AN ALTERNATIVE TREE METHOD FOR CALIBRATION OF THE LOCAL VOLATILITY

WENXIU GONG
College of Mathematics and Systems Science
Shandong University of Science and Technology, Qingdao, 266590, China

ZUOLIANG XU∗
School of Mathematics, Renmin University of China
Beijing, 100872, China

(Communicated by Song Wang)

Abstract. In this paper, we combine the traditional binomial tree and trinomial tree to construct a new alternative tree pricing model, where the local volatility is a deterministic function of time. We then prove the convergence rates of the alternative tree method. The proposed model can price a wide range of derivatives efficiently and accurately. In addition, we research the optimization approach for the calibration of local volatility. The calibration problem can be transformed into a nonlinear unconstrained optimization problem by exterior penalty method. For the optimization problem, we use the quasi-Newton algorithm. Finally, we test our model by numerical examples and options data on the S&P 500 index. Numerical results confirm the excellent performance of the alternative tree pricing model.

1. Introduction. Black and Scholes’ breakthrough shows that options can be replicated by trading bonds and the underlying asset [6]. But it is well known that the famous Black and Scholes(BS) option pricing model assumes the price of underlying asset follows a log-normal distribution with a constant volatility, which is not supported by the market data. In other words, options with various maturities and strike prices have different volatilities by the BS model.

The tree method is widely used in option pricing because of its simplicity, efficiency and the ability to deal with the early exercise characteristic of the American options. The binomial tree method(BTM), as a discrete model proposed by Cox, Ross and Rubinstein(CRR)[11], is the most popular approach to pricing options. There are many researchers who investigate other trees [8, 15, 22, 24]. The option prices computed by these models convergence to the theoretical option value as the the number of time steps increases. However, the prices may converge slowly or even oscillate significantly [18]. Chang and Palmer [9] propose a version of the
CRR tree with a smaller error of $O(\Delta t)$. In order to decrease the oscillation of the binomial tree, Dai and Lyuu [13] introduce a novel tree model, the bino-trinomial tree, that can price smoothly and quickly. The bino-trinomial tree is essentially a binomial tree with occasional trinomial structures for added flexibility.

Trinomial trees are regarded as a generalization of binomial trees. Since the asset price in a trinomial tree moves in three directions compared with only two for a binomial tree, it is more accurate and flexible than the binomial trees for the same number of steps. Trinomial tree method (TTM) is constructed by Boyle [7] for option pricing using moments matching techniques. After that, Ahn and Song [1] study the convergence of the TTM, and prove the convergence and accuracy of the model. However, the extra degrees of freedom will lead to cost much more time to compute than binomial tree. In addition, the binomial tree and trinomial tree cannot fit the implied volatility surface well because they assume the volatility is a constant.

As a remedy, Dupire [16] considers the local volatility model, where the volatility is a deterministic function of asset price and time, i.e. $\sigma = \sigma(S,t)$. The value of European call option $C(S,t)$ satisfies the following equation:

$$\frac{\partial C}{\partial t} + \frac{1}{2} \sigma^2(S,t)S^2 \frac{\partial^2 C}{\partial S^2} + rS \frac{\partial C}{\partial S} - rC = 0, (0 \leq S < \infty, 0 \leq t < T), \quad (1)$$

$$C(S,T) = (S - K)^+, (0 \leq S < \infty). \quad (2)$$

He derives Dupire’s equation, which provides a very useful insight into the inverse problem of calibrating the local volatility model. Amin [2] builds a tree model with time-varying volatility functions for option pricing. A tree for a known local volatility surface is called local volatility tree, whereas an implied tree fits the implied volatility surface generated by a local volatility tree. The construction of the implied tree is thus an inverse problem, which in general is ill-posed [4]. Derman et al. [14] propose an implied trinomial tree, but negative probabilities will linger. Barle and Cakici [5] improve the DKC (Derman, Kani and Chriss) model with a better placement of nodes which aims to eliminate the occurrences of negative transition probabilities. Li [19] develops an implied binomial tree which sets the transition probabilities to some constant, so negative probabilities will vanish by default. Crépey [12] calibrates the local volatility in a trinomial tree using Tikhonov regularization. Talias [23] calibrates the non-recombining tree by genetic algorithms. Charalambous et al. [10] construct an implied non-recombining tree and calibrate the volatility smile by quasi-Newton algorithm. Lok and Lyuu [20] find a potentially fundamental reason why the negative probabilities linger and construct a waterline tree for separable local volatility which combines two kinds of binomial trees by matching the moments of the underlying asset price and logarithmic asset price. Gong and Xu [17] propose an implied non-recombining trinomial tree and calibrate the volatility smile. Table 1 shows some tree methods for calibration of the local volatility.

In this paper, we combine the traditional binomial tree and trinomial tree to propose a new alternative tree pricing model that the local volatility function only depend on time. We then prove the convergence rates of the alternative tree pricing model. In addition, we calibrate the alternative tree based on optimization. In particular, we attempt to minimize the least squares error between the market prices and the theoretical prices with respect to the underlying asset values at each node of the alternative tree, subject to constraints of risk neutrality and no-arbitrage.
### Table 1. Some tree methods for calibration of the local volatility

| Author          | Tree method                  | Volatility function |
|-----------------|------------------------------|---------------------|
| Derman(1996), Barle(1999) | Recombining TTM                | $\sigma = \sigma(S, t)$ |
| Li(2001)        | Recombining BTM               | $\sigma = \sigma(S, t)$ |
| Crépey (2003)   | TTM with regularization       | $\sigma = \sigma(S, t)$ |
| Charalambous et al. (2007) | Nonrecombining BTM        | $\sigma = \sigma(t)$ |
| Lok and Lyuu (2017) | Recombining waterline tree  | $\sigma = \sigma(S)\sigma(t)$ |
| Gong and Xu (2019) | Nonrecombining TTM          | $\sigma = \sigma(t)$ |

Calibrating an alternative tree means that finding the stock prices and transition probabilities at each node, which make the tree reproduce the option price that is consistent with today’s market price. The calibration problem is a nonlinear optimization problem with constraints. Especially, we adopt a penalty method to transform the nonlinear optimization problem with constraints into a nonlinear unconstrained problem. For the optimization, we use a quasi-Newton algorithm.

In contrast to other implied trees, our model neither employs node substitutions nor sets the transition probabilities to some constant. In addition, we do not need any interpolation or extrapolation across strikes and time in the process of calibration, and our model can be easily modified to account for options with different maturities. Finally, if we need, we can add smoothness constraints on the distribution of the underlying asset for the extra degrees of freedom and the analytical structure of the model.

The paper is organized as follows. Brief introduction to binomial and trinomial tree pricing model appears in section 2. In section 3, we consider the proposed alternative tree pricing model, including the construction of alternative tree and risk neutrality and no-arbitrage constraints. In section 4, we discuss the convergence rates of the proposed method. In section 5, we discuss the optimization algorithm and adopt a penalty method. For the optimization, we use a quasi-Newton algorithm. Numerical and empirical results are presented in section 6. Conclusions are in section 7.

#### 2. Binomial and trinomial tree pricing model

The BS model assumes the stock price evolves according to the geometric Brownian motion:

$$
\frac{dS_t}{S_t} = r dt + \sigma dW_t \tag{3}
$$

where $S_t$ is the stock price at time $t$, $r$ is the risk-free rate, $\sigma$ is the stock price's constant volatility, and $W_t$ is the standard Brownian motion.

Now consider a binomial tree for the geometric Brownian motion (3). Let $u > d$, $ud = 1$, $\Delta t = \frac{T}{n}$, where $T$ is the expiration time which is divided by total number of time steps $n$. After one time step $\Delta t$, the stock price $S$ can move to $uS$ with probability $p$ or to $dS$ with probability $1 - p$. According to the mean and variance of the stock price, we have the popular CRR tree parameters:

$$
u = e^{\sigma \sqrt{\Delta t}}, \quad d = e^{-\sigma \sqrt{\Delta t}}, \quad p = \frac{e^{r\Delta t} - d}{u - d} \tag{4}$$

It is well known that the CRR tree converges to the BS model as the time steps go to infinity. However, the CRR tree pricing model oscillates significantly. We can
see from the left figure of the Figure 1 that the oscillation depends heavily on the parity of the numbers of time steps.

TTM is regarded as a much better approximation to the BS model. $P_u$, $P_m$, $P_d$ denote the risk-neutral probabilities corresponding to the stock price increases, remains the same and decreases, respectively. According to moment information of the stock price, we have the parameters:

\[ u = e^{\sigma \sqrt{\Delta t}}, d = e^{-\sigma \sqrt{\Delta t}}, P_u = \frac{(1 + d)e^{r\Delta t} - e^{(2r + \sigma^2)\Delta t} - d}{(d - u)(u - 1)}, P_m = \frac{(1 + u)e^{r\Delta t} - e^{(2r + \sigma^2)\Delta t} - u}{(d - u)(1 - d)}, P_d = \frac{(1 + u)e^{r\Delta t} - e^{(2r + \sigma^2)\Delta t} - u}{(d - u)(1 - d)}. \]

It can be seen from the right figure of the Figure 1 that the trinomial tree pricing model depends less on the parity of the time steps, and gradually converges to BS option price steadily. However, when the number of time steps is small, the trinomial tree pricing model is not stable and oscillates largely. Although it gradually converges as the number of time steps increases, the calculation will cost much more time.

**Figure 1.** The left figure presents CRR method and steps while the right figure presents TTM and steps. The blue line denotes the BS price. The red line denotes the CRR and TTM price. The green line denotes CRR price with odd steps while the black line denotes CRR price with even steps.

3. **A new alternative tree for local volatility models.** According to the traditional binomial tree and trinomial tree pricing method, we know that the CRR tree pricing model depends heavily on the parity of the time steps, and waves around the price of BS model. Trinomial tree pricing model depends less on the parity of the time step number, and gradually converges to BS option price steadily. When the number of time steps is small, the trinomial tree pricing model is not stable and waves largely. Although it gradually converges as the number of steps increases, the cost of computing will increase. Therefore, we select the time steps that make the binomial tree wave oppositely to the trinomial tree pricing model, and then combine with the trinomial tree to construct a new alternative tree to eliminate error. From the right figure of Figure 1, we can see that the trinomial tree increasingly converges to the theoretical value, so we select the steps that make the binomial
tree decreasingly converge to the theoretical value. From the left figure of Figure 1, we will select the odd time steps for binomial tree.

Our goal is to construct an arbitrage-free risk neutral model that attempts to fit the implied volatility, and then can be used to price options. In order to allow more degrees of freedom and reduce its compute, we construct an alternative tree based on trinomial and binomial tree. In this section, we will derive the proposed model, and consider the risk neutrality and no-arbitrage constraints of the model.

3.1. **Construction of the alternative tree.** Before constructing the alternative tree, we discuss the conditions under which the alternative tree could be constructed.

**Assumption 1** The underlying asset price $S$ is governed by the following stochastic differential equation

$$
\frac{dS_t}{S_t} = r dt + \sigma(t) dW_t
$$

(6)

where $S_t$ is the stock price at time $t$, $r$ is the risk-free interest rate, $\sigma(t)$ is the local volatility function of time, and $W_t$ follows the Wiener process with mean of 0 and variance of $t$.

**Assumption 2** The risk-free interest rate $r$ and the local volatility function $\sigma(t)$ are continuous and non-negative.

Under the assumptions, we construct an alternative tree. Figure 2 shows an alternative tree with two steps. Let $i$ and $j$ denote the time and asset dimension. The time step is $\Delta t = \frac{T}{n-1}$; the value of the underlying asset at node $(i,j)$ is $S(i,j)$; the current value of the underlying asset is $S(1,1)$.

Let $u_1(i,j)$, $d_1(i,j)$ denote the up and down factors by which the underlying asset price can move in a single time step $\Delta t$ of the trinomial tree, and let $u_1(i,j)d_1(i,j) = 1$. Let $S(i+1,3j)$, $S(i+1,3j-1)$ and $S(i+1,3j-2)$ be the asset prices when the underlying asset price $S(i,j)$ increases, remains the same and decreases, respectively. $P_{u_1}(i,j)$, $P_{m_1}(i,j)$, $P_{d_1}(i,j)$ denote the risk-neutral probabilities corresponding to the asset price increases, remains the same and decreases.
respectively. Using the moment information of underlying asset price, we have

\[
\begin{align*}
P_{u_1}(i,j) + P_{m_1}(i,j) + P_{d_1}(i,j) &= 1, \\
P_{u_1}(i,j)S(i + 1, 3j) + P_{m_1}(i,j)S(i + 1, 3j - 1) + P_{d_1}(i,j)S(i + 1, 3j - 2) \\
&\quad = S(i,j)e^{r\Delta t}, \\
P_{u_1}(i,j)S(i + 1, 3j)^2 + P_{m_1}(i,j)S(i + 1, 3j - 1)^2 + P_{d_1}(i,j)S(i + 1, 3j - 2)^2 \\
&\quad = S(i,j)^2e^{(2r+\sigma_j^2)\Delta t}, \\
P_{u_1}(i,j)S(i + 1, 3j)^3 + P_{m_1}(i,j)S(i + 1, 3j - 1)^3 + P_{d_1}(i,j)S(i + 1, 3j - 2)^3 \\
&\quad = S(i,j)^3e^{(3r+3\sigma_j^2)\Delta t}, \\
\end{align*}
\]

Let \( u_1(i,j) \) be the up factor by which the underlying asset price can move in a single time step \( \Delta t \) of the binomial tree,

\[
\begin{align*}
\begin{cases}
u_1(i,j) &= e^{\sigma_j\sqrt{3\Delta t}}, \\d_1(i,j) &= e^{-\sigma_j\sqrt{3\Delta t}}, \\
P_{u_1}(i,j) &= \frac{A(i,j)}{[S(i + 1, 3j - 2) - S(i + 1, 3j)][S(i + 1, 3j) - S(i + 1, 3j - 1)]}, \\
P_{m_1}(i,j) &= \frac{B(i,j)}{[S(i + 1, 3j - 1) - S(i + 1, 3j - 2)][S(i + 1, 3j) - S(i + 1, 3j - 1)]}, \\
P_{d_1}(i,j) &= \frac{C(i,j)}{[S(i + 1, 3j - 2) - S(i + 1, 3j)][S(i + 1, 3j - 1) - S(i + 1, 3j - 2)]},
\end{cases}
\end{align*}
\]

where

\[
\begin{align*}
A(i,j) &= [S(i + 1, 3j - 1) + S(i + 1, 3j - 2)]S(i,j)e^{r\Delta t} - e^{(2r+\sigma_j^2)\Delta t}S(i,j)^2 - S(i + 1, 3j - 1)S(i + 1, 3j - 2), \\
B(i,j) &= [S(i + 1, 3j) + S(i + 1, 3j - 2)]S(i,j)e^{r\Delta t} - e^{(2r+\sigma_j^2)\Delta t}S(i,j)^2 - S(i + 1, 3j - 1)^2, \\
C(i,j) &= [S(i + 1, 3j - 1) + S(i + 1, 3j)]S(i,j)e^{r\Delta t} - e^{(2r+\sigma_j^2)\Delta t}S(i,j)^2 - S(i + 1, 3j)S(i + 1, 3j - 1).
\end{align*}
\]

Similarly, let \( u_2(i,j), d_2(i,j) \) be the up and down factors by which the underlying asset price can move in a single time step \( \Delta t \) of the binomial tree, and let \( u_2(i,j)d_2(i,j) = 1 \). \( P_{u_2}(i,j), P_{d_2}(i,j) \) denote the risk-neutral probabilities corresponding to the asset price increases and decreases, respectively. We have

\[
\begin{align*}
\begin{cases}
u_2(i,j) + P_{d_2}(i,j) &= 1, \\
u_2(i,j)S(i + 1, 2j) + P_{d_2}(i,j)S(i + 1, 2j - 1) &= S(i,j)e^{r\Delta t}, \\
u_2(i,j)S(i + 1, 2j)^2 + P_{d_2}(i,j)S(i + 1, 2j - 1)^2 &= S(i,j)^2e^{(2r+\sigma_j^2)\Delta t}, \\
u_2(i,j)d_2(i,j) &= 1.
\end{cases}
\end{align*}
\]
Hence

\[
\begin{align*}
  u_2(i, j) &= e^{\sigma \sqrt{\Delta t}}, \\
  d_2(i, j) &= e^{-\sigma \sqrt{\Delta t}}, \\
  P_{u_2}(i, j) &= \frac{S(i, j)e^{r\Delta t} - S(i + 1, 2j - 1)}{S(i + 1, 2j) - S(i + 1, 2j - 1)}, \\
  P_{d_2}(i, j) &= \frac{S(i + 1, 2j) - S(i, j)e^{r\Delta t}}{S(i + 1, 2j) - S(i + 1, 2j - 1)}.
\end{align*}
\] (11)

The call option value at the last time step is given by

\[ C_{n,j} = \max\{S(n, j) - K, 0\}, \quad j = 1, 2, \ldots \] (12)

The value of the call option at each node can be calculated backwardly by

\[
C_{i,j} = e^{-r\Delta t} \begin{cases} 
  P_{u_1}(i, j)C_{i+1,3j} + P_{m_1}(i, j)C_{i+1,3j-1} + P_{d_1}(i, j)C_{i+1,3j-2}, & i \in \text{odd}, \\
  P_{u_2}(i, j)C_{i+1,2j} + P_{d_2}(i, j)C_{i+1,2j-1}, & i \in \text{even},
\end{cases}
\] (13)

where \( C_{i,j} \) denotes the option value at nodes \((i, j)\).

With the above equations in place, we are ready to construct the alternative tree for local volatility. The procedure contains four parts: (a) analyzing the relations between binomial tree pricing model and time steps, (b) analyzing the relations between trinomial tree pricing model and time steps, (c) determining time steps \( n \), and (d) deriving the option values with equation (13).

\subsection*{3.2. Risk neutrality and no-arbitrage constraints.}

In this section, we describe the risk neutrality and no-arbitrage constraints which ensure the feasibility of the alternative tree pricing model. In order to preserve the transition probabilities defined in equations (8) and (11) to be valid, they should satisfy the risk-neutrality constraints. For the trinomial tree, we get

\[
g_1(i, j) = S(i + 1, 3j - 1)S(i + 1, 3j - 2) + e^{(2r + \sigma^2_1)\Delta t}S(i, j)^2 - [S(i + 1, 3j - 1) + S(i + 1, 3j - 2)]S(i, j)e^{r\Delta t} \geq 0.
\] (14)

\[
g_2(i, j) = [S(i + 1, 3j - 1) + S(i + 1, 3j - 2)] [S(i + 1, 3j) - S(i, j)e^{r\Delta t}] + e^{(2r + \sigma^2_1)\Delta t}S(i, j)^2 - S(i + 1, 3j)^2 \geq 0.
\] (15)

\[
g_3(i, j) = S(i + 1, 3j)S(i + 1, 3j - 1) + e^{(2r + \sigma^2_1)\Delta t}S(i, j)^2 - [S(i + 1, 3j - 1) + S(i + 1, 3j - 2)]S(i, j)e^{r\Delta t} \geq 0.
\] (16)

\[
g_4(i, j) = [S(i + 1, 3j - 1) + S(i + 1, 3j)] [S(i + 1, 3j - 2) - S(i, j)e^{r\Delta t}] + e^{(2r + \sigma^2_1)\Delta t}S(i, j)^2 - S(i + 1, 3j - 2)^2 \geq 0.
\] (17)

\[
g_5(i, j) = S(i + 1, 3j)S(i + 1, 3j - 2) + e^{(2r + \sigma^2_1)\Delta t}S(i, j)^2 - [S(i + 1, 3j) + S(i + 1, 3j - 2)]S(i, j)e^{r\Delta t} \geq 0.
\] (18)

Similarly, we have the following risk-neutrality constraints for the binomial tree

\[
g_6(i, j) = S(i, j)e^{r\Delta t} - S(i + 1, 2j - 1) \geq 0.
\] (19)

\[
g_7(i, j) = S(i + 1, 2j) - S(i, j)e^{r\Delta t} \geq 0.
\] (20)
In addition, to avoid profitable opportunities for arbitrageurs, option prices should be above lower bounds and below upper bounds. We include the no-arbitrage constraints

\[ g_8(k) = C_{mod}(k) - \max\{S(1,1) - Ke^{-rT}, 0\} \geq 0. \] (21)

\[ g_9(k) = S(1,1) - C_{mod}(k) \geq 0. \] (22)

Where \( C_{mod}(k) \) denotes the model price of the \( k \)th call option, \( k = 1, \ldots, N \).

Also, every value of the underlying asset on the tree should be greater or equal to zero. Thus, we also impose the constraint

\[ g_{10}(i,j) = S(i,j) \geq 0. \] (23)

4. Convergence rates of the alternative tree method. In this section, we prove the convergence rates of the alternative tree method for pricing options. For simplicity in the following, we prefer the short form, omit \((i,j)\) and let \( k \in \{u_1, m_1, d_1\}, l \in \{u_2, d_2\} \). The option price calculated by the alternative tree with \(2 \Delta t\) is

\[ C(t_{2(i-1)}, S_j) = e^{-2rt} \left[ \sum_k \left( P_k \left( \sum_l P_l C(t_{2i}, klS_j) \right) \right) \right], \] (24)

where \( i = 1, 2, \ldots, \frac{n}{2} \).

**Lemma 4.1.** For the alternative tree, we have the following estimations:

\[ \sum_k \left( P_k \left( \sum_l P_l \right) \right) = 1, \] (25)

\[ \sum_k \left( P_k \left( \sum_l P_l (kl - 1) \right) \right) = 2r \Delta t + O(\Delta t)^{3/2}, \] (26)

\[ \sum_k \left( P_k \left( \sum_l P_l (kl - 1)^2 \right) \right) = (\sigma_{2i-1}^2 + \sigma_{2i}^2) \Delta t + O(\Delta t)^{3/2}, \] (27)

\[ \sum_k \left( P_k \left( \sum_l P_l (kl - 1)^3 \right) \right) = O(\Delta t)^{3/2}. \] (28)

**Proof.** (1) Clearly

\[ \sum_k \left( P_k \left( \sum_l P_l \right) \right) = P_{u_1}(P_{u_2} + P_{d_2}) + P_{m_1}(P_{u_2} + P_{d_2}) + P_{d_1}(P_{u_2} + P_{d_2}) = 1. \] (29)
CALIBRATION OF THE LOCAL VOLATILITY

(2) Using the Taylor expansions for equations (8) and (11), we have

\[
\begin{align*}
    u_1 &= 1 + \sigma_i \sqrt{3 \Delta t} + \frac{3}{2} \sigma_i^2 \Delta t + O(\Delta t)^{\frac{3}{2}}, \\
    d_1 &= 1 - \sigma_i \sqrt{3 \Delta t} + \frac{3}{2} \sigma_i^2 \Delta t + O(\Delta t)^{\frac{3}{2}}, \\
    P_{u_1} &= \frac{1}{6} + \left( \frac{r}{2\sigma_i} - \frac{\sigma_i}{4} \right) \sqrt{\Delta t / 3} + O(\Delta t), \\
    P_{m_1} &= \frac{2}{3} + O(\Delta t), \\
    P_{d_1} &= \frac{1}{6} - \left( \frac{r}{2\sigma_i} - \frac{\sigma_i}{4} \right) \sqrt{\Delta t / 3} + O(\Delta t). \\
    u_2 &= 1 + \sigma_i \sqrt{\Delta t} + \frac{1}{2} \sigma_i^2 \Delta t + O(\Delta t)^{\frac{3}{2}}, \\
    d_2 &= 1 - \sigma_i \sqrt{\Delta t} + \frac{1}{2} \sigma_i^2 \Delta t + O(\Delta t)^{\frac{3}{2}}, \\
    P_{u_2} &= \frac{1}{2} + \left( \frac{r}{2\sigma_i} - \frac{\sigma_i}{4} \right) \sqrt{\Delta t} + O(\Delta t), \\
    P_{d_2} &= \frac{1}{2} - \left( \frac{r}{2\sigma_i} - \frac{\sigma_i}{4} \right) \sqrt{\Delta t} + O(\Delta t).
\end{align*}
\]

Hence

\[
\sum_k \left( P_k \left( \sum_l P_l (kl - 1) \right) \right) \\
= P_{u_1} P_{u_2}(u_1 u_2 - 1) + P_{u_1} P_{d_2}(u_1 d_2 - 1) + P_{m_1} P_{u_2}(m_1 u_2 - 1) \\
+ P_{m_1} P_{d_2}(m_1 d_2 - 1) + P_{d_1} P_{u_2}(d_1 u_2 - 1) + P_{d_1} P_{d_2}(d_1 d_2 - 1) \\
= 2r \Delta t + O(\Delta t)^{\frac{3}{2}}.
\]

This establishes (26). Similarly, we have (27) and (28).

Lemma 4.2. The remainder term is estimated by \(O(\Delta t)^{\frac{3}{2}}\).

Proof. First define the local remainder at asset price \(S_j\) and time \(t_{2(i-1)}\) by

\[
R_{i-1} = C(t_{2(i-1)}, S_j) - e^{-2r\Delta t} \left[ \sum_k \left( P_k \left( \sum_l P_l C(t_{2i}, klS_j) \right) \right) \right].
\]

where \(C(t_{2(i-1)}, S_j)\) is the exact value of the call option at node \(S_j\) and time \(t_{2(i-1)}\).

Applying the Taylor theorem to \(C(t_{2(i-1)}, S_j)\) at time \(t_{2i}\), we have

\[
C(t_{2(i-1)}, S_j) = C(t_{2i}, S_j) - \frac{\partial C(t_{2i}, S_j)}{\partial t} (2 \Delta t) + O(\Delta t)^2.
\]

(33)
Taking the Taylor expansion for \( C(t_{2i}, u_1 u_2 S_j), C(t_{2i}, u_1 d_2 S_j), C(t_{2i}, m_1 u_2 S_j), C(t_{2i}, m_1 d_2 S_j), C(t_{2i}, d_1 u_2 S_j) \) and \( C(t_{2i}, d_1 d_2 S_j) \) at \( S_j \), and inserting into expression of (32) gives

\[
R_{i-1} = C(t_{2i}, S_j) - \frac{\partial C(t_{2i}, S_j)}{\partial t} 2 \Delta t + O(\Delta t)^2
\]

\[
\begin{align*}
- e^{2r \Delta t} \left[ \sum_k \left( P_k \left( \sum_l P_l \right) \right) C(t_{2i}, S_j) + \frac{\partial C(t_{2i}, S_j)}{\partial S} S_j \sum_k \left( P_k \left( \sum_l P_l (kl - 1) \right) \right) \right] \\
+ \frac{1}{2} \frac{\partial^2 C(t_{2i}, S_j)}{\partial S^2} S_j^2 \sum_k \left( P_k \left( \sum_l P_l (kl - 1)^2 \right) \right) \\
+ \frac{1}{6} \frac{\partial^3 C(t_{2i}, S_j)}{\partial S^3} S_j^3 \sum_k \left( P_k \left( \sum_l P_l (kl - 1)^3 \right) \right) + O\left( \sum \left( \sum (kl - 1)^4 \right) \right)
\end{align*}
\]

(34)

Consequently, applying Lemma 4.1 and using the following expansions

\[
e^{-2r \Delta t} = 1 - 2r \Delta t + O(\Delta t)^2,
\]

\[
\sum_k \left( \sum (kl - 1)^4 \right) = O(\Delta t)^2,
\]

we deduce the following formula which is similar to Lemma 3.2 in [21],

\[
R_{i-1} = C(t_{2i}, S_j) - \frac{\partial C(t_{2i}, S_j)}{\partial t} (2 \Delta t) + O(\Delta t)^2
\]

\[
\begin{align*}
- \left[ 1 - 2r \Delta t + O(\Delta t)^2 \right] C(t_{2i}, S_j) + \frac{\partial C(t_{2i}, S_j)}{\partial S} S_j \left( 2r \Delta t + O(\Delta t)^2 \right) \\
+ \frac{1}{2} \frac{\partial^2 C(t_{2i}, S_j)}{\partial S^2} S_j^2 \left( (\sigma^2_{2i-1} + \sigma^2_{2i}) \Delta t + O(\Delta t)^2 \right) \\
+ \frac{1}{6} \frac{\partial^3 C(t_{2i}, S_j)}{\partial S^3} S_j^3 O(\Delta t)^2 + O(\Delta t)^2
\end{align*}
\]

\[
= - \left[ 2 \frac{\partial C(t_{2i}, S_j)}{\partial t} + 2r S_j \frac{\partial C(t_{2i}, S_j)}{\partial S} + \frac{1}{2} \left( \sigma^2_{2i-1} + \sigma^2_{2i} \right) S_j^2 \frac{\partial^2 C(t_{2i}, S_j)}{\partial S^2} \\
- 2r C(t_{2i}, S_j) \right] \Delta t + O(\Delta t)^2.
\]

(36)

Since the assumption 2 and equation (1), when \( \Delta t \to 0 \), the first term of (36) is zero, we obtain that \( R_{i-1} = O(\Delta t)^2 \). The proof is completed. \( \square \)

**Theorem 4.3.** Let \( \varepsilon(S_j) = C(t_{2i}, S_j) - C_{2i,j} \) be the error of the alternative tree method at node \( S_j \) and time \( t_{2i} \), and define the infinity norm \( \| \varepsilon^i \|_{\infty} = \max_j |\varepsilon^i(S_j)| \).

Then the convergence rates of the alternative tree method (13) are estimated by

\[
\| \varepsilon^i \|_{\infty} = O(\Delta t)^{\frac{3}{2}}.
\]

(37)

**Proof.** According to Lemma 4.2, we have

\[
C(t_{2i}, S_j) = R_i + e^{-2r \Delta t} \left[ \sum_k \left( P_k \left( \sum_l P_l C(t_{2i+2}, klS_j) \right) \right) \right].
\]

(38)

Then

\[
\varepsilon = R_i + e^{-2r \Delta t} \left[ \sum_k \left( P_k \left( \sum_l P_l \varepsilon^i_{l+1} \right) \right) \right].
\]

(39)
According to the infinity norm, we obtain
\[ \| \varepsilon_i \|_\infty \leq O(\triangle t)^{\frac{1}{2}} + e^{-2r\Delta t} \| \varepsilon_{i+1} \|_\infty \]
\[ = O(\Delta t)^{\frac{1}{2}} + (1 - 2r \Delta t + O(\Delta t)^2) \| \varepsilon_{i+1} \|_\infty \]
\[ \leq O(\Delta t)^{\frac{1}{2}} + (1 + O(\Delta t)^2) \| \varepsilon_{i+1} \|_\infty \]
\[ \leq (1 + O(\Delta t)^2)^{\frac{3}{2}} - i \| \varepsilon_{\frac{n}{2}} \|_\infty + O(\Delta t)^{\frac{3}{2}} \left[ 1 - (1 + O(\Delta t)^2)^{\frac{3}{2}} - i \sum_{m=0}^{\frac{n}{2} - i} C_m^{\frac{n}{2} - i} (O(\Delta t)^2)^m \right]. \]

Since \( \| \varepsilon_{\frac{n}{2}} \|_\infty = 0 \) at the final time step \( t_{2i} = T \), we deduce
\[ \| \varepsilon_{i} \|_\infty \leq O(\Delta t)^{-\frac{1}{2}} \left[ -1 + (1 + O(\Delta t)^2)^{\frac{3}{2}} - i \sum_{m=0}^{\frac{n}{2} - i} C_m^{\frac{n}{2} - i} (O(\Delta t)^2)^m \right]. \]

(41)

**Remark 1.** In our alternative tree, we will select the steps of binomial tree. However, in the proof of Theorem 4.3, we do not consider the selection, the proof is more generalized. In fact, the convergence rates of our alternative tree will be faster than \( O(\Delta t)^{\frac{1}{2}} \).

5. **The optimization algorithm for the alternative tree.** This section describes how to calibrate the local volatility by using the alternative tree method. For the calibration problem, the optimization approach is used. In this paper, we employ the quasi-Newton algorithm.

5.1. **Optimization model.** The objective of the calibration is to minimize the least squares error between the market prices and the theoretical prices with respect to the underlying asset values at each node of the alternative tree, subject to constraints of risk neutrality and no-arbitrage. Let \( C_{mkt}(k) \) and \( C_{mod}(x,k) \), \( k = 1, ..., N \) be the market and model price respectively of the \( k \)th call option. The vector \( x \) contains the variables of the model that are the values of the underlying asset at each node of the tree, excluding its current value. Thus, we have the optimization problem
\[ \min_x \frac{1}{2} \sum_{k=1}^{N} (C_{mod}(x,k) - C_{mkt}(k))^2 \]
subject to the constraints
\[ g_1(i,j) \geq 0, \ i = 1, 3, ..., j = 1, ..., 6^{\frac{i-1}{2}}. \]
\[ g_2(i,j) \geq 0, \ i = 1, 3, ..., j = 1, ..., 6^{\frac{i-1}{2}}. \]
\[ g_3(i,j) \geq 0, \ i = 1, 3, ..., j = 1, ..., 6^{\frac{i-1}{2}}. \]
\[ g_4(i,j) \geq 0, \ i = 1, 3, ..., j = 1, ..., 6^{\frac{i-1}{2}}. \]
\[ g_5(i,j) \geq 0, \ i = 1, 3, ..., j = 1, ..., 6^{\frac{i-1}{2}}. \]
\[ g_6(i,j) \geq 0, \ i = 2, 4, ..., j = 1, ..., \frac{1}{2}6^{\frac{j}{2}}. \]
\[ g_7(i,j) \geq 0, \ i = 2, 4, ..., j = 1, ..., \frac{1}{2}6^{\frac{j}{2}}. \]
\[ g_8(k) \geq 0, \ k = 1, ..., N. \]
\[ g_9(k) \geq 0, \ k = 1, ..., N. \]
\[ g_{10}(i,j) \geq 0, \ i = 2, ..., n. \] (43)

The problem is a nonlinear optimization problem with constraints. We adopt an exterior penalty method to transform the nonlinear constrained optimization problem into a nonlinear unconstrained optimization problem. The exterior penalty objective function is

\[ P(x, \alpha) = \frac{1}{2} P_1(x) + \frac{\alpha}{2} (P_2(x) + P_3(x) + P_4(x) + P_5(x)) \] (44)

where

\[ P_1(x) = \sum_{k=1}^{N} (C_{mod}(x,k) - C_{mkt}(k))^2, \] (45)

\[ P_2(x) = \sum_{i \in \text{odd}} \sum_{j} \left( [\min(g_1(i,j),0)]^2 + [\min(g_2(i,j),0)]^2 + [\min(g_3(i,j),0)]^2 \right. \]
\[ \left. + [\min(g_4(i,j),0)]^2 + [\min(g_5(i,j),0)]^2 \right), \] (46)

\[ P_3(x) = \sum_{i \in \text{even}} \sum_{j} \left( [\min(g_6(i,j),0)]^2 + [\min(g_7(i,j),0)]^2 \right), \] (47)

\[ P_4(x) = \sum_{k=1}^{N} [\min(g_8(k),0)]^2 + [\min(g_9(k),0)]^2, \] (48)

\[ P_5(x) = \sum_{i \geq 2} \sum_{j} [\min(g_{10}(i,j),0)]^2. \] (49)

\[ P_2(x), P_3(x), P_4(x), P_5(x) \] are penalty terms which give a positive contribution if and only if \( x \) infeasible. \( \alpha > 0 \) is a penalty factor which reflects the degree of penalty. For the optimization, we use a quasi-Newton algorithm. Specifically we use the BFGS formula.
5.2. Quasi-Newton method. For the implementation of the optimization method, we need to compute the gradient of cost function $\nabla_x C(x, \alpha)$. Let $C_{1,1}(k)$ replace $C_{mod}(x, k)$, then we need to calculate $\frac{\partial C_{1,1}(k)}{\partial S(i, j)}$. For notational simplicity in the following, we assume that we have only one call option. Define the vector $S_{i,j}^{(t)} = (S(i + 1, 3j - 2), S(i + 1, 3j - 1), S(i + 1, 3j))$ for odd $i$, and define the vector $S_{i,j}^{(t)} = (S(i + 1, 2j - 1), S(i + 1, 2j))$ for even $i$. The procedure is given as follows.

(1) Represent $\sigma_i$ by the price of the underlying asset. According to $S(i + 1, 3j) = S(i, j)u_1(i, j)$, we have

$$\sigma_i^2 \triangle t = \frac{1}{3} \left( \ln \frac{S(i + 1, 3j)}{S(i, j)} \right)^2, \quad i \in \text{odd}. \quad (50)$$

According to $S(i + 1, 2j) = S(i, j)u_2(i, j)$, we have

$$\sigma_i^2 \triangle t = \left( \ln \frac{S(i + 1, 2j)}{S(i, j)} \right)^2, \quad i \in \text{even}. \quad (51)$$

(2) Compute the partial derivatives of the risk neutral transition probabilities. For odd $i$

$$\nabla_{S_{i,j}^{(t)}} P_{u_1}(i, j) = \begin{pmatrix} \frac{\partial P_{u_1}(i, j)}{\partial S(i + 1, 3j - 2)} & \frac{\partial P_{u_1}(i, j)}{\partial S(i + 1, 3j - 1)} & \frac{\partial P_{u_1}(i, j)}{\partial S(i + 1, 3j)} \end{pmatrix}^T$$

$$= D \begin{pmatrix} (S(i, j)e^{r\triangle t} - S(i + 1, 3j - 1) - P_{u_1}(i, j)[S(i + 1, 3j) - S(i + 1, 3j - 1)]) \\ (S(i, j)e^{r\triangle t} - S(i + 1, 3j - 2) - P_{u_1}(i, j)[S(i + 1, 3j) - S(i + 1, 3j - 2)]) \\ - P_{u_1}(i, j)[S(i + 1, 3j - 1) - 2S(i + 1, 3j) + S(i + 1, 3j - 2)] \end{pmatrix} \quad (52)$$

$$\nabla_{S_{i,j}^{(t)}} P_{d_1}(i, j) = \begin{pmatrix} \frac{\partial P_{d_1}(i, j)}{\partial S(i + 1, 3j - 2)} & \frac{\partial P_{d_1}(i, j)}{\partial S(i + 1, 3j - 1)} & \frac{\partial P_{d_1}(i, j)}{\partial S(i + 1, 3j)} \end{pmatrix}^T$$

$$= E \begin{pmatrix} (S(i, j)e^{r\triangle t} - S(i + 1, 3j) - P_{d_1}(i, j)[S(i + 1, 3j) - S(i + 1, 3j)]) \\ (S(i, j)e^{r\triangle t} - S(i + 1, 3j - 1) - P_{d_1}(i, j)[S(i + 1, 3j) - S(i + 1, 3j)]) \\ - P_{d_1}(i, j)[S(i + 1, 3j - 1) - 2S(i + 1, 3j) + S(i + 1, 3j - 2)] \end{pmatrix} \quad (53)$$

where

$$D = \frac{1}{[S(i + 1, 3j - 2) - S(i + 1, 3j)][S(i + 1, 3j) - S(i + 1, 3j - 1)]}, \quad (54)$$

$$E = \frac{1}{[S(i + 1, 3j - 2) - S(i + 1, 3j)][S(i + 1, 3j - 1) - S(i + 1, 3j - 2)]}. \quad (55)$$

Similarly, for even $i$

$$\nabla_{S_{i,j}^{(t)}} P_{u_2}(i, j) = \frac{1}{S(i + 1, 2j) - S(i + 1, 2j - 1)} \begin{pmatrix} - P_{d_2}(i, j) \\ - P_{u_2}(i, j) \end{pmatrix} \quad (56)$$

(3) Compute the partial derivatives of $C_{i,j}$. For odd $i$,

$$\nabla_{S_{i,j}^{(t)}} C_{i,j} = e^{-r\triangle t} \left( F_{i,j} \nabla_{S_{i,j}^{(t)}} P_{u_1}(i, j) + G_{i,j} \nabla_{S_{i,j}^{(t)}} P_{d_1}(i, j) \right)$$

$$+ e^{r\triangle t} \begin{pmatrix} P_{d_1}(i, j) \Delta_{i+1,3j-2} \\ P_{u_1}(i, j) \Delta_{i+1,3j-1} \end{pmatrix} \quad (57)$$
where $F_{i,j} = C_{i+1,3j} - C_{i+1,3j-1}$, $G_{i,j} = C_{i+1,3j-2} - C_{i+1,3j-1}$, and

$$\Delta_{i,j} = \frac{C_{i+1,3j} - C_{i+1,3j-2}}{S(i+1,3j) - S(i+1,3j-2)}.$$  

(58)

For even $i$,

$$\nabla_{s_{i,j}} C_{i,j} = e^{-r\Delta t} \left( P_{d_2}(i,j)(\Delta_{i+1,2j-1} - \Delta_{i,j}) \right)$$  

(59)

where $\Delta_{i,j} = \frac{C_{i+1,2j} - C_{i+1,2j-1}}{S(i+1,2j) - S(i+1,2j-1)}$.

(4) Compute the partial derivatives of $C_{n,j}$ with respect to $S(n,j)$.

Since $C_{n,j} = \max(S(n,j) - K, 0)$ is non-differentiable at $S(n,j) = K$. We propose the following smoothing approximation to $C_{n,j}$

$$C_{z,n,j} = \begin{cases} 0, & \frac{S(n,j)}{K} \leq 1 - \frac{z}{2}, \\ \frac{1}{2z} \left( \frac{S(n,j)}{K} - 1 \right) + \frac{z}{2}, & 1 - \frac{z}{2} < \frac{S(n,j)}{K} < 1 + \frac{z}{2}, \\ \frac{S(n,j)}{K} - 1, & \frac{S(n,j)}{K} \geq 1 + \frac{z}{2}. \end{cases}$$  

(60)

where $z$ is a small positive constant, for example 0.01. Then

$$\frac{\partial C_{z,n,j}}{\partial S(n,j)} = \begin{cases} 0, & \frac{S(n,j)}{K} \leq 1 - \frac{z}{2}, \\ \frac{1}{z} \left( \frac{S(n,j)}{K} - 1 \right) + \frac{z}{2}, & 1 - \frac{z}{2} < \frac{S(n,j)}{K} < 1 + \frac{z}{2}, \\ 1, & \frac{S(n,j)}{K} \geq 1 + \frac{z}{2}. \end{cases}$$  

(61)

(5) Compute the partial derivatives $\frac{\partial C_{k+1,i,j}}{\partial S(i,j)}$, $i \geq 3$.

$$\frac{\partial C_{1,1}}{\partial S(i,j)} = \prod P \frac{\partial C_{k-1,i,j}}{\partial S(i,j)} e^{-(i-2)r\Delta t},$$  

(62)

where $P$ denote the probabilities on the path from node $(1,1)$ to node $(i-1,k)$, and for even $i$

$$j = \begin{cases} \frac{j}{3}, & j \mod 3 = 0, \\ \frac{j+2}{3}, & j \mod 3 = 1, \\ \frac{j+1}{3}, & j \mod 3 = 2. \end{cases}$$  

(63)

for odd $i$

$$j = \begin{cases} \frac{j}{2}, & j \mod 2 = 0, \\ \frac{j+1}{2}, & j \mod 2 = 1. \end{cases}$$  

(64)

For example

$$\frac{\partial C_{1,1}}{\partial S(4,6)} = P_{d_1}(1,1)P_{d_2}(2,1) \frac{\partial C_{3,2}}{\partial S(4,6)} e^{-2r\Delta t},$$  

(65)

$$\frac{\partial C_{1,1}}{\partial S(5,3)} = P_{d_1}(1,1)P_{d_2}(2,1)P_{m_1}(3,1) \frac{\partial C_{4,2}}{\partial S(5,3)} e^{-3r\Delta t}.$$  

(66)
In addition, we can easily get the partial derivatives of constraints. In a word, \( \nabla_x P(x, \alpha) \) can be calculated by the above procedure. The quasi-Newton algorithm can be described as follows.

**Algorithm 5.1 (BFGS Quasi-Newton Method)**

1. Choose a function \( \sigma^0(t) \), which will be the initial approximation to the true volatility \( \sigma_{ex}(t) \), that is, an initial guess \( x_0 \).
2. Give the initialization value \( S_0, K, r, T, n, \) tolerance \( \varepsilon \), initial matrix \( H_0 = I \), penalty factor \( \alpha_0 \), increasing coefficient \( \rho \). Let \( k = 0 \).
3. Determine \( C_{mod} \) and \( C_{mkt} \) by the alternative tree method and the BS formula respectively.
   
   4. Compute \( g^k = \nabla P(x_k, \alpha_k) \).
   5. Let \( p^k = -H_k g^k \).
   6. Let \( x_{k+1} = x_k + s_k p^k \), where \( s_k \) satisfies \( P(x_k + s_k p^k, \alpha_k) = \min_s P(x_k + s p^k, \alpha_k) \).
   7. If \( \|\sigma^{k+1} - \sigma^{k}\|_2 < \varepsilon \), then let \( \sigma^t = \sigma^{k+1} \), and end the computation; otherwise, go to the next step.
   8. Let \( \alpha_{k+1} = \rho \alpha_k \), \( g^{k+1} = \nabla P(x_{k+1}, \alpha_{k+1}) \), \( y^k = g^{k+1} - g^k \), \( z^k = x_{k+1} - x_k \); then
   
   \[
   H_{k+1} = H_k + \frac{1}{(z^k)^T y^k} (\beta z^k (y^k)^T - H_k y^k (z^k)^T - z^k (y^k)^T H_k) \]

   where \( \beta = 1 + \frac{(y^k)^T H_k y^k}{(z^k)^T y^k} \). Set \( k = k + 1 \), and go to step (5).

6. **Numerical experiments and market tests.**

   6.1. **Numerical experiments.** In this section, we present the numerical results. To demonstrate the efficiency of our algorithm, we first carry out tests with exact volatility. Suppose the exact volatility is \( \sigma_{ex}(t) \) with which we solve the BS equation to get the option prices. We then treat the prices as the market prices \( C_{mkt} \) which are used to calibrate volatility.

   The parameter values are given as follows

   \[
   S_0 = 50, K = 50, T = 0.3, r = 0.05, \varepsilon = 10^{-8}, \alpha_0 = \min_{1 \leq u \leq m} \alpha_u^0, \quad (68)
   \]

   \[
   \alpha_u^0 = \frac{0.02}{mg_u(x_0) \frac{1}{T} \sum_{k=1}^{N} (C_{mod}(x_0, k) - C_{mkt}(k))^2}, \quad \rho = 10, \quad (69)
   \]

   where \( m \) is the total number of constraints, \( g_u(x_0) \) represents the value of constraints at the initial \( x_0, u = 1, 2, ..., m \).

   First we assume the true volatility function \( \sigma_{ex}(t) \) is defined as

   \[
   \sigma_{ex}(t) = 0.2 + \frac{\cos(10t)}{100}, \quad t \in [0, 0.3]. \quad (70)
   \]

   According to Figure 3, we select binomial tree with odd steps, then combine the binomial tree with trinomial tree to construct a new alternative tree. Next, we choose the binomial tree and the trinomial tree with 3 time steps, then construct the new alternative tree with 6 time steps, that is \( n = 7 \). Figure 4(a) displays the true volatility function which is strictly decreasing. We solve the optimal volatility by Algorithm 5.1, Figure 4(b) shows the comparison between the true volatility \( \sigma_{ex}(t) \) and the estimated \( \sigma(t) \) for \( n = 7 \). It can be observed that the estimated values coincide with the exact value perfectly.
To investigate the stability of our algorithm, we add white noise to the observed data with two forms:

1. \( C^*_mkt = C_{mkt} + \delta \xi \);
2. \( C^*_mkt = (1 + \frac{\xi}{\delta})C_{mkt} \);

where \( \xi \) is a random variable drawn from the standardized normal distribution, \( \delta = 1\%, 2\%, 5\% \), \( \delta_1 = 100, 50, 20 \). Then we use \( C^*_mkt \) for the calibration. The results, presented in Figure 5, verify the stability of our algorithm.

Now we consider the true volatility function \( \sigma_{ex}(t) \) as follows

\[
\sigma_{ex}(t) = 0.2 + \frac{\sin(5t)}{100}, t \in [0, 0.3].
\]  

(71)

Figure 6 shows the comparison of the exact value and the estimated value with alternative tree, trinomial tree and CRR tree for \( n = 7 \). The comparison between the true volatility \( \sigma_{ex}(t) \) and the estimated \( \sigma(t) \) by alternative tree is shown in
Figure 6(a). We can see that our algorithm still performs very well. Figure 6(b) and Figure 6(c) show the calibrated results for trinomial tree and CRR tree. Comparing the two results, we can see that the non-recombining trinomial tree is better than the CRR tree. This is because the non-recombining trinomial tree has more degrees of freedom. But the calculation efficiency is lower than CRR tree for more degrees. Figure 6(d) compares the calibration results of the alternative tree, trinomial tree and CRR tree. We can see from Figure 6(d) that our method is better than trinomial tree and CRR tree. In addition, the calculation efficiency of our method is higher than trinomial tree.

**Figure 4.** Volatility function $\sigma_{ex}(t)$ and volatility estimation for $n = 7$

![Volatility function graph](image)

**Figure 5.** Stability analysis of the algorithm

![Stability analysis graphs](image)
In the above optimization procedure, we use the quadratic penalty method. Now we consider the linear penalty method for the optimization problem with constraints. Assume $\psi(s) = \max(-s, 0)$, which is a nonnegative function. Then the exterior penalty objective function is

$$P(x, \alpha) = \frac{1}{2} P_1(x) + \alpha \sum_{u=1}^{m} \psi(g_u(x)),$$  \hspace{1cm} (72)

where $g_u(x)$ is the constraint of the optimization problem. Considering the first numerical example, we can obtain the calibrated volatility which is shown in Figure 7. From Figure 7, we can see that the alternative tree model is effective for linear and quadratic penalty method.
6.2. Market test. In this section we test the performance of our algorithm with data from the S&P 500 index. Using the market data provided in [3], we build the implied alternative tree and calibrate the volatility. The initial index is $S_0 = 1306.17$, the risk free rate is $r = 0.0559$. We select two fixed strike price $K_{S_0} = 100\%, 110\%$, and construct the implied alternative tree with $n = 7$. For every strike price, we can get a set of discrete points of volatility. Figure 8 shows the interpolated local volatility and calibrated volatility which is a function of time for $K_{S_0} = 100\%, 110\%$. The result indicates that our algorithm is effective.

7. Conclusion. In this paper, we introduce a new alternative tree for local volatility model which combine the traditional binomial tree and trinomial tree, discuss its risk neutrality and no-arbitrage constraints, and prove the convergence rates. Then we use the implied alternative trees to calibrate the local volatility. The problem under consideration is an optimization problem with nonlinear constraints. We further adopt a penalty method and use quasi-Newton algorithm for the optimization. The numerical results show that the alternative implied tree is numerically accurate.
and reduce the computations. The tree reconstructs the local volatility that generates the implied volatility well. Moreover, market test is presented to demonstrate the efficiency of our model. The results obtained strongly support our approach.

REFERENCES

[1] J. Ahn and M. Song, Convergence of the trinomial tree method for pricing European/American options, *Appl. Math. Comput.*, **189** (2007), 575–582.

[2] K. Amin, On the computation of continuous time option prices using discrete approximations, *Journal of Financial and Quantitative Analysis*, **26** (1991), 477–495.

[3] L. Andersen and J. Andreasen, Jump-Diffusion processes: Volatility smile fitting and numerical methods for option pricing, *Rev. Derivatives Res.*, **4** (2000), 231–262.

[4] K. Atkinson, *An Introduction to Numerical Analysis*, 2nd edition, John Wiley & Sons, New York, 1989.

[5] S. Barle and N. Cakici, How to grow a smiling tree, *J. Financ. Eng.*, **7** (1999), 127–146.

[6] F. Black and M. Scholes, The pricing of options and corporate liabilities, *J. Polit. Econ.*, **81** (1973), 637–654.

[7] P. P. Boyle, Option valuation using a three-jump process, *Int. Options J.*, **3** (1986), 7–12.

[8] D. M. Chance, A synthesis of binomial option pricing models for lognormally distributed asset, *J. Appl. Finance*, **3** (2008), 7–24.

[9] L. B. Chang and K. Palmer, Smooth convergence in the binomial model, *Finance and Stochastics*, **11** (2007), 91–105.

[10] C. Charalambous, N. Christofides, E. Constantinide and S. Martzoukos, Implied non-recombining trees and calibration for the volatility smile, *Quant. Finance*, **7** (2007), 459–472.

[11] J. C. Cox, S. A. Ross and M. Rubinstein, Option pricing: A simplified approach, *J. Financ. Econ.*, **7** (1979), 229–263.

[12] S. Crépey, Calibration of the local volatility in a trinomial tree using Tikhonov regularization, *Inverse Problems*, **19** (2003), 91–127.

[13] T. S. Dai and Y. D. Lyuu, The Bino-Trinomial tree: A simple model for efficient and accurate option pricing, *J. Deriv.*, (2010), 7–24.

[14] E. Derman, I. Kani and N. Chriss, Implied trinomial trees of the volatility smile, *J. Deriv.*, **3** (1996), 7–22.

[15] F. Diener and M. Diener, Asymptotics of the price oscillations of a European call option in a tree model, *Math. Finance*, **14** (2004), 271–293.

[16] B. Dupire, Pricing with a smile, *Risk*, **7** (1994), 18–20.

[17] W. X. Gong and Z. L. Xu, Non-recombining trinomial tree pricing model and calibration for the volatility smile, *J. Inverse Ill-Posed Probl.*, **27** (2019), 353–366.

[18] D. P. J. Leisen and M. Reimer, Binomial models for option value-examining and improving convergence, *Appl. Math. Finance*, **3** (1996), 319–346.

[19] Y. Li, A new algorithm for constructing implied binomial trees: Does the implied model fit any volatility smile?, *J. Comput. Finance*, **4** (2001), 69–98.

[20] U. H. Lok and Y. D. Lyuu, The waterline tree for separable local-volatility models, *Comput. Math. Appl.*, **73** (2017), 537–559.

[21] J. T. Ma and T. F. Zhu, Convergence rates of trinomial tree methods for option pricing under regime-switching models, *Appl. Math. Lett.*, **39** (2015), 13–18.

[22] J. Rendleman, J. Richard and B. J. Bartter, Two-state option pricing, *J. Finance*, **34** (1979), 1093–1110.

[23] K. Talias, *Implied Binomial Trees and Genetic Algorithms*, Ph.D thesis, Imperial College, 2005.

[24] J. B. Walsh, The rate of convergence of the binomial tree scheme, *Finance Stoch.*, **7** (2003), 337–361.

Received March 2019; revised July 2020.

E-mail address: gwx0807@126.com
E-mail address: xuzl@ruc.edu.cn