# An Accurate and Provably Efficient GARCH Option Pricing Tree

Advisor: Prof. Yuh-Dauh Lyuu

Chi-Ning Wu

Department of Finance

National Taiwan University

**Abstract**

The trinomial-tree GARCH option pricing algorithms of Ritchken and Trevor (1999) and Cakici and Topyan (2000) are claimed to be efficient and accurate. However, this thesis proves that both algorithms generate trees that explode exponentially when the number of partitions per day, $n$, exceeds a typically small number determined by the model parameters. Worse, when explosion happens, the tree cannot grow beyond a certain maturity, making it useless for pricing derivatives with a longer maturity. Meanwhile, a small $n$ has accuracy problems and does not prevent explosion. This thesis then presents a trinomial-tree GARCH option pricing algorithm that solves the above problems. The algorithm provably does not have the short-maturity problem. Furthermore, the tree size is guaranteed to be quadratic if $n$ is less than a threshold easily determined by the model parameters. This result for the first time puts a tree-based GARCH option pricing algorithm in the same complexity class as binomial and trinomial trees under the Black-Scholes model. Extensive numerical evaluation is conducted to confirm the analytical results and the accuracy of the algorithm.

**Keywords:** GARCH model, trinomial tree, path dependency, option pricing, computational complexity, Black-Scholes model

# Contents

# List of Figures

# List of Tables

# 1 Introduction

In the numerical pricing of derivatives, the continuous-time diffusion process for the asset price is often discretized to yield a discrete-time tree first. Derivatives are then priced on the tree by the standard backward-induction algorithm. The lognormal diffusion, for instance, gives rise to the well-known CRR binomial tree of Cox, Ross, and Rubinstein (1979). Two critical features of the CRR tree, as well as its many trinomial variations, are that it recombines and that an $m$-period tree contains only $O(m^2)$ nodes, a quadratic growth (see Fig. 1). As a consequence, simple derivatives such as vanilla options, barrier options, and lookback options can be efficiently priced as shown in Lyuu (2002). However, a polynomial-sized tree may still give rise to an exponential-time pricing algorithm if the derivative itself is complex. (Exponential-time algorithms are said to suffer from combinatorial explosion.) The Asian option fits this characterization because the vast amount of extra states needed by its path-dependent feature makes pricing on an $m$-period tree take time exponential in $m$. Approximations are therefore mandatory.

A qualitatively more difficult problem emerges when the explosion arises from the model itself. If the model generates trees that do not recombine, pricing is expensive even for simple derivatives like vanilla options. For example, when the volatility is not a constant, such as the interest rate model of Cox, Ingersoll, and Ross (1985), a brute-force discretization leads to exploding binomial trees that do not recombine. The problem may be rectified by the technique of Nelson and Ramaswamy (1990) to transform the diffusion process into one with a constant volatility. But the methodology does not guarantee to do away with combinatorial explosion. This issue is particularly relevant when the diffusion process is bivariate. As an example, Chien (2003) demonstrates that the bivariate interest rate tree of Ritchken and Sankarasubramanian (1995) gives rise to exponential-sized trees. The focus of the thesis is also bivariate: the tremendously influential GARCH (generalized autoregressive conditional heteroskedastic) option pricing model.

Bollerslev (1986) and Taylor (1986) independently propose the GARCH process for modeling the stochastic volatility of asset returns. Since then, the model has been generalized and used extensively in the finance literature on the modeling of time series; see Bollerslev et al. (1992) for a survey. As the model has received empirical support, its application to options pricing becomes very important. Duan (1995) is the first to propose a GARCH option pricing model. In terms of pricing algorithms, the massive path dependency of the model favors Monte Carlo simulation over trees. However, the Monte Carlo estimate is probabilistic, and options that can be exercised early (the so-called American options) cannot be priced easily with this method. The situation changes with the appearance of the trinomial tree of Ritchken and Trevor (1999). Other GARCH option pricing techniques include the Markov chain approximation of Duan and Simonato (2001), the Edgeworth tree approximation of Duan et al. (2002), and analytical approximations as in Heston and Nandi (2000).

This thesis investigates the performance of the Ritchken-Trevor algorithm and its modified version by Cakici and Topyan (2000). The results are negative, both theoretically and numerically. It is shown that the Ritchken-Trevor-Cakici-Topyan (RTCT) algorithm creates exponential-sized trees when the number of partitions per day, $n$, exceeds a typically small number. The tree is hence not efficient unless $n$ is small. As a consequence, raising $n$ as a way to improve accuracy can be very costly. Unfortunately, a small $n$ can result in inaccurate option prices; in fact, even the smallest choice of $n = 1$ can result in explosion. In practice, one may be willing to trade efficiency for accuracy by adopting a suitably large $n$. But it does not work here because when explosion occurs, the RTCT tree cannot grow beyond a certain maturity, making it useless for pricing derivatives with a longer maturity. Therefore, when explosion occurs, even infinite computing resources may not help, and the typical tradeoff between efficiency and accuracy is lost.

Because of the problems associated with the RTCT algorithm, this thesis then presents an efficient trinomial-tree GARCH option pricing algorithm that addresses the shortcomings of the RTCT algorithm. The RTCT tree, like typical trinomial trees, takes a flat middle branch from each node as shown in Fig. 2. Our key idea departs from that by making the tree's middle branch track the expected stock price as closely as possible. Therefore, we call it the mean-tracking (MT) algorithm. By tracking the mean, the two outer branches are expected to fan out less in their attempt to match the conditional mean and variance. This in turn leads to more compact trees. The effects are clearest for models with a stochastic volatility. The above argument is not only intuitive but also provable, made possible by the MT tree's simplicity.

The concept of mean tracking is not entirely new. It is explicit in Li et al. (1995) and implicit in Hull and White (1993), both dealing with the calibration of no-arbitrage interest rate models. However, the advantages of mean tracking have not been systematically investigated in theory or practice in the literature until recently. For example, Dai and Lyuu (2003) apply the mean-tracking idea to develop the first exact trinomial-tree Asian option pricing algorithm which breaks the long-standing $3^m$ bound with a provable running time of $3^{O(\sqrt{m})}$. Although still far from being efficient, it represents a substantial step towards a polynomial-time exact pricing algorithm for the Asian option.

The MT tree provably solves the short-maturity problem of the RTCT tree. Hence it accepts any $n$ without having to worry about the tree being cut short. The tradeoff between efficiency and accuracy is hence restored. Note that an exponential-sized MT tree is still useful (albeit costly). It is when a tree's maturity is cut short, as in the case of the exploding RTCT tree, that makes the tree far less applicable in practice.

Perhaps most unexpectedly, the MT tree's size becomes polynomial in maturity if $n$ does not exceed a certain threshold. In fact, the tree size is only quadratic, giving it the same complexity as the CRR tree under the Black-Scholes model. This level of efficiency creates great opportunities for the practical use of the MT tree in pricing. The MT tree is therefore the first tree-based GARCH option pricing algorithm that is provably efficient. Surprisingly, all these positive theoretical results can be proved with only elementary techniques.

Numerical experiments will demonstrate that a small $n$ gives accurate results. They also confirm the quadratic node count. Because the Cakici-Topyan (CT) version of the GARCH option pricing algorithm is slightly superior to the Ritchken-Trevor (RT) version, we will compare the MT tree with the CT tree in the numerical valuation of option prices.

The thesis is organized as follows. The GARCH model is presented in Section 2. Section 3 sketches the RTCT tree from which the MT tree derives. Differences between the two will be pointed out along the way. Section 4 probes into the problems with the RTCT algorithm. Section 5 presents the MT tree and its analysis. Section 6 evaluates the various GARCH option pricing algorithms numerically. Section 7 concludes.

## 2 The GARCH model

Let $S_t$ denote the asset price at date $t$ and $h_t$ the conditional volatility of the return over the $(t+1)$th day $[t, t+1]$. The following risk-neutral process for the logarithmic price $y_t \equiv \ln S_t$ is due to Duan (1995):

$$y_{t+1} = y_t + r - \frac{h_t^2}{2} + h_t \epsilon_{t+1}, \tag{1}$$

where

$$\begin{aligned} h_{t+1}^2 &= \beta_0 + \beta_1 h_t^2 + \beta_2 h_t^2 (\epsilon_{t+1} - c)^2, \qquad (2) \\ \epsilon_{t+1} &\sim N(0,1) \text{ given information at date } t, \\ r &= \text{daily riskless return}, \\ c &\geq 0. \end{aligned}$$

The model is bivariate as its state is described by $(y_t, h_t^2)$. It is postulated that $\beta_0, \beta_1, \beta_2 \geq 0$ to make the squared volatilities $h_t^2$ positive. We further impose $\beta_1 + \beta_2 < 1$ to make the model stationary. A positive $c$ represents a negative correlation between the asset return and the volatility. Updating rule (2) for the square volatility, due to Engle and Ng (1993), is also called the nonlinear asymmetric GARCH or NGARCH for short. Throughout the paper, $N$ will denote the maturity of the tree (in days) and that of the option to be priced by the tree.

## 3 Tree Building and Backward Induction

The RTCT trinomial tree approximates the continuous-state GARCH process with discrete states as follows. Partition a day into $n$ periods. Three successor states follow each state $(y_t, h_t^2)$ after a period. As the trinomial tree recombines within one day, $2n + 1$ states at date $t + 1$ follow each state at date $t$. We next pick the jump size for state $(y_t, h_t^2)$. Let $\gamma = h_0$ and $\gamma_n = \gamma/\sqrt{n}$. (Later, the MT algorithm will choose a different $\gamma$.) The tree is laid out in such a way that adjacent nodes at the same time are spaced by $\gamma_n$ in their logarithmic prices. The jump size will be some integer multiple $\eta$ of $\gamma_n$. We call $\eta$ the jump parameter. The jump parameter measures how much the two outer branches fan out around the middle branch as depicted in Fig. 3. The middle branch leaves the underlying asset's price unchanged. (Later, the MT algorithm will let the middle branch track the mean of $y_{t+1}$.) Figure 4 illustrates a 1-day trinomial tree, where each day is partitioned into $n = 3$ periods. The probabilities for the up, middle, and down branches are

$$p_u = \frac{h_t^2}{2\eta^2\gamma^2} + \frac{r - (h_t^2/2)}{2\eta\gamma\sqrt{n}}, \tag{3}$$

$$p_m = 1 - \frac{h_t^2}{\eta^2\gamma^2}, \tag{4}$$

$$p_d = \frac{h_t^2}{2\eta^2\gamma^2} - \frac{r - (h_t^2/2)}{2\eta\gamma\sqrt{n}}, \tag{5}$$

picked to match the conditional mean and variance of $y_{t+1}$ given $(y_t, h_t^2)$ in the limit. Therefore, the tree converges to the continuous-state model (2). Note that from Eqs. (3)–(5), valid branching probabilities exist (i.e., $0 \leq p_u, p_m, p_d \leq 1$) if and only if

$$\frac{|\, r - (h_t^2/2)\,|}{2\eta\gamma\sqrt{n}} \leq \frac{h_t^2}{2\eta^2\gamma^2} \leq \min(1 - \frac{|\, r - (h_t^2/2)\,|}{2\eta\gamma\sqrt{n}}, \frac{1}{2}). \tag{6}$$

The intraday nodes are dispensed with to create a $(2n+1)$-nomial tree as in Fig. 5 to reduce the node count by a factor of $n$. The resulting model is multinomial with $2n+1$ branches from any state $(y_t, h_t^2)$. Updating rule (2) must be modified to reflect the adoption of the discrete-state tree model. State $(y_t, h_t^2)$ at date $t$ is followed by state $(y_t + \ell\eta\gamma_n, h_{t+1}^2)$ at date $t+1$, where

$$\begin{aligned} h_{t+1}^2 &= \beta_0 + \beta_1 h_t^2 + \beta_2 h_t^2 (\epsilon'_{t+1} - c)^2, \\ \epsilon'_{t+1} &= \frac{\ell\eta\gamma_n - (r - h_t^2/2)}{h_t}. \\ \ell &= 0, \pm 1, \pm 2, \ldots, \pm n. \end{aligned} \tag{7}$$

This transition happens with probability

$$\sum_{j_u, j_m, j_d} \frac{n!}{j_u!\, j_m!\, j_d!}\, p_u^{j_u} p_m^{j_m} p_d^{j_d},$$

where $j_u, j_m, j_d \geq 0$, $n = j_u + j_m + j_d$, and $\ell = j_u - j_d$.

To ensure that the branching probabilities lie between 0 and 1, the RTCT tree may have to pick different jump parameters $\eta$ for different states. This implies varying jump sizes for the tree. As the requirement $p_m \geq 0$ implies

$$\eta \geq h_t/\gamma, \tag{8}$$

the RTCT algorithm goes through

$$\eta = \lceil h_t/\gamma \rceil, \lceil h_t/\gamma \rceil + 1, \lceil h_t/\gamma \rceil + 2, \ldots \tag{9}$$

until valid probabilities are obtained or until their nonexistence is confirmed by inequalities (6). The latter case means the tree cannot grow further. This search procedure makes the theoretical analysis of the RTCT tree very difficult.

Each squared volatility $h_t^2$ picks its own jump parameter $\eta$. Figure 6 depicts a 3-day tree with $n = 1$. The number of possible values of $h_t^2$ at a node equals the number of paths leading to the node, which can be exponentially many. For example, nodes A and B each have one $h_t^2$ and both pick $\eta = 2$. On the other hand, the two $h_t^2$'s at node C pick different jump sizes.

Instead of keeping track of all possible volatilities at every node, the RTCT algorithm creates $K$ volatilities between the maximum and minimum $h_t$ (inclusive) per node. Specifically, the squared volatilities are equally spaced; they are

$$h_{\min}^2 + j\,\frac{h_{\max}^2 - h_{\min}^2}{K-1}, \quad j = 0, 1, 2, \ldots, K-1,$$

where $h_{\min}$ and $h_{\max}$ denote the minimum and maximum volatilities at the node. We call it the linear interpolation scheme. A different distribution will be used by the MT algorithm. Specifically, the logarithms of squared volatilities are equally spaced between $\ln h_{\min}^2$ and $\ln h_{\max}^2$; they are

$$\exp\left[\ln h_{\min}^2 + j\,\frac{\ln h_{\max}^2 - \ln h_{\min}^2}{K-1}\right], \quad j = 0, 1, 2, \ldots, K-1. \tag{10}$$

Smaller volatilities are thus sampled more finely than larger volatilities. We call it the log-linear interpolation scheme.

The volatilities added between the minimum and maximum volatilities at a node will be called interpolated volatilities. For the CT tree, the minimum and maximum volatilities are indeed true volatilities generated by following updating rule (7) of the discrete-state tree model, starting from beginning state $(y_0, h_0^2)$. Interpolated volatilities are artifacts in the sense that they are not generated this way. For the RT tree, however, even the minimum or maximum volatility may be the result of applying the updating rule to an interpolated volatility of the previous date. Hence they may be artifacts, too. See Fig. 7 for illustration.

After the tree is built, backward induction commences. For a volatility $h_{t+1}$ following state $(y_t, h_t^2)$ via updating rule (7), the algorithm finds the two volatilities that bracket $h_{t+1}$. Note that $h_t$ may be an interpolated volatility. The option price corresponding to $h_{t+1}$ is then interpolated linearly from the option prices corresponding to the bracketing volatilities. Figure 8 illustrates the procedure for a branch. After the option prices from all $2n+1$ branches are available, the option price for state $(y_t, h_t^2)$ is finally calculated as their average discounted value based on the branching probabilities. The above approximation paradigm is due to Hull and White (1993) and Ritchken, Sankarasubramanian, and Vijh (1993).

The CT algorithm conceals a serious problem not shared by the RT algorithm. The CT algorithm keeps only the minimum and maximum volatilities at a node in growing the tree. The $K-2$ interpolated volatilities per node are created only after the tree has been built but before the backward induction starts. But an interpolated volatility's successor volatility may reach an unreachable node, which has no option prices at all! When this happens, backward induction cannot continue. Such rare situations do arise when $n$ and $N$ are both large. Following the RT algorithm, the MT algorithm uses all $K$ volatilities of a node in growing the tree, not just the minimum and maximum volatilities. The above-mentioned problem therefore never arises as the interpolated volatilities' branches have been followed through in the tree-building process.

# 4 Problems with the RTCT Algorithm

## 4.1 Explosion and Shallowness

Pricing with the RTCT algorithm is claimed to be efficient and accurate. Unfortunately, the RTCT algorithm explodes exponentially when $n$ exceeds a certain threshold. Worse, when explosion occurs, the tree is forced to be shallow. More specifically, the RTCT tree will be cut short (or, short-dated). Even if one is willing to accept long running time, the tree may not grow to the needed maturity. An exploding tree is therefore of limited use in practice. In this section, we discuss the problems of explosion and shallowness concealed in the RTCT algorithm.

### 4.1.1 Sufficient Conditions for Explosion

One typically increases $n$ for better accuracy. Unfortunately, the largest value of $h_t$ at date $t$ grows exponentially in $t$ if $n$ exceeds a threshold. When this happens, the value of $\eta$ also grows exponentially by virtue of relation (8). Note that the $2n+1$ nodes reached from a state span $1+2n\eta$ nodes (recall Fig. 5). So when explosion happens, the RTCT tree explodes. The RTCT tree must thus be restricted to small $n$'s to have any hope of being efficient. However, a small $n$ is no guarantee that the tree will not explode, as will be been later.

We now provide the argument for the claimed exponential growth of the largest value of $h_t$ at date $t$. Assume $r=0$ and $c=0$ first. Updating rule (7) is now

$$h_{t+1}^2 = \beta_0 + \beta_1 h_t^2 + \beta_2 \left[\, \ell\eta\gamma_n + (h_t^2/2) \,\right]^2, \quad \ell = 0, \pm 1, \pm 2, \ldots, \pm n.$$

To make $h_{t+1}^2$ as large as possible, set $\ell = n$. The updating rule becomes

$$\begin{aligned}
h_{t+1}^2 &= \beta_0 + \beta_1 h_t^2 + \beta_2 \left[\, \sqrt{n}\,\eta\gamma + (h_t^2/2) \,\right]^2 \quad \text{because } \gamma_n = \gamma/\sqrt{n} \\
&\geq \beta_0 + \beta_1 h_t^2 + \beta_2 \left[\, \sqrt{n}\, h_t + (h_t^2/2) \,\right]^2 \quad \text{because } \eta\gamma \geq h_t \\
&\geq \beta_0 + \beta_1 h_t^2 + \beta_2 n h_t^2 \\
&= \beta_0 + (\beta_1 + \beta_2 n)\, h_t^2.
\end{aligned}$$

By induction,

$$
\begin{aligned}
h_{t+1}^2 &\geq \beta_0 \sum_{i=0}^{t} (\beta_1 + \beta_2 n)^i + (\beta_1 + \beta_2 n)^{t+1} h_0^2 \\
&= \frac{\beta_0}{1 - (\beta_1 + \beta_2 n)} + [\, h_0^2 + \frac{\beta_0}{(\beta_1 + \beta_2 n) - 1} \,](\beta_1 + \beta_2 n)^{t+1}.
\end{aligned}
$$

The above expression grows exponentially in $t$ if

$$\beta_1 + \beta_2 n > 1.$$

This inequality is reminiscent of the necessary condition $\beta_1 + \beta_2 \geq 1$ for GARCH to be nonstationary. When $r \neq 0$ or $c \neq 0$, the largest value of $h_t$ at date $t$ still grows exponentially in $t$ as long as $n$ is suitably large. The argument is more tedious but essentially identical. We conclude that the RTCT tree grows exponentially if $n$ is large enough.

Ritchken and Trevor (1999) add extra volatilities between the maximum and minimum $h_t^2$ at each node in building the tree. Our proof depends entirely on showing that the largest value of $h_t$ at date $t$ grows exponentially. These additional volatilities serve only to increase the largest value of $h_t$. Hence the same conclusion for exponential explosion stands for the Ritchken-Trevor version.

### 4.1.2 The Shallowness of an Exploding Tree

Can a large $n$ be chosen to improve accuracy if we are willing to accept long running times? Unfortunately, the RTCT tree does not admit such a tradeoff. The reason is that there is a ceiling on volatility $h_t$ for valid branching probabilities to exist for state $(y_t, h_t^2)$. With the maximum value of $h_t$ growing exponentially, this ceiling will quickly be breached at some nodes and the tree can grow no further! The choice of $n$ is thus capped even if infinite resources are available. We next derive the said upper bound.

Inequalities (6) imply

$$\begin{aligned}
\frac{|\,(h_t^2/2) - r\,|}{2\eta\gamma\sqrt{n}} &\leq \frac{h_t^2}{2\eta^2\gamma^2}, \\
\frac{h_t^2}{2\eta^2\gamma^2} &\leq \frac{1}{2}.
\end{aligned}$$

Hence

$$h_t^2 \leq (\eta\gamma)^2 \leq \left[ \frac{h_t^2\sqrt{n}}{|\,(h_t^2/2) - r\,|} \right]^2,$$

which can be simplified to yield

$$\left[\,(h_t^2/2) - r\,\right]^2 \leq n h_t^2.$$

Finally, the above quadratic inequality (in $h_t^2$) is equivalent to

$$2(r+n) - 2\sqrt{2rn + n^2} \leq h_t^2 \leq 2(r+n) + 2\sqrt{2rn + n^2}\,.$$

We conclude that

$$h_t^2 \leq 2(r+n) + 2\sqrt{2rn + n^2} \tag{11}$$

is necessary for the existence of valid branching probabilities. This condition does not depend on the choice of $\gamma$ because the identity $\gamma = h_0$ did not enter the analysis.

This result may sound puzzling at first. Under the Black-Scholes model, valid branching probabilities always exist if $n$ is large enough. Why, one may ask, can't the same property hold here? The answer lies in the volatility process. The daily volatility in the Black-Scholes model is a constant, which amounts to setting $h_t$ to some fixed number. So every state solves the same Eqs. (3)–(5) for the probabilities, and increasing $n$ will eventually have inequality (11) satisfied for all states. In contrast, the volatility under GARCH fluctuates. So each state $(y_t, h_t^2)$ faces different Eqs. (3)–(5) in solving for the probabilities. Increasing $n$ makes inequality (11) harder to satisfy for those states with a large $h_t^2$, whose existence has been confirmed earlier for GARCH.

### 4.1.3 Numerical Evidence of Explosion and Shallowness

The following parameters from Ritchken and Trevor (1999) and Cakici and Topyan (2000) will be assumed throughout the section unless stated otherwise: $S_0 = 100$, $y_0 = \ln S_0 = 4.60517$, $r = 0$, $h_0^2 = 0.0001096$, $\gamma = 0.010469$, $\beta_0 = 0.000006575$, $\beta_1 = 0.9$, $\beta_2 = 0.04$, and $c = 0$. As $r = c = 0$, combinatorial explosion occurs when

$$n > \frac{1-\beta_1}{\beta_2} = \frac{1-0.9}{0.04} = 2.5.$$

Figure 9 picks $n = 3, 4, 5$ to demonstrate the exponential growth of the RTCT tree. The rate of growth clearly increases with $n$. For comparison, the standard trinomial tree contains only $2t + 1$ nodes at date $t$.

The number of nodes is critical because the running time is proportional to it. We mention earlier that there may be nodes which are not reachable (recall Fig. 6). In theory, if such nodes are numerous, the algorithm can potentially run more efficiently by skipping them. Figure 10 shows that the proportion of unreachable nodes is small for $n = 3, 4, 5$. We will see shortly that the same conclusion also holds for larger $n$'s. As the overwhelming majority of nodes are reachable, clever programming techniques to skip unreachable nodes will bring no substantial benefits.

Now suppose we pick $n = 100$ to seek very high accuracy at the expense of efficiency. The theory predicts that the RTCT tree's final maturity will be cut short. Indeed, with $r = 0$, inequality (11) imposes the universal upper bound $h_t^2 \leq 4n = 400$. This means that a node with $h_t > 20$ cannot have valid branching probabilities and thus cannot grow further. As this ceiling is breached somewhere at date 9 because of the exponential growth of the largest value of $h_t$, the tree stops growing then. Table 1 lists the final dates under various $n$'s exceeding the threshold of explosion. Observe that the tree's final maturity decreases rapidly as $n$ increases. For example, it is 72 days when $n = 5$, 34 days when $n = 10$, and 12 days when $n = 50$. To be useful, $n$ cannot be so large as to make the tree's final maturity fall short of the derivative's. It is therefore important not to pick too large an $n$ for only trees of very short maturities will be generated otherwise. Table 1 also tabulates the total numbers of nodes and unreachable ones among them. Again, the overwhelming majority of the nodes are occupied as claimed earlier.

Some of the calculated option prices in Ritchken and Trevor (1999) use $n$ as large as 25 and option maturity dates as far as 200. These choices contradict our analysis and data: Table 1 says that the RTCT tree with $n = 25$ stops growing as early as date 18. Therefore, their prices must be viewed with caution.

Cakici and Topyan (2000) use $n = 1$ throughout their paper. Explosion is avoided for their choice of parameters, which are also used in this section. Besides, Cakici and Topyan (2000) suggest that accurate results are obtainable with $n = 1$. As this choice of $n$ typically does not lead to explosion, this claim, if true, would imply that the GARCH option pricing with RTCT algorithm is likely to be efficient and accurate in practice after all. Unfortunately, the RTCT algorithm can be inaccurate with $n = 1$; furthermore, explosion can still happen with the smallest choice of $n$. To demonstrate the first point, the option prices in Table 2 are based on parameters from Cakici and Topyan (2000) except that $\beta_0 = 0.000007$ instead of 0.000006575. One can see that the RTCT algorithm deviates from the correct option price for $K \leq 200$. Second, explosion can still happen with $n = 1$: Suppose we use the same parameters as Fig. 9 except that $c = 2$ and $n = 1$. Then the RTCT tree explodes as plotted in Fig. 11. Interestingly the tree's final maturity date is only 54 days!

## 4.2 Failure to Converge with Increasing $n$

Does the RTCT algorithm converge with increasing $n$? Table 3 and Figure 12 show that there is a downward trend in the calculated option prices except when $N = 2$. Moreover, the downward trend accelerates as $n$ increases. To rule out the possibility that the problem originates from the tree approximation of the continuous-state model, Monte Carlo simulation on the RTCT tree model (7) is carried out. The data in Table 3 show that the discrete-state tree model produces Monte Carlo estimates generally consistent with those of the continuous-state model. Therefore, the potential factors causing the RTCT algorithm to deviate numerically are the volatility interpolation scheme, the value of $K$, and the use of interpolation in backward induction. We will see that the prime reason is the volatility interpolation scheme.

# 5 The Mean-Tracking Tree

The RTCT tree has several weaknesses. First, it explodes exponentially when $n$ crosses a threshold. Second, it is not known whether the RTCT tree can escape explosion as long as $n$ does not exceed some threshold. Third, when explosion happens, the tree's maturity is cut short, making it unable to price derivatives with a longer maturity. Fourth, option prices may fail to converge as $n$ increases. We next turn to the MT algorithm that addresses these problems. The MT tree makes two changes to the RT tree. The first is to adopt the log-linear interpolation scheme. This addresses the convergence problem mentioned earlier. The second is to let the middle branch of the multinomial tree track the mean of $y_{t+1}$. This addresses the explosion problem and its consequence of shortened maturity.

## 5.1 Volatility Interpolation

The distribution of the volatilities reaching a node plays a key role in pricing accuracy. The RTCT algorithm essentially assumes that the distribution is uniform: Interpolated squared volatilities are equally spaced between the minimum and the maximum ones. Figure 13, however, shows that the actual distribution is closer to a lognormal distribution than a uniform one. It strongly suggests that there should be more interpolated volatilities at the lower end than at the higher end. This is the rationale for the MT algorithm's adopting the log-linear interpolation scheme, in which the logarithmic volatilities are equally spaced.

## 5.2 Tree Building

At date $t$, let node A be the node closest to the mean of $y_{t+1}$ given $(y_t, h_t^2)$, i.e., $y_t + r - (h_t^2/2)$. For convenience, we use $\mu$ to denote this conditional mean minus the current logarithmic price:

$$\mu \equiv r - \frac{h_t^2}{2}$$

(see Fig. 14). By the geometry of the tree, node A's logarithmic price equals $y_t + a\gamma_n$ for some integer $a$. The criterion by which node A is chosen makes sure that

$$| \, a\gamma_n - \mu \, | \leq \frac{\gamma_n}{2}. \tag{12}$$

To create the multinomial tree, make the middle branch of the $(2n+1)$-nomial tree line up with node A (see Fig. 15). Although a node reaches only $2n+1$ nodes after one day, the top and bottom nodes span over

$$2n\eta + 1 \tag{13}$$

nodes. The probabilities for the upward, middle, and downward branches are set to

$$\begin{aligned} p_u &= \frac{nh_t^2 + (a\gamma_n - \mu)^2}{2n^2\eta^2\gamma_n^2} - \frac{a\gamma_n - \mu}{2n\eta\gamma_n}, \\ p_m &= 1 - \frac{nh_t^2 + (a\gamma_n - \mu)^2}{n^2\eta^2\gamma_n^2}, \\ p_d &= \frac{nh_t^2 + (a\gamma_n - \mu)^2}{2n^2\eta^2\gamma_n^2} + \frac{a\gamma_n - \mu}{2n\eta\gamma_n}. \end{aligned}$$

They match the conditional mean and variance of the GARCH process at date $t+1$ exactly; hence convergence is guaranteed. State $(y_t, h_t^2)$ at date $t$ is followed by state $(y_t + \ell\eta\gamma_n, h_{t+1}^2)$ at date $t+1$, where

$$\begin{aligned} h_{t+1}^2 &= \beta_0 + \beta_1 h_t^2 + \beta_2 h_t^2 (\epsilon''_{t+1} - c)^2, \\ \epsilon''_{t+1} &= \frac{\ell\eta\gamma_n + a\gamma_n - (r - h_t^2/2)}{h_t}, \\ \ell &= 0, \pm 1, \pm 2, \dots, \pm n. \end{aligned} \tag{14}$$

From the underlying trinomial model, this transition occurs with probability

$$\sum_{j_u, j_m, j_d} \frac{n!}{j_u! \, j_m! \, j_d!} \, p_u^{j_u} p_m^{j_m} p_d^{j_d},$$

where $j_u, j_m, j_d \geq 0$, $n = j_u + j_m + j_d$, and $\ell = j_u - j_d$.

The conditions for the probabilities to lie within 0 and 1, i.e., $0 \leq p_u, p_m, p_d \leq 1$, are

$$\begin{aligned}
\frac{|\, a\gamma_n - \mu \,|}{2n\eta\gamma_n} &\leq \frac{nh_t^2 + (a\gamma_n - \mu)^2}{2n^2\eta^2\gamma_n^2}, &(15)\\
\frac{nh_t^2 + (a\gamma_n - \mu)^2}{2n^2\eta^2\gamma_n^2} &\leq \frac{1}{2}, &(16)\\
\frac{nh_t^2 + (a\gamma_n - \mu)^2}{2n^2\eta^2\gamma_n^2} &\leq 1 - \frac{|\, a\gamma_n - \mu \,|}{2n\eta\gamma_n}. &(17)
\end{aligned}$$

Inequalities (15)–(16) are equivalent to

$$\frac{\sqrt{nh_t^2 + (a\gamma_n - \mu)^2}}{n\gamma_n} \leq \eta \leq \frac{nh_t^2 + (a\gamma_n - \mu)^2}{n\gamma_n |\, a\gamma_n - \mu \,|}. \tag{18}$$

Inequalities (12) and (16) together imply inequality (17) because

$$\frac{nh_t^2 + (a\gamma_n - \mu)^2}{2n^2\eta^2\gamma_n^2} + \frac{|\, a\gamma_n - \mu \,|}{2n\eta\gamma_n} \leq \frac{1}{2} + \frac{1}{4n\eta} \leq 1.$$

Hence the probabilities are valid if and only if the much simpler inequalities (18) hold.

Does interval (18) contain a positive integer for $\eta$ to take its value on? It does if certain conditions are met. Let $H_{\min}^2 \equiv \min(h_0^2, \beta_0/(1-\beta_1))$ to make $H_{\min}^2 \leq h_t^2$ for $t \geq 0$ (see Appendix A for proof[1]). The sought-after condition is

$$\gamma_n^2 \leq H_{\min}^2 \tag{19}$$

as proved in Appendix B.

The existence of a valid $\eta$ guaranteed, the MT tree never stops growing beyond a certain maturity. Hence it solves the short-maturity problem of the RTCT tree. Rather than searching for an $\eta$ to satisfy inequalities (18), the MT tree simply takes

$$\eta = \left\lceil \frac{\sqrt{nh_t^2 + (a\gamma_n - \mu)^2}}{n\gamma_n} \right\rceil \tag{20}$$

based on inequalities (18). Although other choices are clearly possible, this particular choice is amenable to later analysis on the size of the MT tree.

[1] All the appendices are not available for public; please contact the author if interested (in private via E-mail: r90723065@ms90.ntu.edu.tw).

Finally, we discuss the choice of $\gamma$, hence $\gamma_n$ as well because $\gamma_n = \gamma/\sqrt{n}$. If $\gamma \leq H_{\min}$, then $\gamma_n$ satisfies inequality (19) for all $n$. We compare two reasonable choices:

**(a)** $\gamma = H_{\min}$, and

**(b)** $\gamma = H_{\min}/2$.

A smaller $\gamma$ generally leads to larger trees, hence longer running times. On the other hand, a smaller $\gamma$ results in better accuracy. Figure 16 demonstrates that, with choice (b), the option prices fall within the 95% confidence interval of Monte Carlo simulation of the continuous-state model (2) for all $n > 1$. Hence choice (b) is better in terms of overall accuracy and convergence speed. With the choice of

$$\gamma_n = \frac{H_{\min}}{2\sqrt{n}}, \tag{21}$$

all the parameters of the MT tree have been chosen.

## 5.3 Sufficient Condition for Nonexplosion

In practice, it may be critical to know a choice of $n$ will not result in an exploding tree before tree building is attempted. Without this knowledge, tree building may take a long time if the tree explodes and may even end up with a tree not meeting the required maturity if shortened maturity is a problem. In the case of the MT tree, the criterion for nonexplosion is a very simple one: The MT tree does not explode if

$$n \leq \left( \sqrt{\frac{1-\beta_1}{\beta_2}} - c \right)^2. \tag{22}$$

Furthermore, when the above relation holds, the tree size is only quadratic in maturity, the same as the CRR tree. (See Appendix C for proof.) The MT tree is thus the first tree-based GARCH option pricing algorithm that is provably efficient.

To check that this conclusion makes intuitive sense, observe that a trinomial tree under the Black-Scholes model obtains by letting $\beta_1 = 0$ and $\beta_2 \to 0$. In the limit, relation (22) holds for any $n$, and the tree has quadratic size by our claim. This node count agrees with the well-known fact about the trinomial tree's quadratic size (recall Fig. 2). The MT tree's size is hence asymptotically optimal.

Earlier in section 4.1.1, we show that the RTCT tree size explodes if $n$ exceeds some threshold; in fact, the threshold is $(1-\beta_1)/\beta_2$ when $c=0$. But nothing could be concluded about the node count when that threshold is not breached. The positive result (22) of the MT tree therefore fills the void because it says that the MT tree is efficient if $n$ does *not* exceed some threshold. Surprisingly, the sufficient condition for nonexplosion reduces to $n \leq (1-\beta_1)/\beta_2$ when $c=0$. As this is the threshold of explosion for the RTCT tree, our threshold is in some sense tight.

# 6 Evaluation of the MT Tree

Table 2 shows that the CT algorithm can be inaccurate in options pricing with $n = 1$. Table 4 expands on Table 2 by adding cases $n = 2, 3, 4$. It is clear that in every combination of $n$ and $K$, the CT algorithm deviates from the simulation results. On the other hand, the MT algorithm produces prices within the 95% confidence interval for $K \geq 2$ with $n = 1$, $K \geq 10$ with $n = 2$, and $K \geq 50$ with $n = 3, 4$. Hence the MT algorithm can succeed where the RTCT algorithm cannot. Observe that $K$ needs to be increased for a larger $n$ because the resulting increase in the number of volatilities per node demands more resolution. This phenomenon agrees with the theoretical findings of Dai et al. (2002) in Asian option pricing, where it is shown that $K$ should grow at least with $\sqrt{n}$ to guarantee convergence.

Next we benchmark the MT algorithm's performance with increasing $n$. For this purpose, we duplicate the settings in Table 3 for the RTCT algorithm to produce Table 5 for the MT algorithm. Select prices are plotted in Fig. 17 for illustration. Unlike the RTCT algorithm, all prices generated by the MT algorithm are within the 95% confidence interval of Monte Carlo simulation on the tree model (14). Obviously, the MT algorithm is better than the RTCT algorithm not only in terms of accuracy but also in terms of convergence speed. It provides results very close to the true option price even with a small $n$.

All the numerical experiments up to now assume $r = c = 0$. Table 6 investigates MT algorithm's accuracy for options with various strike prices under a GARCH model with nonzero $r$ and $c$: $r = 5\%$ (annual) and $c = 0.5$. Although a few of the computed option prices are outside the 95% confidence interval, they are nontheless quite close to the Monte Carlo estimates. Furthermore, they are as good as the best computed prices in Table 3 of Duan and Simonato (2001).

Relation (22) is a sufficient condition for the MT tree to be efficient in size. In this case, the number of tree nodes grows linearly with time; thus the total node count becomes quadratic in maturity. We use two concrete cases to check the theoretical result. The first setting adopts $\beta_1 = 0.8$, $\beta_2 = 0.1$, and $c = 0$. The tree should not explode because

$$n = 1 < \left( \sqrt{\frac{1 - 0.8}{0.1}} - 0 \right)^2 = 2.$$

Indeed, the tree size grows linearly with time as shown in Fig. 18. The total number of nodes is therefore quadratic in maturity in complete agreement with the theoretical analysis. Take another setting with $\beta_1 = 0.8$, $\beta_2 = 0.1$, and $c = 0.9$. Then relation (22) is violated because

$$n = 1 > \left( \sqrt{\frac{1 - 0.8}{0.1}} - 0.9 \right)^2 = 0.264416.$$

The tree also turns out to grow exponentially as shown in Fig. 19. Unlike the RTCT tree, an exploding MT tree is still useful because it will not be cut short.

Because relation (22) is not a necessary condition for the MT tree not to explode, its violation does not necessarily imply combinatorial explosion. Take the parameters $\beta_1 = 0.8$, $\beta_2 = 0.1$, and $c = 0.5$ in Table 6 for example. The criterion for nonexplosion is violated for all $n \geq 1$. But in fact, the case $n = 1$ does not result in an exponential size even though the tree size is more than quadratic. Exponential explosion sets in for $n > 1$. This is why the table does not compute prices for $n > 1$ when the maturity of the option exceeds 30. They simply take too much time.

## 7 Conclusions

GARCH option pricing is difficult because of the GARCH model's bivariate and path-dependent nature. By modifying the well-known Ritchken-Trevor-Cakici-Topyan (RTCT) tree, we have come up with a simple tree, the mean-tracking (MT) tree, that is both accurate and provably efficient when $n$ does not exceed a simple threshold. Specifically, its tree size is quadratic in maturity if $n$ does not exceed the threshold. This is the first tree-based GARCH option pricing algorithm that provably does not explode if certain conditions are met. Both the threshold and the quadratic size are tight. The MT tree does not suffer from the short-maturity problem of the RTCT tree. We conclude that the MT tree is a provably efficient tree for derivatives pricing under the GARCH option pricing model. All our theoretical results are proved and then backed up by extensive numerical experiments.

# References

[1] Bollerslev, T. (1986) Generalized Autoregressive Conditional Heteroskedasticity. *Journal of Econometrics*, 31, pp. 307–327.

[2] Bollerslev, T., Chou, R.Y., and Kroner, K. (1992) ARCH Modeling in Finance: A Review of the Theory and Empirical Evidence. *Journal of Econometrics*, 52, pp. 5–59.

[3] Cakici, N., and Topyan, K. (2000) The GARCH Option Pricing Model: A Lattice Approach. *Journal of Computational Finance*, 3(4), pp. 71–85.

[4] Chien, H.-H. *On the Complexity of the Ritchken-Sankarasubramanian Interest Rate Model*. MBA Thesis. Department of Finance, National Taiwan University, Taiwan, 2003.

[5] Cox, J.C., Ingersoll, J.E., and Ross, S.A. (1985) A Theory of the Term Structure of Interest Rates. *Econometrica*, 53(2), pp. 385–407.

[6] Cox, J.C., Ross, S.A., and Rubinstein, M. (1979) Option Pricing: a Simplified Approach. *Journal of Financial Economics*, 7(3), pp. 229–263.

[7] Dai, T.-S., Huang, G.-S., and Lyuu, Y.-D. (2002) Extremely Accurate and efficient Algorithms for European-Style Asian Options with Range Bounds. In *Proc. 2002 NTU International Conference on Finance*, National Taiwan University, Taiwan.

[8] Dai, T.-S., and Lyuu, Y.-D. (2003) A Subexponential-Time Asian Option Pricing Algorithm. Manuscript, June 2003.

[9] Duan, J.-C. (1995) The GARCH Option Pricing Model. *Mathematical Finance*, 5(1), pp. 13–32.

[10] Duan, J.-C., Gauthier, G., Sasseville, C., and Simonato, J.-G. (2002) Seize the Moments: Approximating American Option Prices in the GARCH Framework. Manuscript, 9 July 2002. To appear in *Journal of Futures Markets*.

[11] Duan, J.-C., and Simonato, J.-G. (2001) American Option Pricing under GARCH by a Markov Chain Approximation. *Journal of Economic Dynamics & Control*, 22, pp. 1689–1718.

[12] Engle, R., and Ng, V. (1993) Measuring and Testing of the Impact of News on Volatility. *Journal of Finance*, 48, pp. 1749–1778.

[13] Heston, S.L., and Nandi, S. (2000) A Closed-Form GARCH Option Valuation Model. *Review of Financial Studies*, 13(3), pp. 585–625.

[14] Hull, J.C., and White, A. (1993) Efficient Procedures for Valuing European and American Path-Dependent Options. *The Journal of Derivatives*, 1(1), pp. 21–31.

[15] Hull, J.C., and White, A. (1993) One-Factor Interest-Rate Models and the Valuation of Interest-Rate Derivative Securities. *Journal of Financial and Quantitative Analysis*, 28(2), pp. 235–254.

[16] Li, A., Ritchken, P., and Sankarasubramanian, L. (1995) Lattice Models for Pricing American Interest Rate Claims. *The Journal of Finance*, 50(2), pp. 719–737.

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

[18] Lyuu, Y.-D., and Wu, C.-N. (2003) Complexity of the Ritchken-Trevor-Cakici-Topyan GARCH Option Pricing Algorithm. In *Proceedings of IASTED International Conference on Financial Engineering and Applications (FEA 2003)*, Banff, Canada, July 2–4, 2003.

[19] NELSON, D.B., AND RAMASWAMY, K. (1990) Simple Binomial Processes as Diffusion Approximations in Financial Models. *Review of Financial Studies*, 3(3), pp. 393–430.

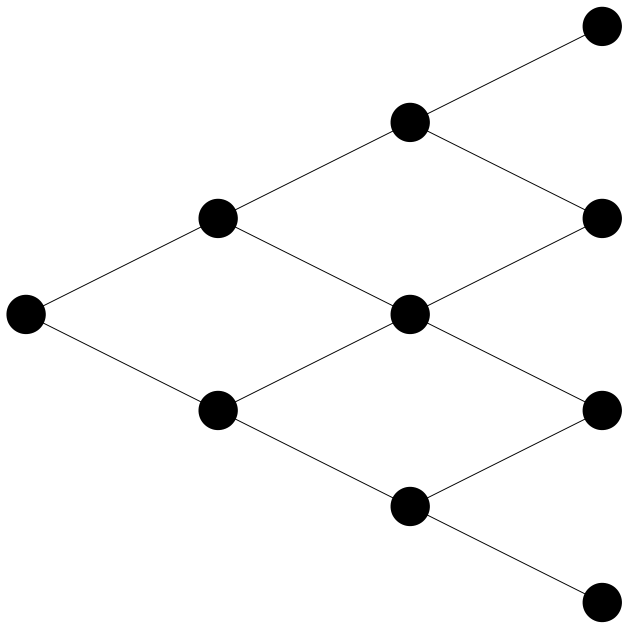
[20] RITCHKEN, P., AND SANKARASUBRAMANIAN, L. (1995) Volatility Structures of Forward Rates and the Dynamics of the Term Structure. *Mathematical Finance*, 5(1), pp. 55–72.

[21] RITCHKEN, P., SANKARASUBRAMANIAN, L., AND VIJH, A.M. (1993) The Valuation of Path Dependent Contracts on the Average." *Management Science*, 39(10), pp. 1202–1213.

[22] RITCHKEN, P., AND TREVOR, R. (1999) Pricing Options under Generalized GARCH and Stochastic Volatility Processes. *Journal of Finance*, 54(1), pp. 377–402.

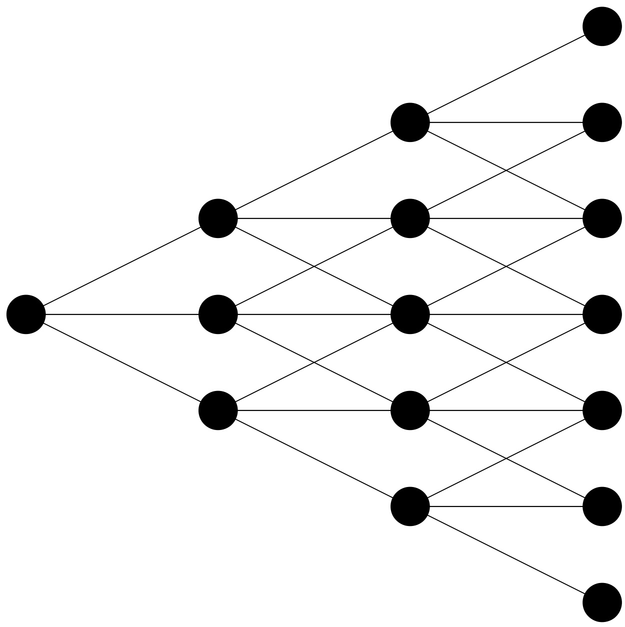
[23] TAYLOR, S. (1986) *Modelling Financial Time Series*. New York: Wiley.

Figure 1: **Binomial tree.**



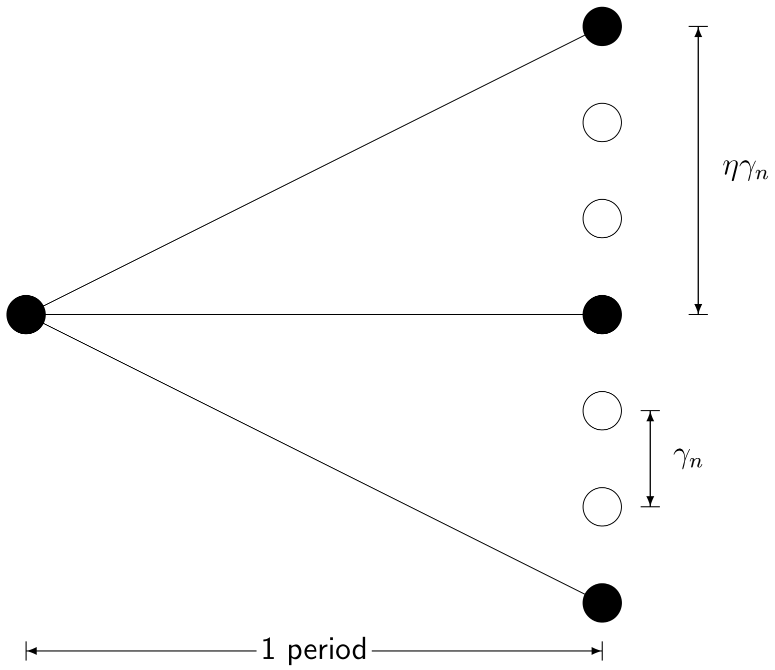
Each node has two successor nodes. The number of nodes at any time $t$ is $t+1$, a linear growth. The total number of nodes of an $m$-period binomial tree is thus $(m+2)(m+1)/2$, a quadratic growth in maturity $m$.

Figure 2: **Trinomial tree.**



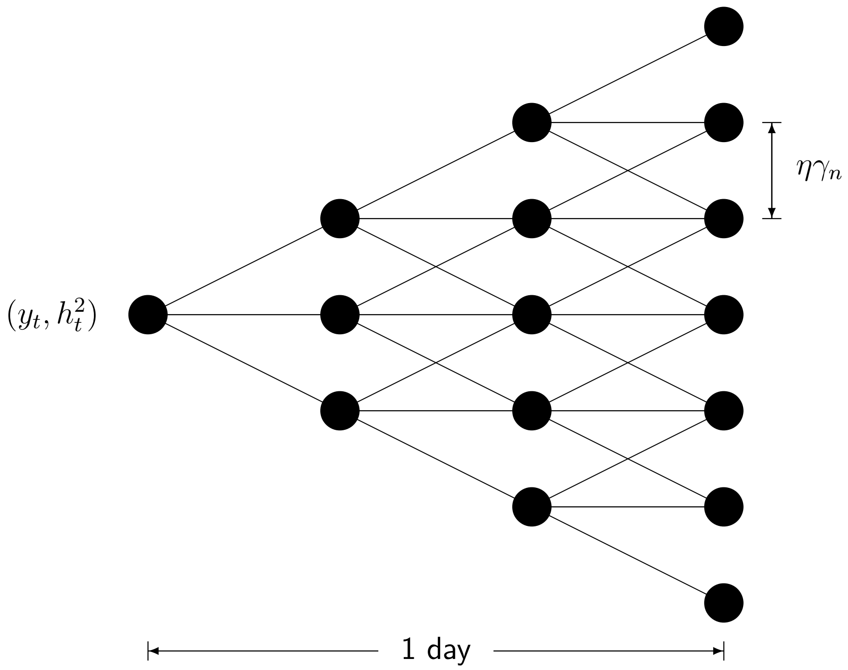
Each node has three successor nodes. The number of nodes at any time $t$ is $2t+1$, a linear growth. The total number of nodes of an $m$-period trinomial tree is thus $(m+1)^2$, a quadratic growth in maturity $m$. Note that the trinomial tree recombines without exploding.

Figure 3: **Trinomial tree, jump parameter $\eta$, and jump size $\eta\gamma_n$.**



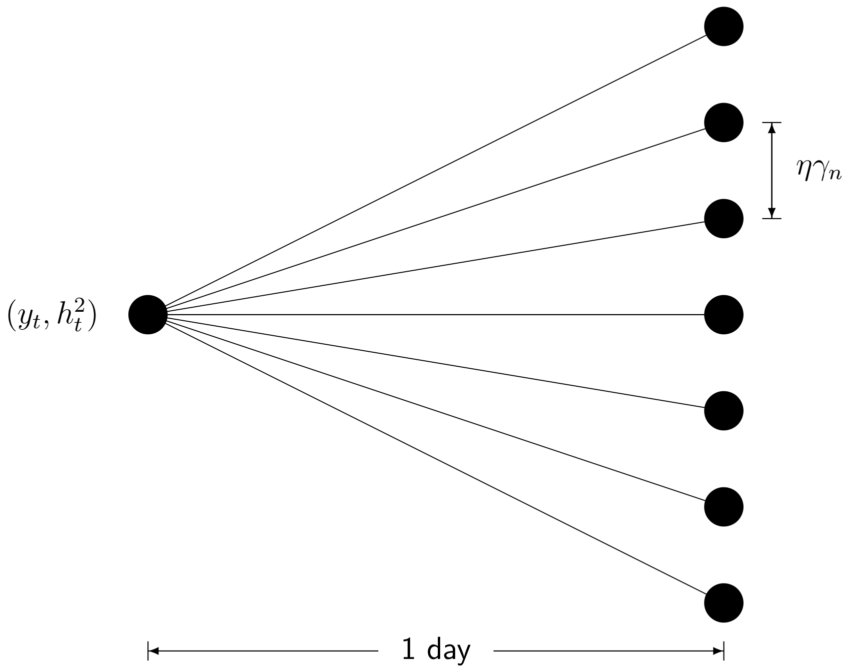

Adjacent nodes at the same time are spaced by $\gamma_n$. The two outer branches fan out around the middle branch to reach nodes that are $\eta$ nodes away from the center. Hollow nodes are not reached from the node on the left. Here $\eta = 3$.

Figure 4: **RTCT trinomial tree for logarithmic price.**



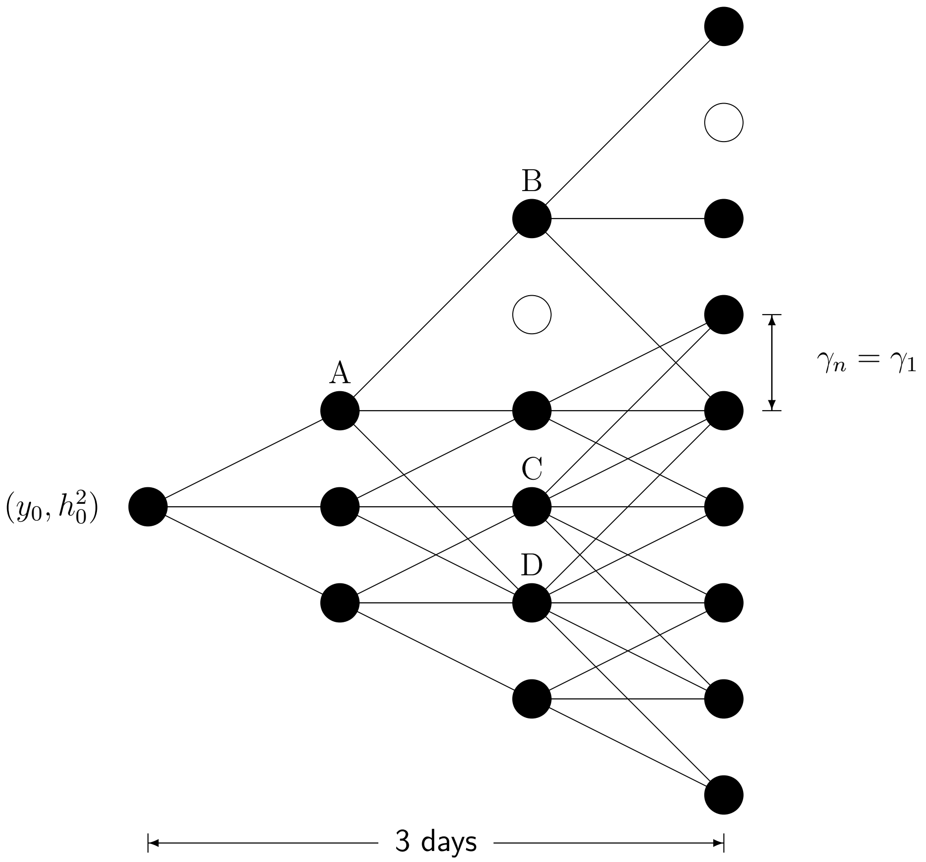

A day is partitioned into $n = 3$ periods, and the jump size is $\eta\gamma_n$. The 7 values on the right should approximate the distribution of $y_{t+1}$ given $(y_t, h_t^2)$. Recall from Fig. 3 that there are $\eta - 1$ nodes, which are not drawn, between any two adjacent nodes at the same time.

Figure 5: **RTCT multinomial tree for logarithmic price.**



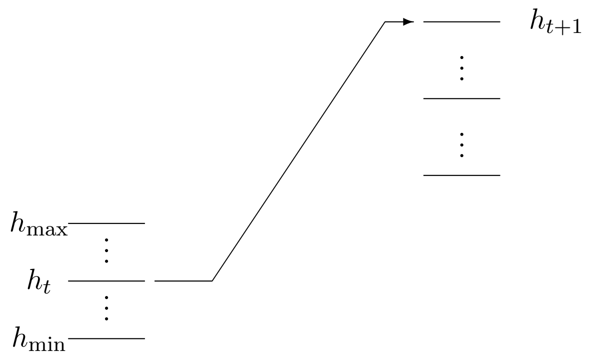

This heptanomial tree is the outcome of the trinomial tree in Fig. 4 after its intraday nodes are removed. Recall that $n = 3$. In general, we infer from Fig. 3 that there are $1 + 2n\eta$ nodes at date $t + 1$ between the top and bottom nodes (inclusive), only $2n + 1$ of which are reachable from $(y_t, h_t^2)$ and are drawn above.

Figure 6: **Possible geometry of a 3-day RTCT tree.**



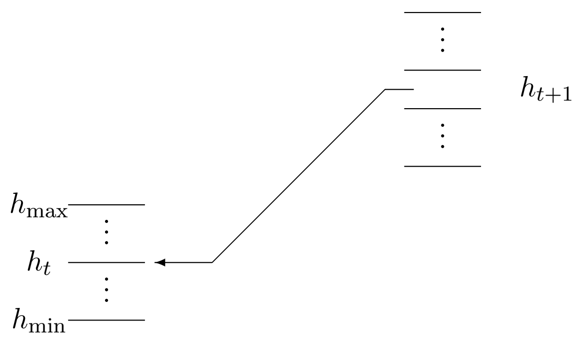

A day is partitioned into $n = 1$ period. Nodes A and B have a jump size of $2\gamma_1$. Nodes C with two $h_t^2$'s and D with three $h_t^2$'s have two jump sizes: $\gamma_1$ and $2\gamma_1$. All other nodes have a jump size of $\gamma_1$. Nodes that are not reachable are shown as hollow nodes.

Figure 7: **Case where maximum volatility follows an interpolated volatility.**



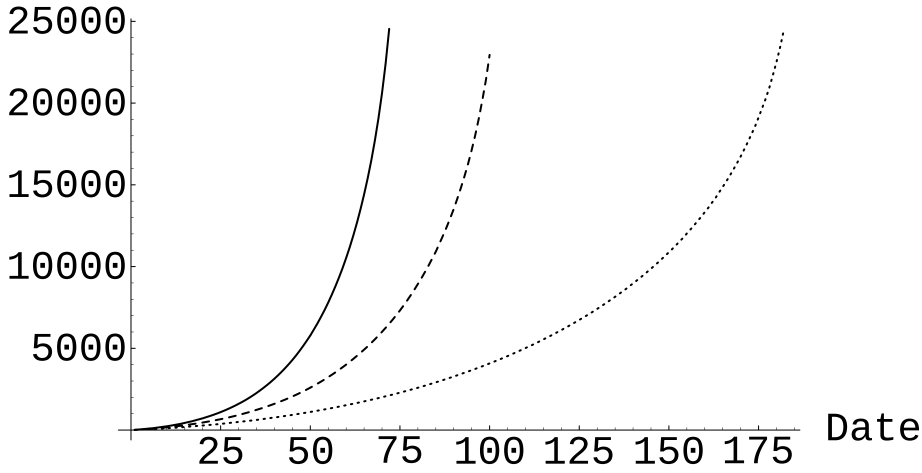

Maximum volatility $h_{t+1}$ at the node on the right follows interpolated volatility $h_t$ by the updating rule. Hence $h_{t+1}$ is an artifact.

Figure 8: **Backward induction on the tree.**



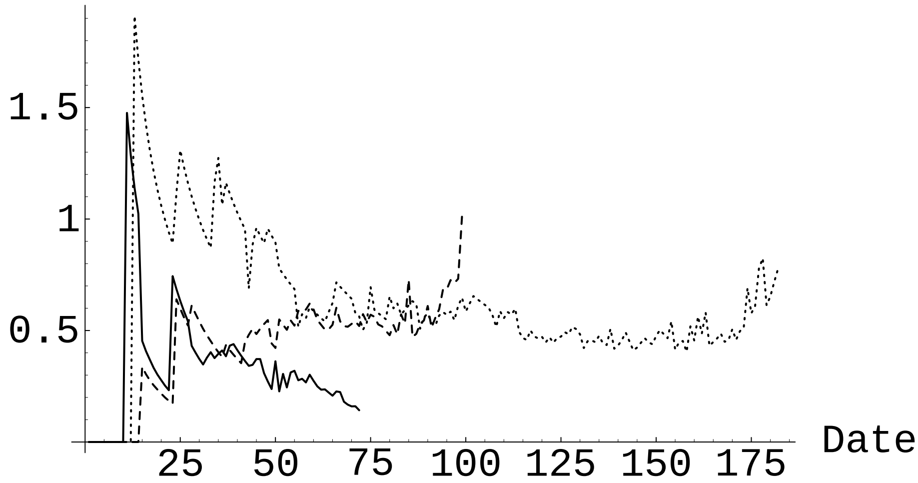

Volatility $h_{t+1}$ follows $h_t$ by the updating rule. Because it does not match any interpolated volatility, its corresponding option value is found by interpolating from the two option values whose volatilities bracket $h_{t+1}$.

Figure 9: **Exponential growth of the RTCT tree.**




The parameters are $S_0 = 100$, $y_0 = \ln S_0 = 4.60517$, $r = 0$, $h_0^2 = 0.0001096$, $\gamma = 0.010469$, $\beta_0 = 0.000006575$, $\beta_1 = 0.9$, $\beta_2 = 0.04$, and $c = 0$. The dotted line is based on $n = 3$, the dashed line on $n = 4$, and the solid line on $n = 5$. The standard trinomial tree, in contrast, has only $2t + 1$ nodes at date $t$.

Figure 10: **The percent of unreachable nodes.**




The parameters are $S_0 = 100$, $y_0 = \ln S_0 = 4.60517$, $r = 0$, $h_0^2 = 0.0001096$, $\gamma = 0.010469$, $\beta_0 = 0.000006575$, $\beta_1 = 0.9$, $\beta_2 = 0.04$, and $c = 0$. The plots show the percent of unreachable nodes among all nodes at each date. The dotted line is based on $n = 3$, the dashed line on $n = 4$, and the solid line on $n = 5$. The number of unreachable nodes is insignificant in all 3 lines.

Table 1: **Final maturity dates and sizes of exploding trees.**

| $n$ | Final date ($t$) | Total number of nodes | Total number of unreachable nodes |
|---|---|---|---|
| 3 | 182 | 1,017,327 | 5,565 |
| 4 | 100 | 499,205 | 3,028 |
| 5 | 72 | 368,523 | 947 |
| 10 | 34 | 222,935 | 42 |
| 25 | 18 | 286,844 | 6,925 |
| 50 | 12 | 305,113 | 448 |
| 100 | 9 | 578,710 | 3,961 |
| 150 | 8 | 795,309 | 2,011 |
| 200 | 7 | 652,808 | 1,596 |
| 250 | 7 | 1,747,758 | 20,291 |
| 300 | 7 | 2,929,508 | 11,510 |
| 350 | 6 | 1,179,157 | 3,151 |

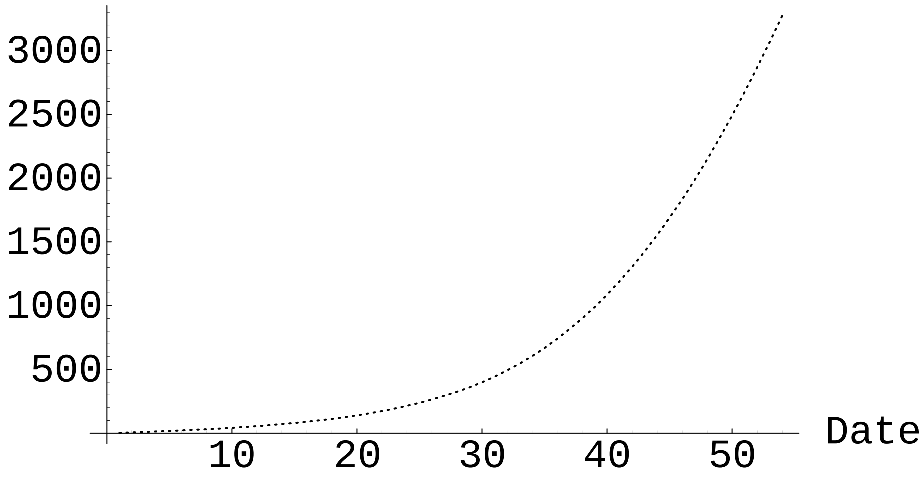
The parameters are $S_0 = 100$, $y_0 = \ln S_0 = 4.60517$, $r = 0$, $h_0^2 = 0.0001096$, $\gamma = 0.010469$, $\beta_0 = 0.000006575$, $\beta_1 = 0.9$, $\beta_2 = 0.04$, and $c = 0$. With $n > 2.5$, all RTCT trees in the table explode. The final maturity date of the tree shortens quickly as $n$ increases. The total number of nodes in each tree far exceeds the $(t+1)^2$ of the standard trinomial tree. The overwhelming majority of nodes are reachable in all trees.

Table 2: **Case where CT fails.**

| $K$ | 2 | 10 | 20 | 50 | 100 | 200 |
|---|---|---|---|---|---|---|
| Option price | 4.2301 | 4.2365 | 4.2267 | 4.2274 | 4.2265 | 4.2268 |
| ${}^{\infty}L$ | 4.2714 | | | | | |
| ${}^{\infty}U$ | 4.3087 | | | | | |

The parameters are $S_0 = 100$, $r = 0$, $h_0^2 = 0.0001096$, $\gamma = h_0 = 0.010469$, $\beta_0 = 0.000007$, $\beta_1 = 0.9$, $\beta_2 = 0.04$, $n = 1$, and $c = 0$. $K$ denotes the number of interpolated volatilities per node used in backward induction. The option is a European call with a strike price of 100 and a maturity of 100 days. ${}^{\infty}L$ and ${}^{\infty}U$ form the 95% confidence interval for the true option price based on Monte Carlo simulation on the continuous-state model (2) with 500,000 paths.

Figure 11: **Exponential growth of the Cakici-Topyan tree when $n = 1$.**




The parameters are $S_0 = 100$, $r = 0$, $h_0^2 = 0.0001096$, $\gamma = 0.010469$, $\beta_0 = 0.000006575$, $\beta_1 = 0.9$, $\beta_2 = 0.04$, $c = 2$, and $n = 1$. Clearly, the Cakici-Topyan tree explodes. The final maturity date of the tree is 54 days.

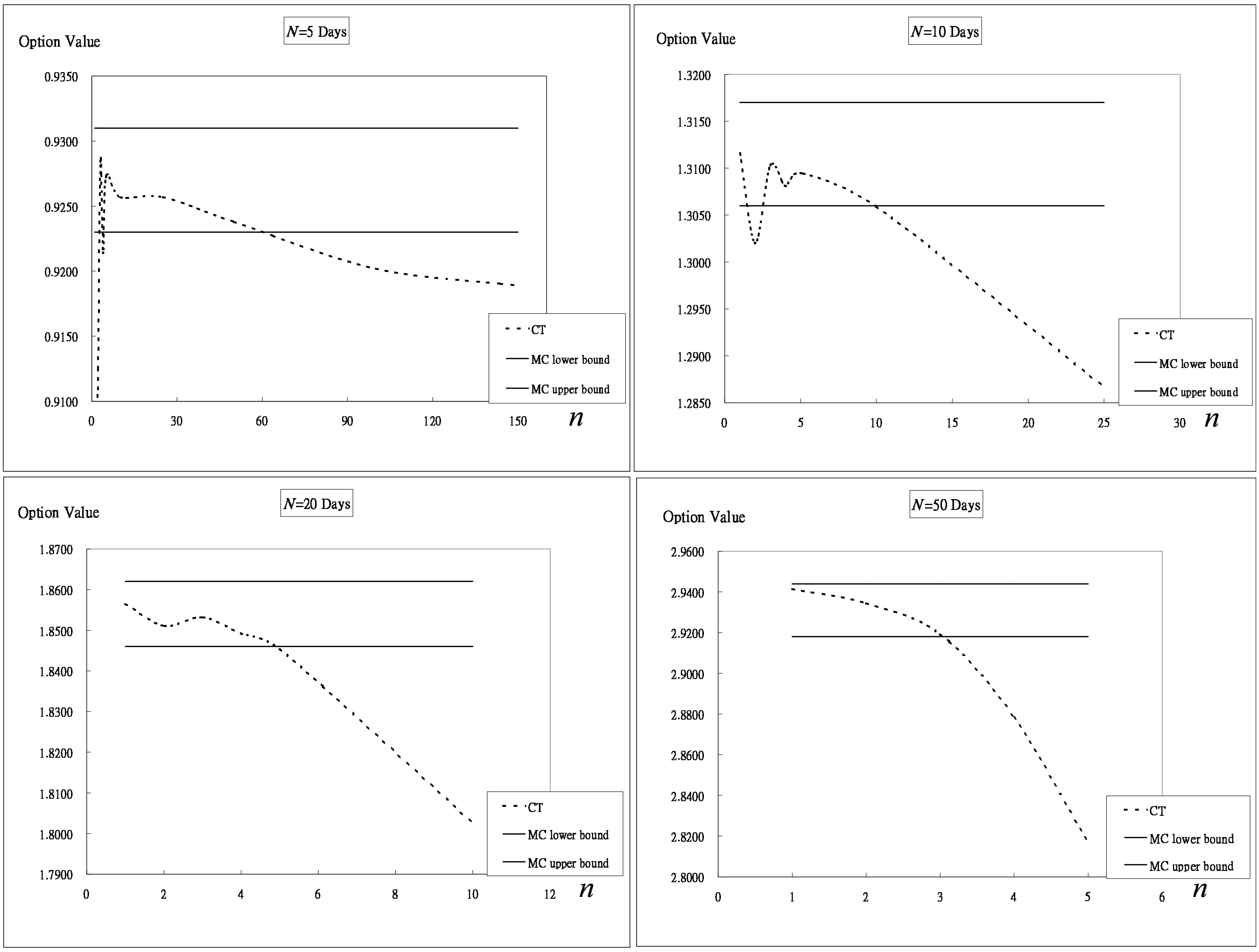
Table 3: **Cases where RTCT fails to converge.**

| | Maturity of Option (Days) | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 2 | | | 5 | | | 10 | | | 20 | | |
| $n$ | CT | $^{\infty}DU$ | $^{\infty}DL$ | CT | $^{\infty}DU$ | $^{\infty}DL$ | CT | $^{\infty}DU$ | $^{\infty}DL$ | CT | $^{\infty}DU$ | $^{\infty}DL$ |
| 1 | 0.5888 | 0.5903 | 0.5856 | 0.9093* | 0.9023 | 0.8945 | 1.3116* | 1.3400 | 1.3291 | 1.8565* | 1.8765 | 1.8609 |
| 2 | 0.5674 | 0.5701 | 0.5652 | 0.9091 | 0.9132 | 0.9054 | 1.3020 | 1.3126 | 1.3016 | 1.8511 | 1.8604 | 1.8447 |
| 3 | 0.5736 | 0.5746 | 0.5697 | 0.9284 | 0.9319 | 0.9242 | 1.3103 | 1.3190 | 1.3080 | 1.8532 | 1.8604 | 1.8448 |
| 4 | 0.5742 | 0.5758 | 0.5709 | 0.9214 | 0.9289 | 0.9211 | 1.3081 | 1.3115 | 1.3006 | 1.8492 | 1.8540 | 1.8384 |
| 5 | 0.5836 | 0.5841 | 0.5793 | 0.9273 | 0.9322 | 0.9244 | 1.3095 | 1.3172 | 1.3062 | 1.8454 | 1.8605 | 1.8449 |
| 10 | 0.5839 | 0.5843 | 0.5795 | 0.9257 | 0.9298 | 0.9221 | 1.3059* | 1.3174 | 1.3064 | 1.8026* | 1.8623 | 1.8467 |
| 25 | 0.5877 | 0.5923 | 0.5875 | 0.9257 | 0.9310 | 0.9233 | 1.2867* | 1.3175 | 1.3065 | stopped | | |
| 50 | 0.5874 | 0.5915 | 0.5866 | 0.9238* | 0.9335 | 0.9258 | 1.2651* | 1.3172 | 1.3062 | stopped | | |
| 100 | 0.5876 | 0.5886 | 0.5837 | 0.9202* | 0.9314 | 0.9237 | stopped | | | stopped | | |
| 150 | 0.5876 | 0.5899 | 0.5851 | 0.9189* | 0.9304 | 0.9226 | stopped | | | stopped | | |
| 200 | 0.5877 | 0.5897 | 0.5849 | 0.9179* | 0.9308 | 0.9231 | stopped | | | stopped | | |
| $^{\infty}L$ | 0.5870 | | | 0.9230 | | | 1.3060 | | | 1.8460 | | |
| $^{\infty}U$ | 0.5920 | | | 0.9310 | | | 1.3170 | | | 1.8620 | | |

| | 50 | | | 75 | | | 100 | | | 200 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $n$ | CT | $^{\infty}DU$ | $^{\infty}DL$ | CT | $^{\infty}DU$ | $^{\infty}DL$ | CT | $^{\infty}DU$ | $^{\infty}DL$ | CT | $^{\infty}DU$ | $^{\infty}DL$ |
| 1 | 2.9415 | 2.9626 | 2.9376 | 3.6043 | 3.6331 | 3.6021 | 4.1647 | 4.1964 | 4.1603 | 5.8926 | 5.9190 | 5.8667 |
| 2 | 2.9345 | 2.9464 | 2.9213 | 3.5976* | 3.5947 | 3.5637 | 4.1570 | 4.1739 | 4.1379 | 5.8863 | 5.9096 | 5.8533 |
| 3 | 2.9193* | 2.9536 | 2.9285 | 3.5567* | 3.6240 | 3.5928 | 4.0794* | 4.1874 | 4.1512 | stopped | | |
| 4 | 2.8784* | 2.9482 | 2.9231 | 3.4499* | 3.5967 | 3.5658 | 3.8945* | 4.1789 | 4.1427 | stopped | | |
| 5 | 2.8168* | 2.9473 | 2.9222 | stopped | | | stopped | | | stopped | | |
| 10 | stopped | | | stopped | | | stopped | | | stopped | | |
| 25 | stopped | | | stopped | | | stopped | | | stopped | | |
| $^{\infty}L$ | 2.9180 | | | 3.5730 | | | 4.1420 | | | 5.8620 | | |
| $^{\infty}U$ | 2.9440 | | | 3.6050 | | | 4.1790 | | | 5.9160 | | |

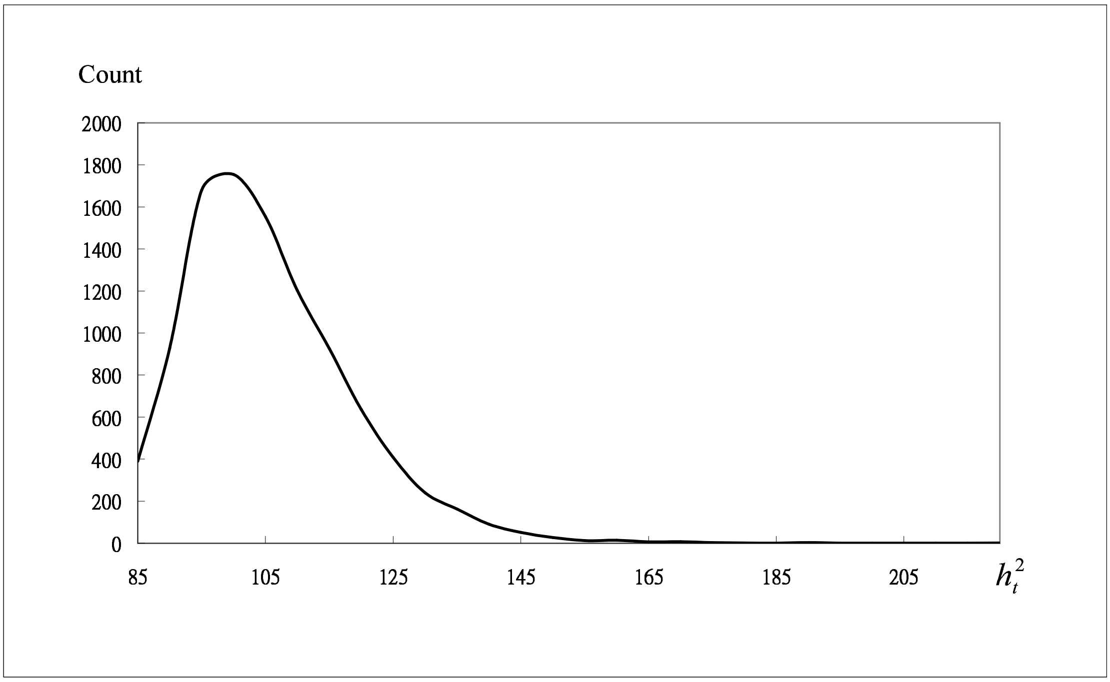
The parameters are $S_0 = 100$, $r = 0$, $h_0^2 = 0.0001096$, $\gamma = h_0 = 0.010469$, $\beta_0 = 0.000006575$, $\beta_1 = 0.9$, $\beta_2 = 0.04$, $K = 20$, and $c = 0$. The CT algorithm is used here. The option is a European call with a strike price of 100. The CT tree may stop growing before the required maturity if $n$ crosses a threshold. $^{\infty}DL$ and $^{\infty}DU$ form the 95% confidence interval for the option price based on Monte Carlo simulation of the tree model (7) with 500,000 paths. Asterics mark option prices that lie outside this interval. $^{\infty}L$ and $^{\infty}U$ form the 95% confidence interval for the true option price based on Monte Carlo simulation of the continuous-state model (2) with 500,000 paths.

Figure 12: **Select option prices from Table 3**.



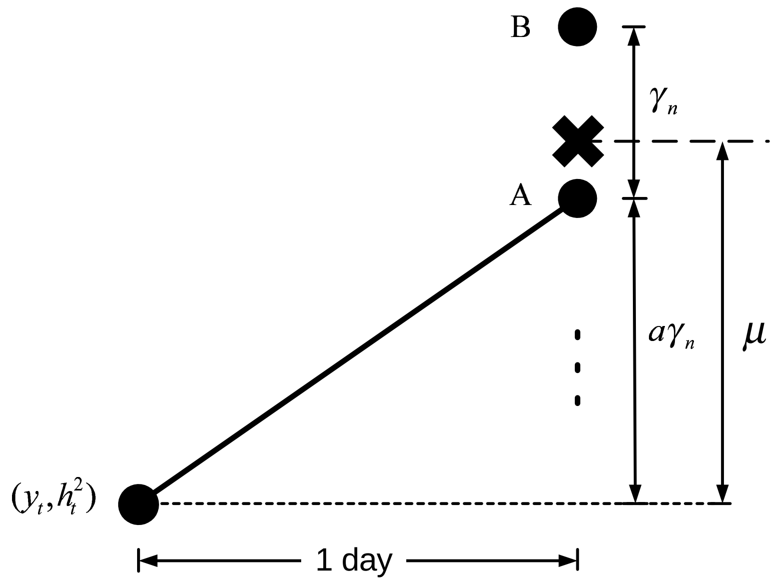

`MC lower bound` equals ${}^{\infty}L$; `MC upper bound` equals ${}^{\infty}U$.

Figure 13: **Volatility distribution**.



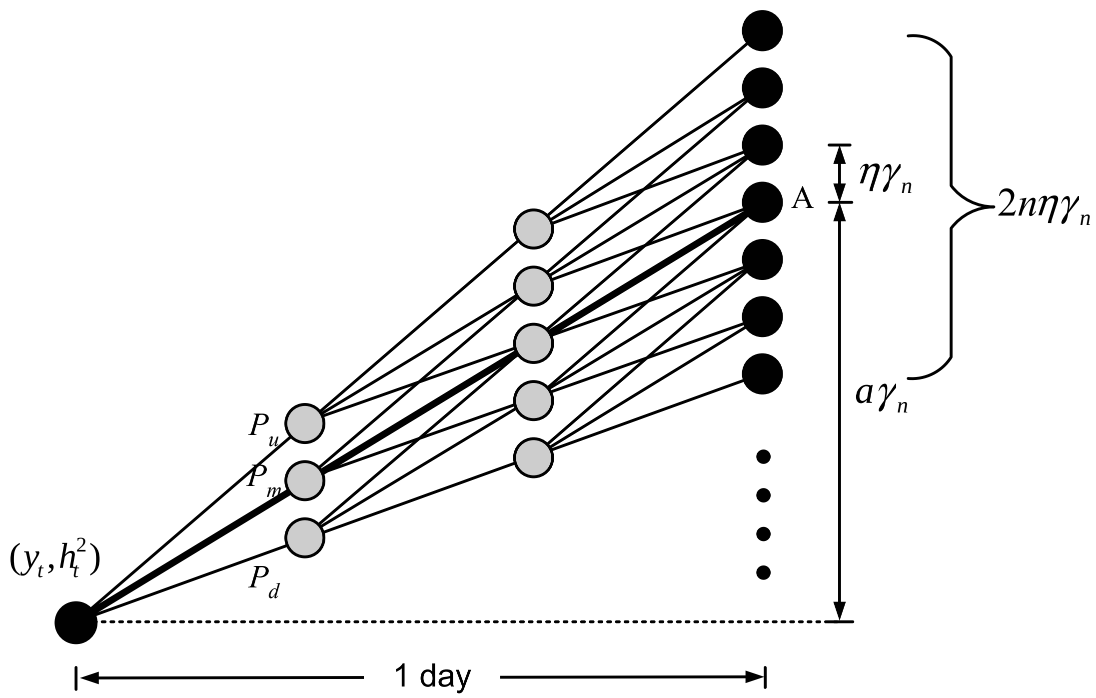

All parameters are from Table 3 with $n = 10$ and $N = 10$. Monte Carlo simulation on the tree model (7) is used to record the volatilities at the middle node at maturity. Out of 500,000 paths, about 10,000 reach the node. The recorded minimum and maximum squared volatilities are 0.000085 and 0.000217, respectively. Squared volatilities $h_t^2$ are multiplied by 1,000,000 for easy reading.

Figure 14: **Location of next middle node.**



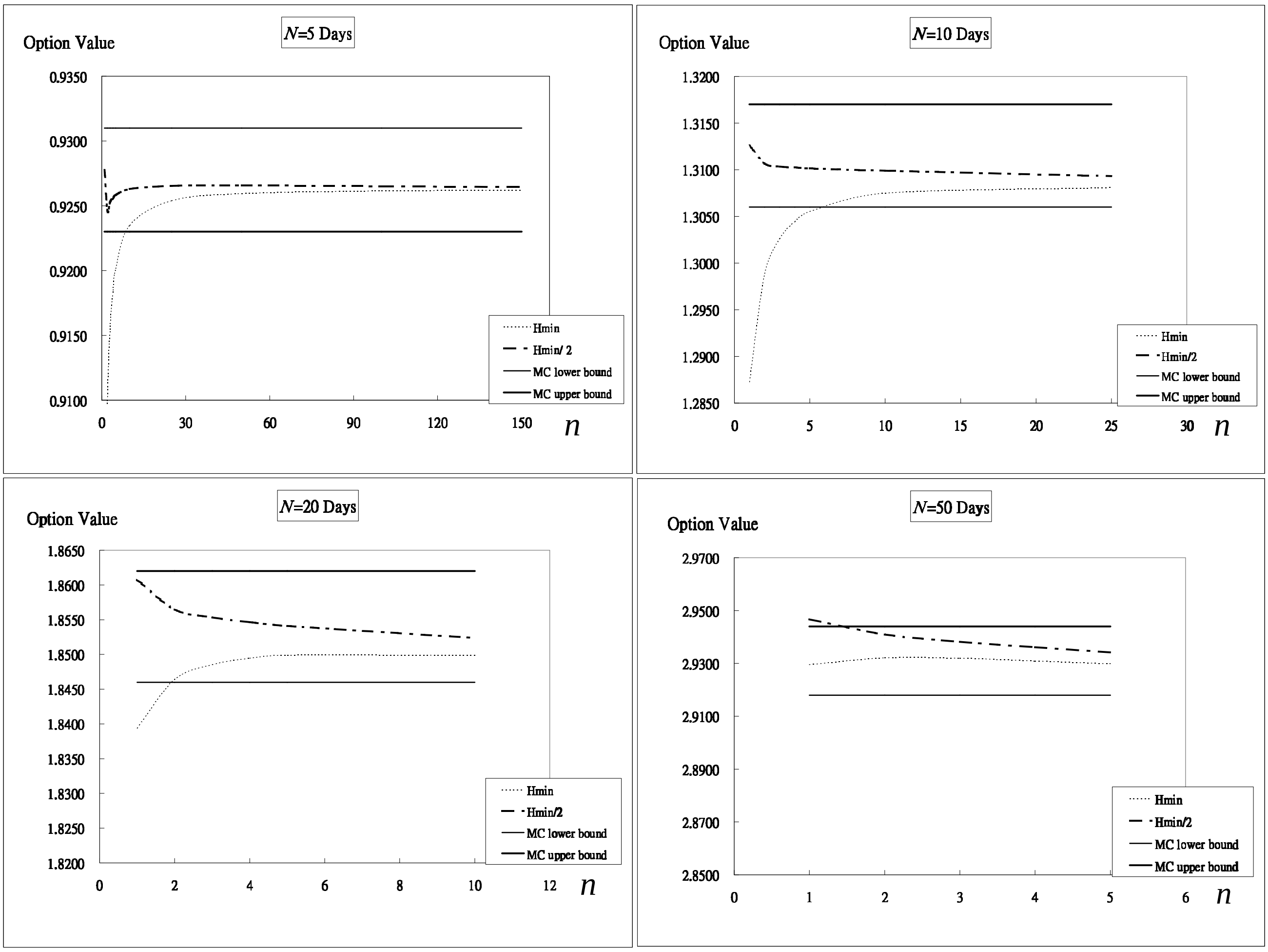

The cross identifies the true mean of $y_{t+1}$. Two nodes, A and B, bracket it. Between them, node A has a logarithmic price closer to the mean. The number $a\gamma_n$ denotes the difference between $y_t$ and node A's logarithmic price at date $t+1$.

Figure 15: **The MT trinomial tree for logarithmic price.**




A day is partitioned into $n = 3$ periods, and the three jump sizes in each period are $(a\gamma_n/n) + \eta\gamma_n$ (upward), $a\gamma_n/n$ (middle), and $(a\gamma_n/n) - \eta\gamma_n$ (downward). The central branch of the tree lines up with node A, the node closest to the mean of $y_{t+1}$ as stated in Fig. 14. The gray nodes are for illustration only; only the solid nodes are actually used in pricing. Except for the drift, the actual tree is heptanomial as in Fig. 5.

Figure 16: **Two choices of $\gamma$ in MT.**




All parameters are from Table 3. `Hmin` means the option prices are obtained with the choice $\gamma = H_{\min}$. `Hmin/2` means the option prices are obtained with the choice $\gamma = H_{\min}/2$. `MC upper bound` and `MC lower bound` form the 95% confidence interval for the true option price based on Monte Carlo simulation of the continuous-state model (2) with 500,000 paths.

Table 4: **Case where RTCT fails but MT succeeds.**

CT

| | $K$ | | | | | |
|---|---|---|---|---|---|---|
| $n$ | 2 | 10 | 20 | 50 | 100 | 200 |
| 1 | 4.2301 | 4.2365 | 4.2267 | 4.2274 | 4.2265 | 4.2268 |
| 2 | 4.1769 | 4.2617 | 4.2677 | 4.2697 | 4.2705 | 4.2704 |
| 3 | 3.8894 | 4.1326 | 4.1899 | 4.2361 | 4.2543 | 4.2636 |
| 4 | 3.7162 | 3.9266 | 4.0006 | 4.0877 | 4.1433 | 4.1883 |
| $^{\infty}L$ | 4.2714 | | | | | |
| $^{\infty}U$ | 4.3087 | | | | | |

MT

| | $K$ | | | | | |
|---|---|---|---|---|---|---|
| $n$ | 2 | 10 | 20 | 50 | 100 | 200 |
| 1 | 4.2741 | 4.2799 | 4.2804 | 4.2805 | 4.2806 | 4.2806 |
| 2 | 4.1941 | 4.2714 | 4.2746 | 4.2753 | 4.2754 | 4.2754 |
| 3 | 3.9900 | 4.2619 | 4.2700 | 4.2731 | 4.2737 | 4.2744 |
| 4 | 3.7989 | 4.2518 | 4.2658 | 4.2714 | 4.2725 | 4.2742 |
| $^{\infty}L$ | 4.2714 | | | | | |
| $^{\infty}U$ | 4.3087 | | | | | |

All parameters are from Table 2 except that $n$ is varied. Although the RTCT tree explodes with $n \geq 3$, the cut-off maturity happens to exceed $N = 100$ days here. Hence the option for $n \geq 3$ can still be priced albeit with great difficulty. We stop at $n = 4$ because the RTCT tree is cut short for $n > 4$ as proved in section 4.1.3.

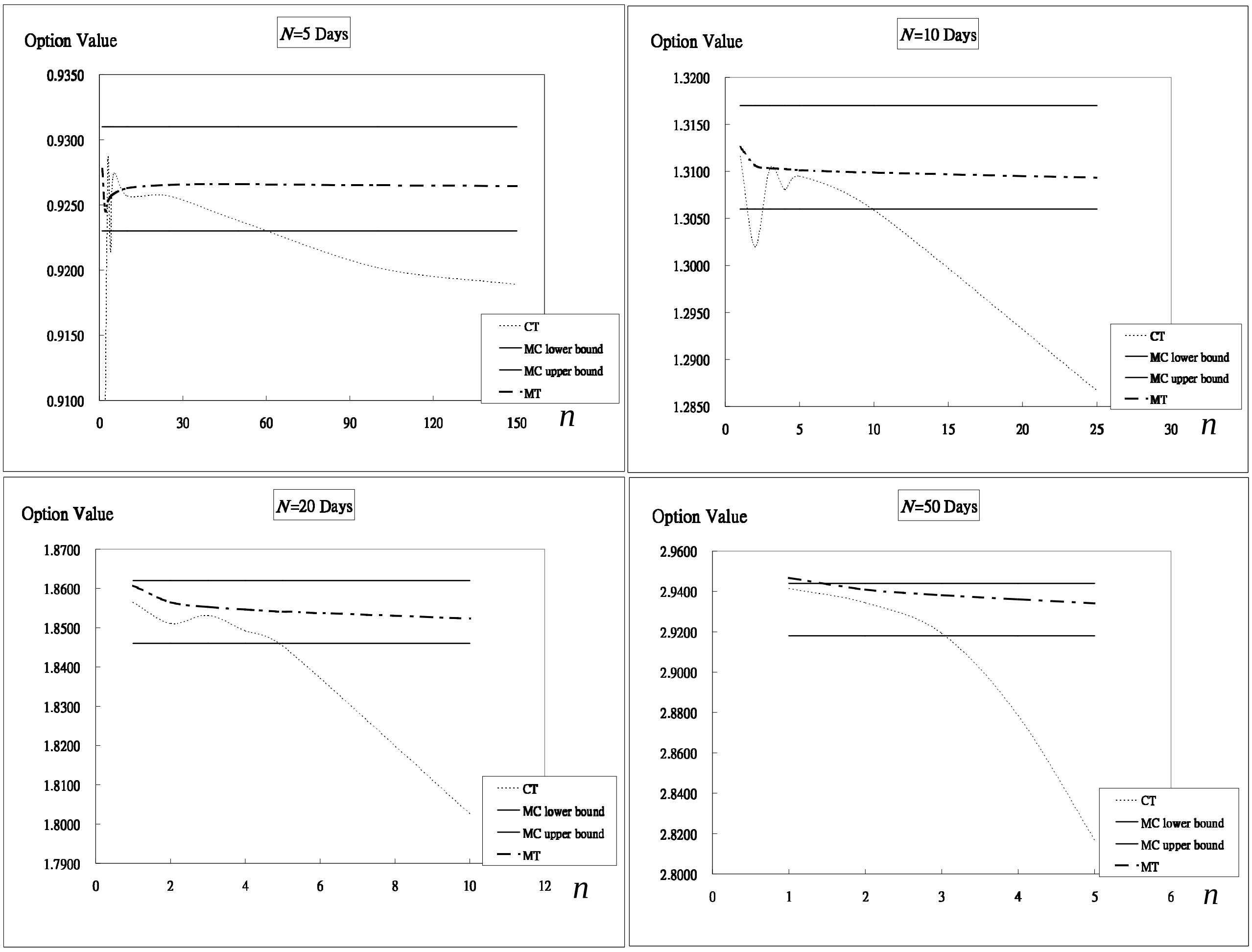
Table 5: **Convergence of MT.**

| | Maturity of Option (Days) | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 2 | | | 5 | | | 10 | | | 20 | | |
| $n$ | MT | $^{\infty}DU$ | $^{\infty}DL$ | MT | $^{\infty}DU$ | $^{\infty}DL$ | MT | $^{\infty}DU$ | $^{\infty}DL$ | MT | $^{\infty}DU$ | $^{\infty}DL$ |
| 1 | 0.5626 | 0.5648 | 0.5599 | 0.9278 | 0.9319 | 0.9242 | 1.3126 | 1.3190 | 1.3080 | 1.8608 | 1.8763 | 1.8607 |
| 2 | 0.5799 | 0.5841 | 0.5792 | 0.9246 | 0.9264 | 0.9187 | 1.3107 | 1.3190 | 1.3080 | 1.8565 | 1.8675 | 1.8519 |
| 3 | 0.5833 | 0.5854 | 0.5806 | 0.9253 | 0.9308 | 0.9231 | 1.3104 | 1.3149 | 1.3039 | 1.8553 | 1.8674 | 1.8517 |
| 4 | 0.5845 | 0.5874 | 0.5825 | 0.9256 | 0.9307 | 0.9229 | 1.3103 | 1.3170 | 1.3060 | 1.8547 | 1.8590 | 1.8434 |
| 5 | 0.5851 | 0.5883 | 0.5835 | 0.9258 | 0.9274 | 0.9197 | 1.3102 | 1.3129 | 1.3020 | 1.8541 | 1.8635 | 1.8479 |
| 10 | 0.5864 | 0.5868 | 0.5820 | 0.9263 | 0.9279 | 0.9202 | 1.3099 | 1.3132 | 1.3022 | 1.8524 | 1.8646 | 1.8490 |
| 25 | 0.5872 | 0.5897 | 0.5848 | 0.9265 | 0.9302 | 0.9225 | 1.3093 | 1.3126 | 1.3017 | | | |
| 50 | 0.5874 | 0.5907 | 0.5859 | 0.9266 | 0.9314 | 0.9237 | 1.3090 | 1.3139 | 1.3029 | | | |
| 100 | 0.5876 | 0.5880 | 0.5832 | 0.9265 | 0.9301 | 0.9223 | | | | | | |
| 150 | 0.5876 | 0.5891 | 0.5842 | 0.9265 | 0.9289 | 0.9212 | | | | | | |
| 200 | 0.5876 | 0.5888 | 0.5840 | 0.9265 | 0.9292 | 0.9224 | | | | | | |
| $^{\infty}L$ | 0.5870 | | | 0.9230 | | | 1.3060 | | | 1.8460 | | |
| $^{\infty}U$ | 0.5920 | | | 0.9310 | | | 1.3170 | | | 1.8620 | | |

| | 50 | | | 75 | | | 100 | | | 200 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $n$ | MT | $^{\infty}DU$ | $^{\infty}DL$ | MT | $^{\infty}DU$ | $^{\infty}DL$ | MT | $^{\infty}DU$ | $^{\infty}DL$ | MT | $^{\infty}DU$ | $^{\infty}DL$ |
| 1 | 2.9468 | 2.9547 | 2.9296 | 3.6105 | 3.6335 | 3.6025 | 4.1698 | 4.1919 | 4.1558 | 5.8974 | 5.9346 | 5.8822 |
| 2 | 2.9410 | 2.9539 | 2.9288 | 3.6045 | 3.6153 | 3.5843 | 4.1640 | 4.1908 | 4.1546 | 5.8926 | 5.9205 | 5.8682 |
| 3 | 2.9383 | 2.9576 | 2.9324 | 3.6009 | 3.6290 | 3.5980 | 4.1595 | 4.1770 | 4.1409 | | | |
| 4 | 2.9362 | 2.9536 | 2.9285 | 3.5978 | 3.6260 | 3.5949 | 4.1555 | 4.1704 | 4.1344 | | | |
| 5 | 2.9342 | 2.9554 | 2.9302 | | | | | | | | | |
| 10 | | | | | | | | | | | | |
| 25 | | | | | | | | | | | | |
| $^{\infty}L$ | 2.9180 | | | 3.5730 | | | 4.1420 | | | 5.8620 | | |
| $^{\infty}U$ | 2.9440 | | | 3.6050 | | | 4.1790 | | | 5.9160 | | |

All parameters are from Table 3. $^{\infty}DL$ and $^{\infty}DU$ form the 95% confidence interval for the true option price based on Monte Carlo simulation of the tree model (14) with 500,000 paths. None of the option prices lie outside this confidence interval. $^{\infty}L$ and $^{\infty}U$ form the 95% confidence interval for the true option price based on Monte Carlo simulation of the continuous-state model (2) with 500,000 paths.

Figure 17: **Select option prices from Table 5**.




`MC lower bound` equals ${}^{\infty}L$; `MC upper bound` equals ${}^{\infty}U$. The option prices under the CT algorithm come from Fig. 12.

Table 6: **Accuracy of MT with nonzero $r$ and $c$.**

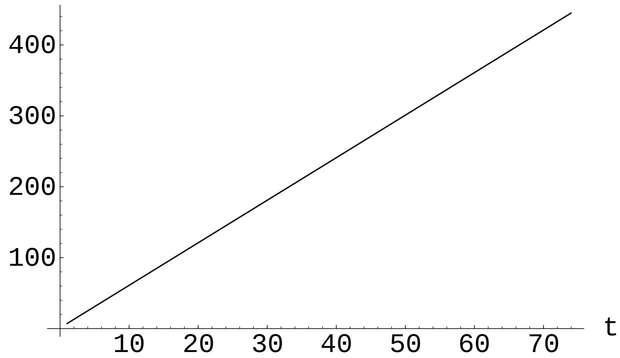
| | Maturity of Option (Days) = 30 | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *X=55* | | | | *X=50* | | | | *X=45* | | | |
| | *K* | | | | *K* | | | | *K* | | | |
| *n* | 10 | 20 | 50 | 100 | 10 | 20 | 50 | 100 | 10 | 20 | 50 | 100 |
| 1 | 4.8278 | 4.8576 | 4.8275 | 4.8275 | 1.1038 | 1.1043 | 1.1045 | 1.1045 | 0.0746 | 0.0745 | 0.0744 | 0.0744 |
| 2 | 4.8350 | 4.8342 | 4.8340 | 4.8340 | 1.0931 | 1.0950 | 1.0955 | 1.0955 | 0.0767 | 0.0762 | 0.0761 | 0.0760 |
| 3 | 4.8386 | 4.8366 | 4.8360 | 4.8359 | 1.0864 | 1.0912 | 1.0926 | 1.0927 | 0.0780 | 0.0769 | 0.0765 | 0.0765 |
| $^{\infty}L$ | 4.8364 | | | | 1.0862 | | | | 0.0764 | | | |
| $^{\infty}U$ | 4.8412 | | | | 1.0898 | | | | 0.0792 | | | |
| DS | 4.8377 | | | | 1.0884 | | | | 0.0715 | | | |

| | Maturity of Option (Days) = 90 | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *X=55* | | | | *X=50* | | | | *X=45* | | | |
| | *K* | | | | *K* | | | | *K* | | | |
| *n* | 10 | 20 | 50 | 100 | 10 | 20 | 50 | 100 | 10 | 20 | 50 | 100 |
| 1 | 4.9513 | 4.9519 | 4.9520 | 4.9520 | 1.8361 | 1.8393 | 1.8398 | 1.8399 | 0.4210 | 0.4215 | 0.4216 | 0.4216 |
| $^{\infty}L$ | 4.9505 | | | | 1.8162 | | | | 0.4131 | | | |
| $^{\infty}U$ | 4.9587 | | | | 1.8232 | | | | 0.4185 | | | |
| DS | 4.9550 | | | | 1.8197 | | | | 0.4036 | | | |

| | Maturity of Option (Days) = 270 | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *X=55* | | | | *X=50* | | | | *X=45* | | | |
| | *K* | | | | *K* | | | | *K* | | | |
| *n* | 10 | 20 | 50 | 100 | 10 | 20 | 50 | 100 | 10 | 20 | 50 | 100 |
| 1 | 5.4806 | 5.4869 | 5.4880 | 5.4881 | 2.8505 | 2.8565 | 2.8573 | 2.8574 | 1.2016 | 1.2036 | 1.2039 | 1.2039 |
| $^{\infty}L$ | 5.4708 | | | | 2.8361 | | | | 1.1902 | | | |
| $^{\infty}U$ | 5.4838 | | | | 2.8471 | | | | 1.1988 | | | |
| DS | 5.4899 | | | | 2.8471 | | | | 1.1867 | | | |

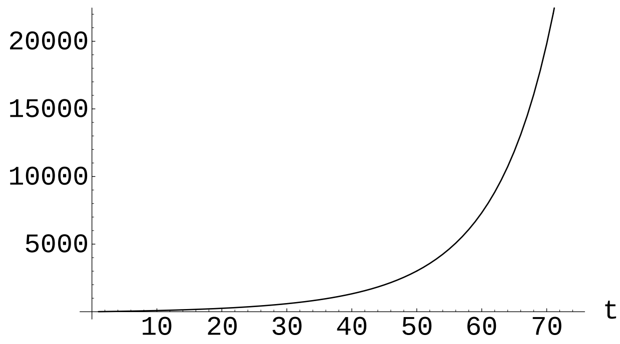
All parameters are from Duan and Simonato (2001): $S_0 = 50$, $r = 5\%$ (annual), $h_0^2 = 0.0001096$, $\beta_0 = 0.00001$, $\beta_1 = 0.8$, $\beta_2 = 0.1$, and $c = 0.5$. The option is a European put with a strike price of $X$. $^{\infty}L$ and $^{\infty}U$ form the 95% confidence interval for the true option price based on Monte Carlo data from Duan and Simonato (2001). DS lists the option prices from Duan and Simonato (2001) using the most computational efforts.

Figure 18: **Linear growth of the MT tree.**




The parameters are from Table 6 except for $n = 1$, $c = 0$, $N = 74$, and $K = 20$. This set of parameters satisfy inequality (22). The number of nodes grows linearly with date $t$. Hence the total number of nodes grows quadratically in maturity $N$.

Figure 19: **Exponential growth of the MT tree.**




The parameters are identical to those in Fig. 18 except for $c = 0.9$. This set of parameters violate inequality (22). The number of nodes grows exponentially with date $t$.