat{AB}$ belongs to both set α_1 and set α_2. Similarly, define β_i as the set of paths whose f value contains $\overbrace{L^+H^+L^+\cdots}^{i}$ with $i \geq 1$. For example, the path $\widehat{AB}$ belongs to set β_1. The number of elements in set α_i and β_i can be calculated by repeatedly using the reflection principle. The number of elements in each set is described in Eq. (9). Note that each path that hits the barrier may belong to more than one set. For example, $\widehat{AB}$ in Fig. 7 belongs to set α_1, α_2, and β_1. To avoid counting paths more than once, the inclusion-exclusion principle is applied to calculate the exact number of paths from node A to node B and reaching either barrier L or H at least once, which is done in Eq. (8).

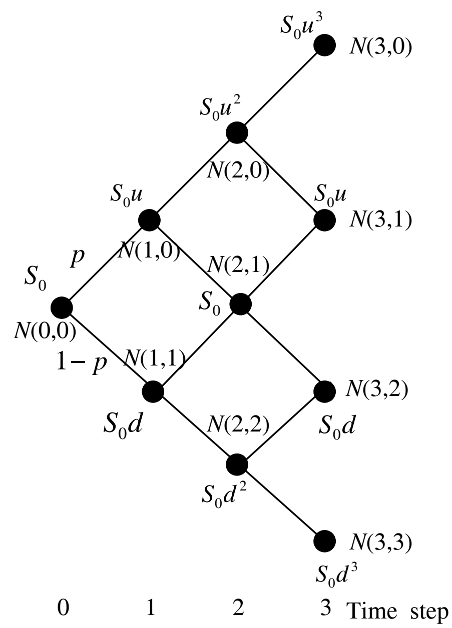

Figure 1: **T**he CRR Tree. The initial stock price is S_0. The upward and downward multiplicative factors for the stock price are u and d, respectively. The upward and downward branching probabilities are p and $1-p$, respectively. $N(i,j)$ denotes a node at time step i resulting from j down moves and $i-j$ up moves from time step 0.

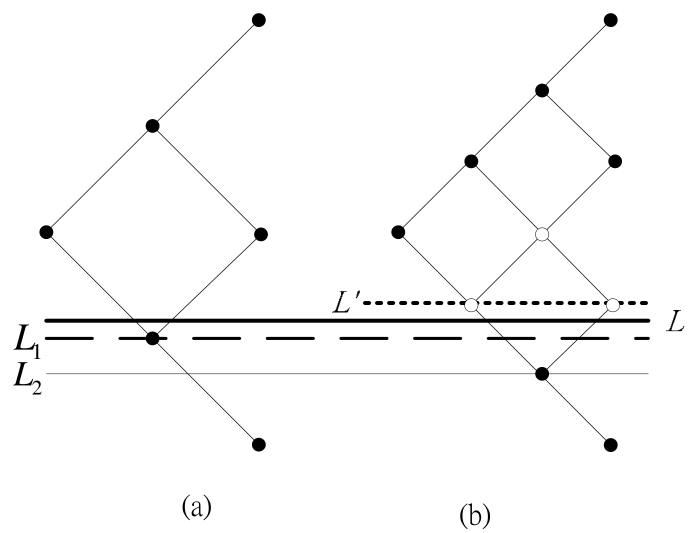

Figure 2: **B**arrier and the CRR Tree Model at Different Numbers of Time Steps. The contractual barrier is denoted by L. (a) The effective barrier for the 2-time-step tree is L_1. (b) The effective barrier for the 3-time-step tree is L_2. L' and L_2 are the inner and the outer barriers to L, respectively.

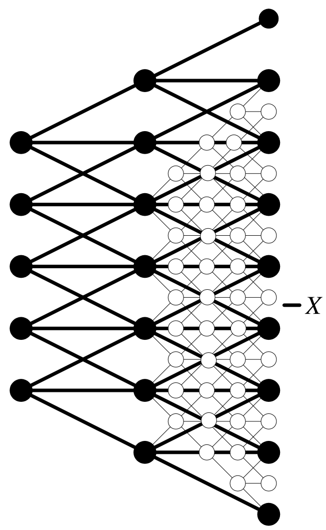

Figure 3: **P**ricing a Vanilla Option by the AMM. The base tree is denoted by thick lines and solid circles. The finer trees are denoted by thin lines and hallow circles. The strike price X is also marked.

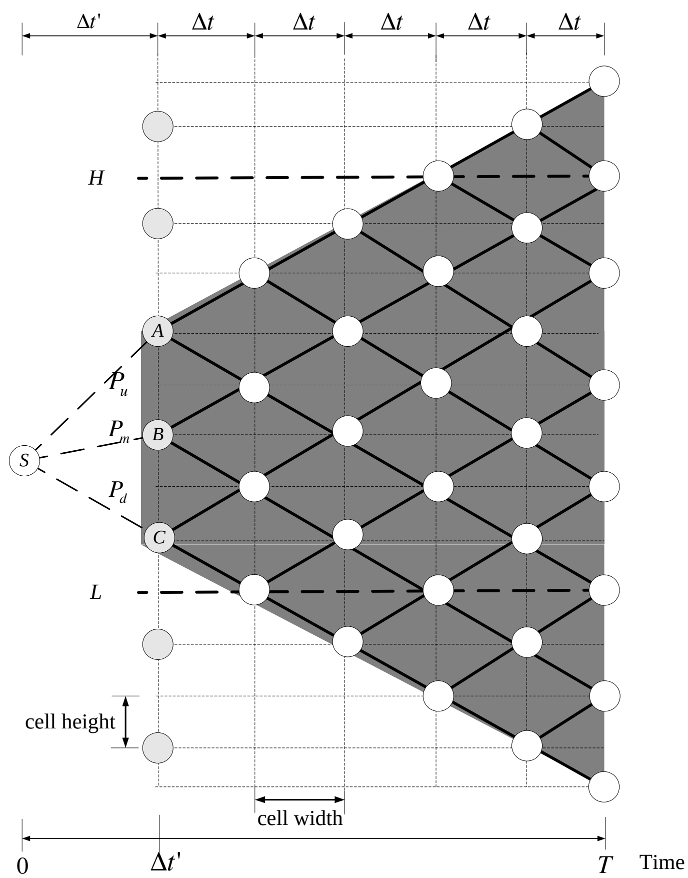

Figure 4: **T**he BTT for Pricing Double-Barrier Options. Two barriers, L and H, are denoted by thick dashed lines. The root of the BTT is denoted by node S. The CRR tree (with the first two time steps truncated) that comprises the bulk of the BTT is shadowed. This CRR tree is placed on a grid (thin lines) that as two layers coinciding with the barriers.

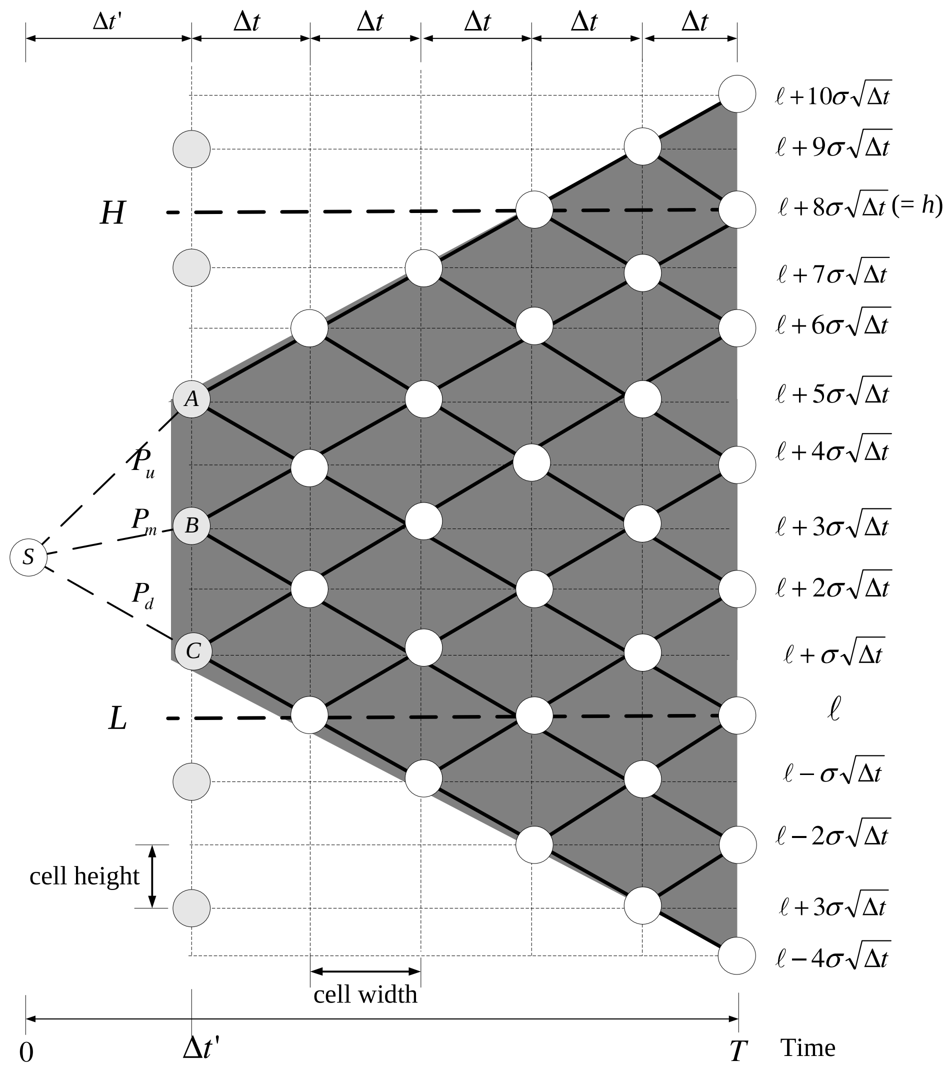

Figure 5: **T**he BTT for Pricing Double-Barrier Options. Two barriers, L and H, are denoted by thick dashed lines. The root of the BTT is denoted by node S. The CRR tree (with the first two time steps truncated) that comprises the bulk of the BTT is shadowed. This CRR tree is put on a grid (marked in thin lines). The log-prices for the nodes at maturity are listed next to these nodes.

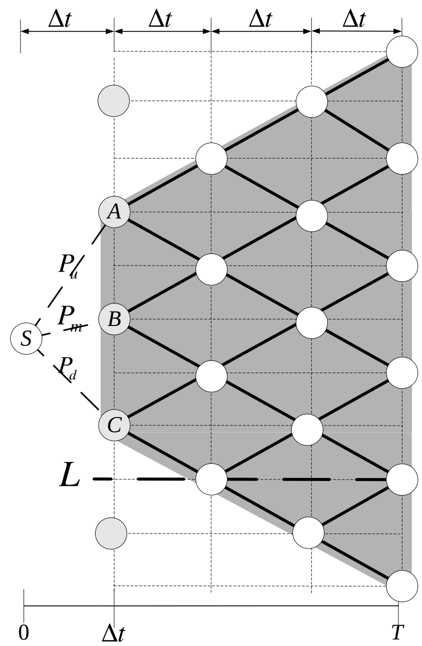

Figure 6: **T**he BTT for Pricing Single-Barrier Options. Barrier L is denoted by thick dotted lines. The root of the BTT is denoted by node S. The CRR tree (with the first two time steps truncated) that comprises the bulk of the BTT is shadowed. This CRR tree is placed on a grid in thin dotted lines.

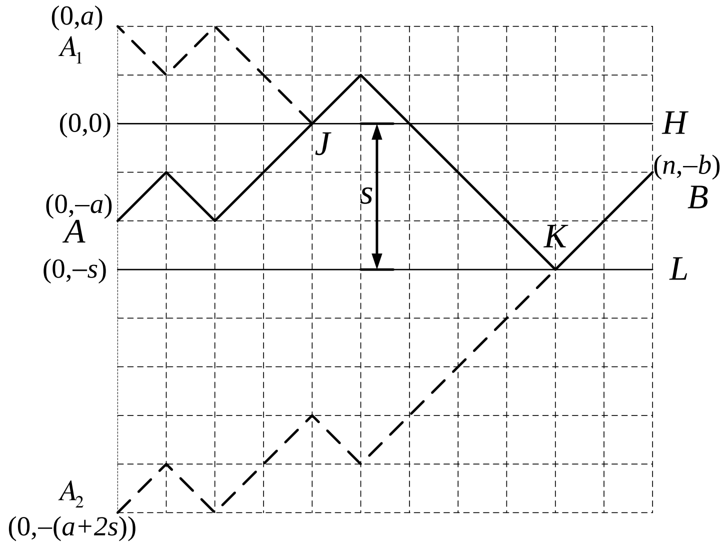

Figure 7: **Paths That Hit** H **or** L**.** Barrier H is the horizontal line $y = 0$, and barrier L is the horizontal line $y = -s$.

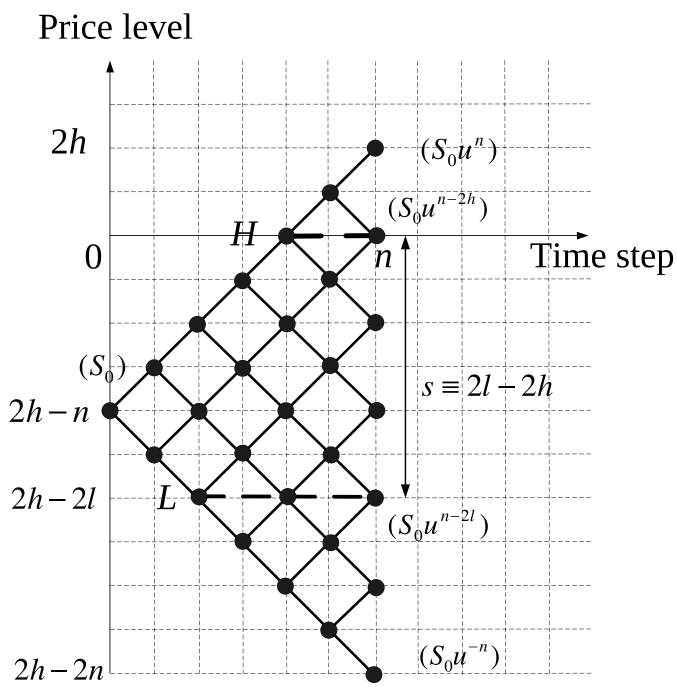

Figure 8: **Placing a CRR Tree on a Grid.** A CRR tree is placed on a grid. The x-axis and the y-axis of this grid are denoted by thin solid lines. The coordinate of the root node of the CRR tree is $(0, 2h - n)$. The two barriers L and H are denoted by thick dotted lines, and s denotes the distance between H and L. The values in parentheses denote the stock prices.

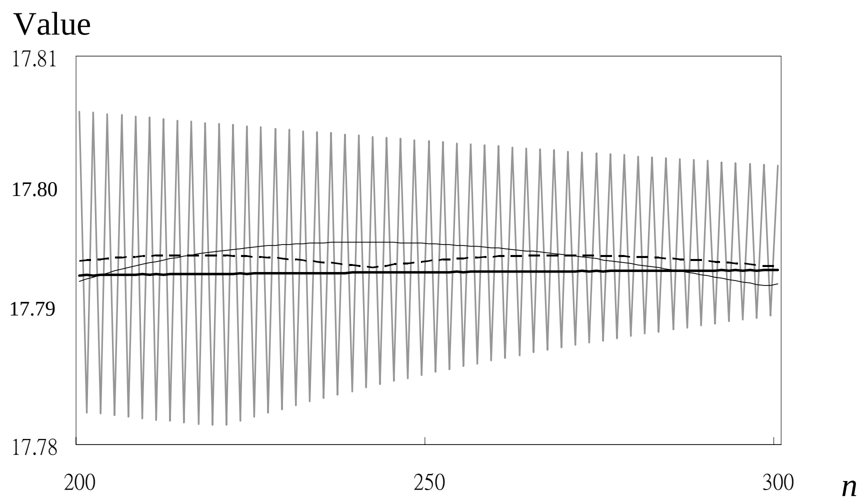

Figure 9: **Oscillations of the CRR, Klassen's Binomial Model, the AMM, and the BTT in Valuing Vanilla Call Options.** The x-axis denotes the number of time steps of each tree model. The y-axis denotes the option value. The initial stock price is 100, the exercise price is 98, the risk-free rate is 10% per annum, the volatility of the stock price is 30%, and the time to maturity is 1 year. The line that oscillates significantly is the CRR tree model. The line formed by hollow squares denotes Klassen's binomial tree model. The thin solid line denotes the AMM with AMM level 1. The dotted line denotes the AMM with AMM level 2. The thick solid line denotes the BTT. The analytical option value is 17.7943.

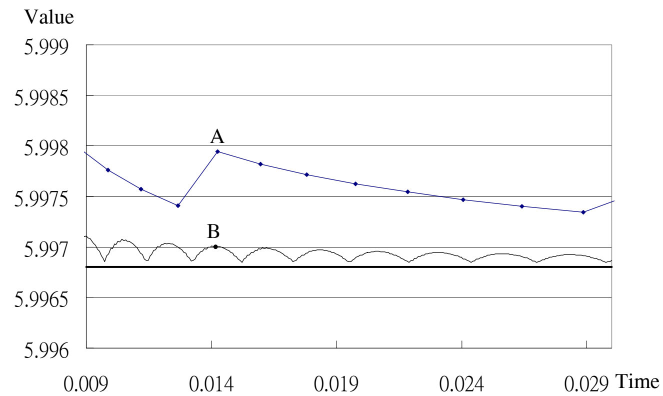

Figure 10: **Convergence of the BTT and Ritchken's Trinomial Model in Pricing a Down-and-Out Single-Barrier Call Option.** The initial stock price is 95, the exercise price is 100, the risk-free rate is 10% per annum, the volatility of the stock price is 25%, the time to maturity is 1 year, and the barrier is 90. The horizontal line $y = 5.9968$ denotes the true value, the line passing through point B denotes the BTT, and the line passing through point A denotes Ritchken's model.

Table 1: **Ritchken's Model, the BTT, and the AMM When the Barrier Is Close to the Initial Stock Price.**

Stock	`Ritchken`			`AMM`		`BTT`			True
Price	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

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).

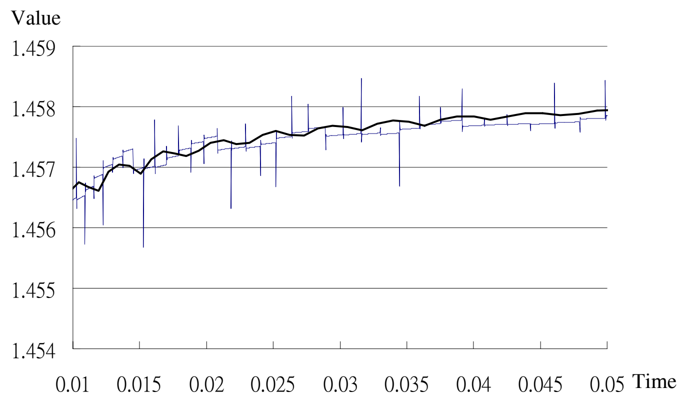

Figure 11: **Pricing Knock-Out Double-Barrier Options**. The initial stock price is 95, the exercise price is 100, the risk-free rate is 10% per annum, the volatility of the stock price is 25%, the time to maturity is 1 year, and the two barriers are 140 and 90. The thin curve (with large oscillations) and the thick curve denote the Ritchken's trinomial tree model and the BTT, respectively.

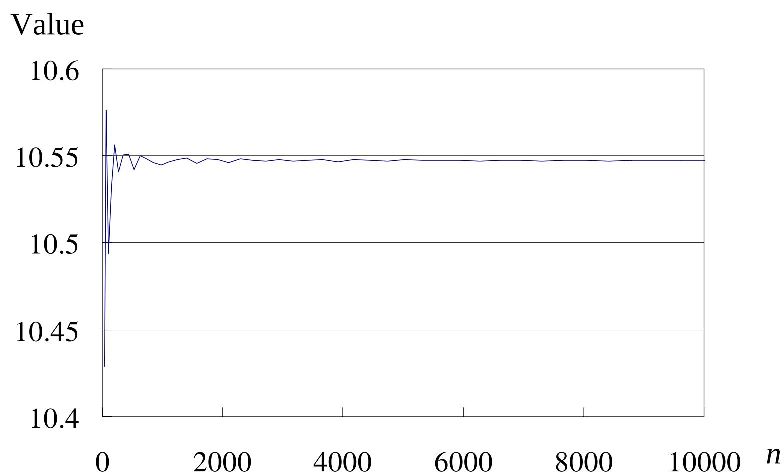

Figure 12: **Convergence of the BTT When the Barrier Is Close to the Initial Stock Price.** The initial stock price is 90.05, the exercise price is 100, the risk-free rate is 10% per annum, the volatility of the stock price is 25%, the time to maturity is 1 year, and the two barriers are 140 and 90.

Table 2: **Convergence Rates of the AMM and the BTT.**

AMM				BTT	
Level	n	Inner Barrier	Outer Barrier	n	Value
3	167	0.000000	0.000014	163	0.000002
2	671	0.000001	0.000015	655	0.000003
1	2686	0.000001	0.000005	2625	0.000003

A knock-out double-barrier call option is priced above. The initial stock price is 100, the exercise price is 100, the risk-free rate is 10% per annum, the volatility of the stock price is 30%, the time to maturity is 1 year, and the two barriers are 99.5 and 120. The pricing results of the BTT (running up to 20,000 time steps) converge to 0.000003.