An Efficient and Accurate Lattice for Pricing Derivatives under a Jump-Diffusion Process

Abstract

Derivatives are popular financial instruments whose values depend on other more fundamental financial assets (called the underlying assets). As they play essential roles in financial markets, evaluating them efficiently and accurately is critical. Most derivatives have no simple valuation formulas; as a result, they must be priced by numerical methods such as lattice methods. In a lattice, the prices of the derivatives converge to theoretical values when the number of time steps increases. Unfortunately, the nonlinearity error introduced by the nonlinearity of the option value function may cause the pricing results to converge slowly or even oscillate significantly. The lognormal diffusion process, which has been widely used to model the underlying asset's price dynamics, does not capture the empirical findings satisfactorily. Therefore, many alternative processes have been proposed, and a very popular one is the jump-diffusion process. This paper proposes an accurate and efficient lattice for the jump-diffusion process. Our lattice is accurate because its structure can suit the derivatives' specifications so that the pricing results converge smoothly. To our knowledge, no other lattices for the jump-diffusion process have successfully solved the oscillation problem. In addition, the time complexity of our lattice is lower than those of existing lattice methods by at least half an order. Numerous numerical calculations confirm the superior performance of our lattice to existing methods in terms of accuracy, speed, and generality.
Keywords: complexity, jump-diffusion process, pricing algorithm.

1 Introduction

Derivative securities are financial instruments whose values depend on other more fundamental financial assets called underlying assets such as stocks, indexes, currencies, commodities, bonds, mortgages, other derivatives, and temperatures [19]. The most common derivatives are futures, swaps, and options. They are essential for speculation and the management of financial risk. With the rapid growth and the deregulation of financial markets, many complex derivatives have been structured to meet specific financial goals. But, although financial innovations make the market more efficient, they create new problems in pricing and hedging.

In this paper, the underlying asset is assumed to be stock and the derivative is assumed to be option for convenience. By assuming that the price process of the stock follows

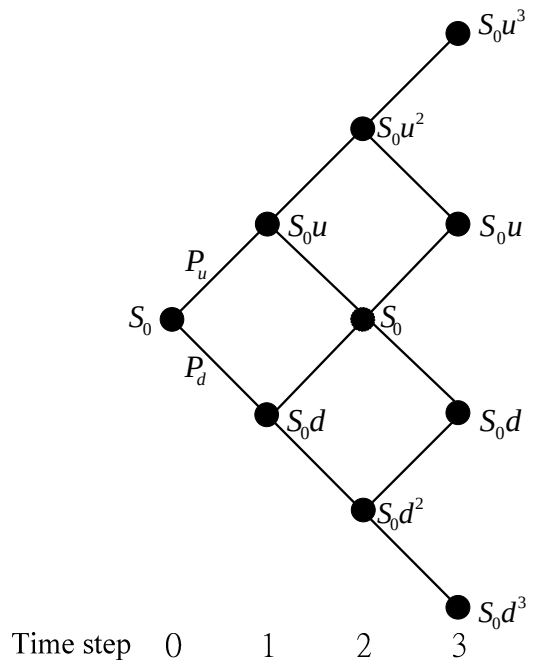

Figure 1: **The CRR Lattice.** The initial stock price is S_0. The upward and downward multiplicative factors for the stock price are u and d, respectively. The branching probabilities are P_u and $P_d = 1 - P_u$.

a lognormal diffusion process, Black and Scholes arrive at their ground-breaking option pricing formula [4]. However, most financial derivatives have no analytical formulas and thus must be priced by numerical methods like lattices. A lattice divides the time span from now to the option's maturity date into n time steps and specifies the stock prices at each time step. Take a 3-time-step Cox-Ross-Rubinstein (CRR) [7] lattice in Fig. 1 as an example. (The details of the CRR lattice will be described later.) The time interval between now and the maturity date is evenly divided into 3 time steps. The stock price at time step 0 is S_0. From an arbitrary node with stock price S, the CRR lattice stipulates that the stock price after one time step equals Su (the up move) with probability P_u and Sd (the down move) with probability P_d, where $d < u$ and $ud = 1$. The probabilities (P_u and P_d) and the multiplication factors (u and d) are determined by matching the stock process's moments asymptotically. As $n \to \infty$, the pricing results converge to the theoretical price [11].

Developing efficient lattice algorithms for complex derivatives has been widely studied [1, 3, 8, 9, 20, 24]. Although a lattice can be efficient for simple derivatives, evaluating complex derivatives can be computationally intractable as some problems are so-called #P-hard [6]. Another issue is that pricing some complex derivatives like barrier options by a lattice could result in substantial oscillations in prices [5]. Oscillations can be suppressed by tweaking the lattice structure to suit derivatives' specifications [12].

Several empirical studies show that the distribution of stock returns has heavier tails and

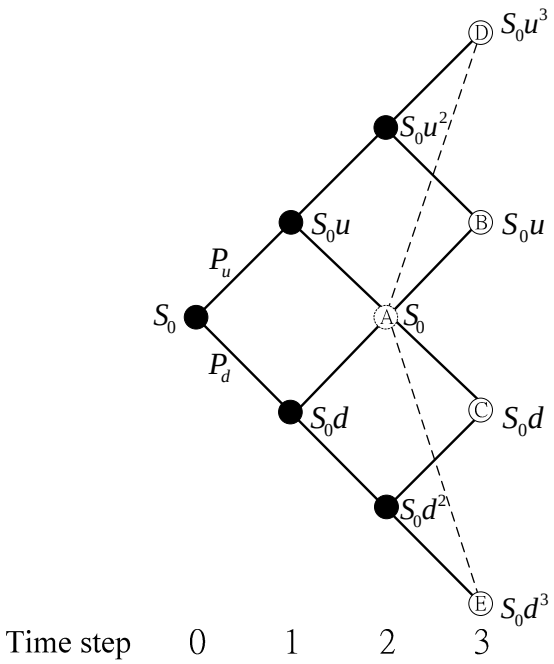

Figure 2: **The Multinomial Structure of Amin's Lattice.** The stock price from node A moves to either node B or C similar to the CRR lattice. It moves to node D or E via the dashed lines when a jump event occurs.

higher peaks than the lognormal diffusion process employed by Black and Scholes [17, 18]. To address this problem, alternative processes such as the jump-diffusion process [23], the GARCH process [10], and the stochastic volatility process [15] have been proposed. Despite the ability to better model the real world stock price, these alternative processes give rise to new problems for lattice methods. For example, a lattice for the GARCH process may still give rise to an exponential-time pricing algorithm [22]. The major contribution of this paper is an efficient and flexible lattice for the jump-diffusion process.

The jump-diffusion process has a jump component superimposed on a diffusion component. The diffusion component is the lognormal diffusion process, and the jump component is composed of lognormal jumps driven by a Poisson process. The lognormal diffusion process, the lognormal jumps, and the Poisson process are assumed to be independent. Since most derivatives have no simple valuation formula under the jump-diffusion process, pricing them by a lattice is one of several alternatives. But lattice algorithms for the jump-diffusion process remain few. Amin uses the multinomial structure to approximate the jump-diffusion process as shown in Fig. 2 [2]. His lattice first constructs a CRR lattice for the lognormal diffusion process. Then it allows each node on the CRR lattice to jump to other nodes at the next time step. For example, the stock price at node A in Fig. 2 can move to either node B or C when no jump occurs, or can jump to nodes D or E when a jump event occurs. However, as mentioned in Hillard and Schwartz's paper, the results are less accurate when the volatility of the lognormal jumps is relatively larger than that of the diffusion compo-

nent [16]. This is due to the fact that only the nodes on the CRR lattice, which are related to the diffusion component, are used to approximate the jump component. To address this problem, Hilliard and Schwartz's (HS) lattice matches the first few local moments of the lognormal jumps to generate accurate pricing results. Towards that end, they position extra nodes in their lattice and allow each node to jump to these extra nodes. Their lattice is sketched in Fig. 3 by assuming that jump events can only occur between time step 0 and time step 1 to simplify the presentation. The white nodes (called the diffusion nodes) and solid branches describe the lognormal diffusion process (i.e., stock process without jump events), while the gray nodes (called the jump nodes) and dashed branches describe the lognormal jumps (i.e., stock process with jump events alone). The jump nodes at time step 1 are positioned to match the first few local moments of the lognormal jumps. The stock price at the diffusion node at time step 0 can either move to the diffusion nodes at time step 1 or jump to the jump nodes at time step 1. Then the nodes at time step 1 follow the CRR structure to move to the diffusion nodes at time step 2. It will be proved that the node count of their lattice grows at a rate of $O(n^3)$. Their lattice lacks the flexibility to suppress the price oscillations.

This paper proposes a new lattice to more efficiently and accurately price a wide range of derivatives under the jump-diffusion setup. We first develop a feasible trinomial structure. We then make a truncated CRR lattice align with the derivatives' specifications and use the trinomial structure to complete the diffusion part of our lattice. Additionally, this trinomial structure is used to connect the jump nodes to the diffusion nodes after one time step to lower the node count of the lattice. Figure 4 describes connecting the jump nodes to the diffusion nodes by the trinomial structures with the same assumption in Fig. 3. The branches of the jump nodes at time step 1 are trinomial, and the jump nodes are connected to the diffusion nodes at time step 2. Therefore, the number of nodes at the second time step and consequently the node count of the whole lattice are smaller than those of the HS lattice. This paper will prove that our lattice grows as $O(n^{2.5})$. In addition, the trinomial structure gives the diffusion part of our lattice more degrees of freedom to address the oscillation issue.

Our paper is organized as follows: The mathematical models and background financial knowledge are introduced in Section 2. We review how to construct a CRR lattice and the HS lattice, and why the pricing results oscillate significantly in Section 3. Section 4 describes a simple trinomial structure that can be easily incorporated into a CRR lattice and shows how this structure is used in our lattice. Section 4 also proves the time complexities of our lattice. Numerical results given in Section 5 show the superior performance of our lattice. Section 6 concludes.

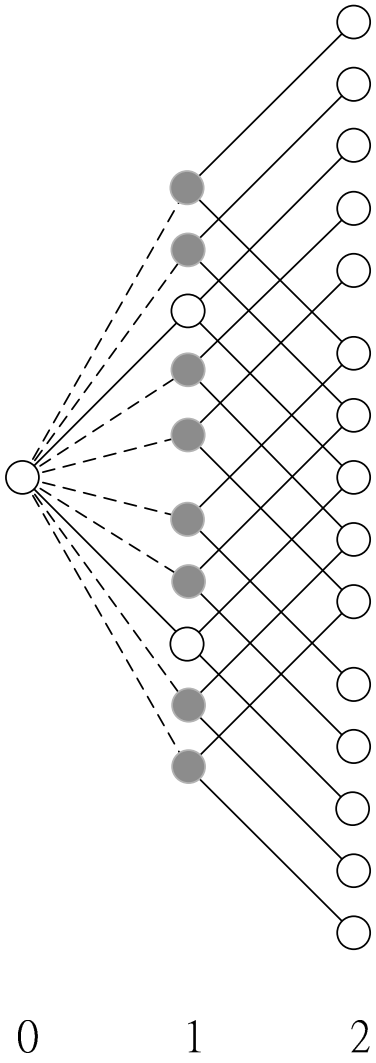

Figure 3: **The Structure of the HS Lattice.** The white nodes and solid branches describe the dynamics of the diffusion process. The gray nodes and dashed branches describe the dynamics of stock price process when a jump occurs. The branches from the gray nodes and white nodes at time step 1 follow the CRR binomial-branch structure as illustrated in Fig. 1.

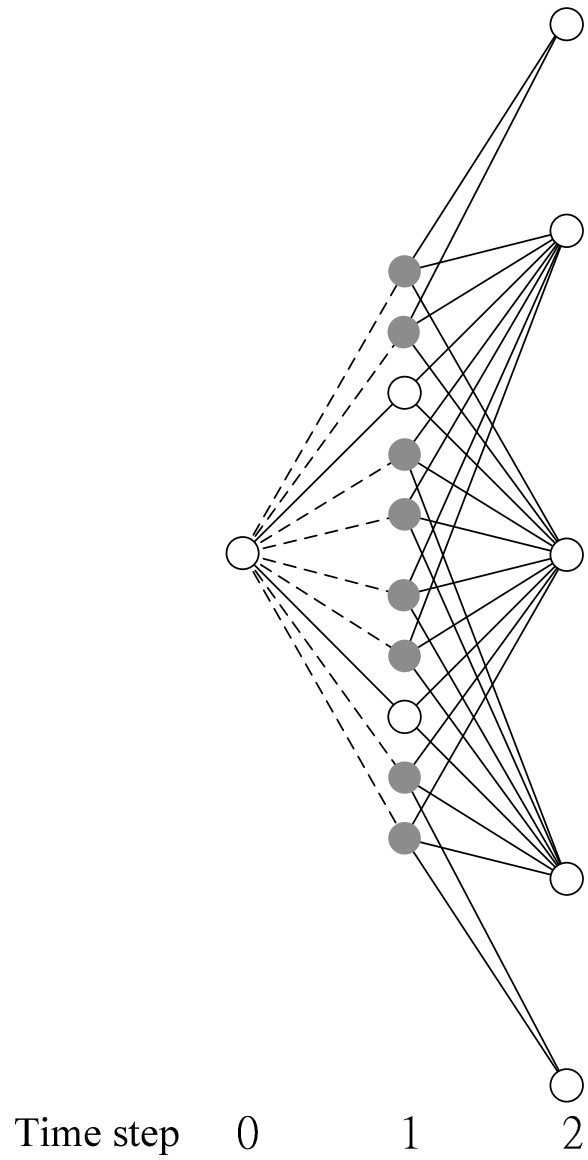

Figure 4: **The Structure of Our Lattice.** The white nodes and solid branches describe the dynamics of the diffusion process. The gray nodes and dashed branches describe the dynamics of the stock price process when a jump occurs. The branches from gray nodes at time step 1 are trinomial.

2 Modeling and Definitions

2.1 The Mathematical Model of the Jump-Diffusion Process

Define S_t as the stock price at time t. The risk-neutralized version of the underlying asset's jump-diffusion process is

$$S_t = S_0 e^{(r-\lambda\bar{k}-0.5\sigma^2)t+\sigma z(t)} Y(n), \tag{1}$$

where $Y(n) = \prod_{i=0}^{n(t)} (1+k_i)$ and $k_0 = 0$ [23]. In the above equation, $z(t)$ denotes a standard Brownian motion, r denotes the risk-free rate, and σ denotes the volatility of the diffusion component of the stock price process. Jump events are governed by the Poisson process $n(t)$ with jump intensity λ. The random jump magnitude k_i $(i > 0)$ satisfies $\ln(1+k_i) \sim N(\gamma, \delta)$, where $\bar{k} \equiv E(k_i) = e^{\gamma+0.5\delta^2} - 1$. Recall that the diffusion component, the random jump magnitude, and the Poisson process are assumed to be independent.

Define the S-log-price of the stock price S' as $\ln(S'/S)$ and the log-distance between stock prices S and S' as $|\ln(S) - \ln(S')|$. Hilliard and Schwartz decompose the S_0-log-price of S_t (called V_t throughout this paper) into the diffusion component and the jump component by rewriting Eq. (1) as follows:

$$V_t \equiv \ln\left(\frac{S_t}{S_0}\right) \equiv X_t + Y_t, \tag{2}$$

where the diffusion component

$$X_t \equiv \left(r - \lambda\bar{k} - 0.5\sigma^2\right) t + \sigma z(t) \tag{3}$$

is an Ito process, and the jump component

$$Y_t \equiv \sum_{i=0}^{n(t)} \ln(1+k_i) \tag{4}$$

is normal under Poisson compounding.

2.2 Background Financial Knowledge

Vanilla options give their owners the rights to buy or sell the underlying stocks for the exercise price X and have no other unusual features. European-style vanilla options allow holders to exercise the options only at the maturity date T with the payoff

$$\mathcal{P}(T) = \max(\theta(S_T - X), 0),$$

where $\theta = 1$ for call options and $\theta = -1$ for put options.

Barrier options are a kind of options whose payoffs depend on whether the underlying stock's price path ever touches certain price levels called the barriers. There are many different types of barrier options. Our paper focuses on the evaluation of the knock-out barrier options. Such options cease to exist when the underlying stock's price touches one

of the barrier(s). The payoff of a single-barrier option with a low barrier L (where $L < S_0$) at the maturity date is

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

where $S_{\inf}(T) = \inf_{0 \le t \le T} S_t$. The payoff of a double-barrier option with a low barrier L and a high barrier H (where $L < S_0 < H$) at the maturity date T is

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

where $S_{\sup}(T) = \sup_{0 \le t \le T} S_t$. A fundamental theorem in finance says that a European option's value at time 0 equals the discounted expected payoff at time T [13], i.e.,

$$e^{-rT} \mathrm{E}\left[\, \mathcal{P}(T) \,\right]. \tag{5}$$

3 Preliminaries

We first review the principles underlying building a lattice that converges to the lognormal diffusion process as the number of time steps n goes to infinity. Next, we explain how to price an option on the lattice. Then we proceed to describe how Hilliard and Schwartz incorporate the jump component into the CRR lattice. Finally, we describe a mechanism to solve the oscillation problem.

3.1 Constructing a Lattice under the LognormalDiffusion Process

The well-known Cox-Ross-Rubinstein (CRR) binomial lattice forms the foundation of our and the HS lattices. A lattice partitions the time span from time 0 to time T into n equal time steps and specifies the stock price at each time step. The length of one time step Δt thus equals T/n. A 3-time-step CRR binomial lattice is illustrated in Fig. 1. At each time step, the stock price S can either make an up move to become Su with probability P_u or a down move to become Sd with probability $P_d \equiv 1 - P_u$. The relation

$$ud = 1 \tag{6}$$

is enforced by the CRR binomial lattice.

As the lognormal diffusion process is a special case of the jump-diffusion process with $\lambda = 0$, the S_0-log-price of $S_{i\Delta t}$ can be represented by X_t (see Eq. (3)) with $\lambda = 0$ under the lognormal diffusion process. The mean (μ) and variance (Var) of the S_t-log-price of $S_{t+\Delta t}$ are derived from Eq. (3) thus:

$$\mu \equiv \left(r - 0.5\sigma^2\right) \Delta t, \tag{7}$$

$$\mathrm{Var} \equiv \sigma^2 \Delta t. \tag{8}$$

To make sure the lattice converges to the lognormal diffusion process X_t, the mean and variance of the lattice should match μ and Var, respectively, asymptotically:

$$P_u \ln u + P_d \ln d \quad \to \quad \mu, \tag{9}$$

$$P_u (\ln u - \mu)^2 + P_d (\ln d - \mu)^2 \quad \to \quad \text{Var}. \tag{10}$$

In addition,

$$P_u + P_d = 1 \tag{11}$$

must hold. The four parameters P_u, P_d, u, and d are uniquely obtained by solving Eqs. (6) and (9)–(11). The CRR lattice adopts this solution: $u = e^{\sigma\sqrt{\Delta t}}$, $d = e^{-\sigma\sqrt{\Delta t}}$, $P_u = \frac{e^{r\Delta t} - d}{u - d}$ and $P_d = \frac{e^{r\Delta t} - u}{d - u}$. The requirements $0 \le P_u, P_d \le 1$ can be met by suitably increasing n [21]. Moreover, with a small enough Δt, Eq. (7) implies

$$-\sigma\sqrt{\Delta t} \le \mu \le \sigma\sqrt{\Delta t}. \tag{12}$$

In general, each node on a lattice can branch into ℓ nodes at the next time step, which will be called an ℓ-nomial lattice. Ideas similar to the above can be applied to construct an ℓ-nomial lattice. $\ell \ge 2$. Note that 2ℓ degrees of freedom are provided by an ℓ-nomial lattice. They include ℓ price multiplicative factors (like the u and d of the CRR binomial lattice) and ℓ branching probabilities (like the P_u and P_d of the CRR binomial lattice). As a result, 2ℓ linearly independent equations are needed to determine these 2ℓ variables uniquely. The matching of the mean and variance provides 2 equations. The branching probabilities sum to 1, providing another one. Note that the branching probabilities must lie between 0 and 1. An additional $2\ell - 3$ equations are added to modify the structure of the lattice or to meet more constraints. For example, Eq. (6) is used in the CRR lattice to have flat price levels (i.e., $S_0 ud = S_0$). In this paper, we construct trinomial structures (i.e., $\ell = 3$) to connect the jump nodes to the diffusion nodes after one time step to reduce the node count of the lattice. We also use the trinomial structures and additional techniques to make a lattice align with the derivatives' specifications to suppress price oscillations.

3.2 Hilliard and Schwartz's Lattice for the Jump-Diffusion Process

In the HS lattice, the diffusion component is modeled by a binomial lattice as in Section 3.1 except that the mean of the S_t-log-price of $S_{t+\Delta t}$ (see Eq. (7)) is replaced by

$$\mu \equiv \left(r - \lambda\bar{k} - 0.5\sigma^2\right) \Delta t \tag{13}$$

to account for the jump contributions. The jump component is modeled by positioning extra jump nodes to match the first few local moments of the lognormal jumps. Motivated by the decomposition in Eq. (2), the S_0-log-prices of the $2 \times (2m + 1)$ nodes at time step $i + 1$ following node $V_{i\Delta t}$ can be represented by

$$V_{(i+1)\Delta t} = V_{i\Delta t} + c\sigma\sqrt{\Delta t} + jh, \qquad c = \pm 1,\ j = 0, \pm 1, \pm 2, \ldots, \pm m. \tag{14}$$

Above, $c \in \{-1, 1\}$ denotes the upward or the downward movement of the stock price driven by the diffusion component (called the diffusion move), and j denotes the number of positions above or below the node $V_{i\Delta t} + c\sigma\sqrt{\Delta t}$ (called the jump move). Note that $j = 0$ corresponds to the diffusion moves. In addition, the magnitude of the basic jump unit is set to

$$h = \sqrt{\gamma^2 + \delta^2}\,. \tag{15}$$

The probabilities q_j ($j = 0, \pm 1, \pm 2, \ldots, \pm m$) are chosen to match the first $2m$ local moments of the continuous-time distribution of the jump component. Formally, we set

$$\sum_{j=-m}^{j=m} (jh)^i q_j = \mu_i' \equiv \mathrm{E}\left[\sum_{w=0}^{n(\Delta t)} \ln(1 + k_w)\right]^i, \qquad i = 1, 2, \ldots, 2m,$$

where μ_i' is the ith local moment of $Y_{\Delta t}$ (see Eq. (4)). In addition,

$$\sum_{j=-m}^{j=m} q_j = 1$$

must hold. By solving the above $2m + 1$ equations, we obtain $2m + 1$ probabilities q_j ($j = 0, \pm 1, \pm 2, \ldots, \pm m$). Since the diffusion component and the jump component are assumed to be independent, the probability of moving from node $V_{i\Delta t}$ to node $V_{i\Delta t} + \sigma\sqrt{\Delta t} + jh$ is $P_u q_j$ and it is $P_d q_j$ to node $V_{i\Delta t} - \sigma\sqrt{\Delta t} + jh$.

Pricing options is done by evaluating Eq. (5) on the lattice by the so-called backward induction. Define $F(V_{i\Delta t}, i)$ as the option value for the case that the stock price is $S_0 e^{V_{i\Delta t}}$ at time step i. In particular, $F(V_{n\Delta t}, n)$ equals $\mathcal{P}(T)$. For European-style options, the value on the node with stock price $S_0 e^{V_{i\Delta t}}$ at time step i can be computed by

$$\begin{aligned} F(V_{i\Delta t}, i) = e^{-r\Delta t} \sum_{j=-m}^{m} \Big[& F\left(V_{i\Delta t} + \sigma\sqrt{\Delta t} + jh, (i+1)\,\Delta t\right) P_u q_j \\ & + F\left(V_{i\Delta t} - \sigma\sqrt{\Delta t} + jh, (i+1)\,\Delta t\right) P_d q_j \Big]. \end{aligned}$$

Note that $F(V_{i\Delta t}, i)$ is set to 0 if $V_{i\Delta t} \le \ln(L/S_0)$ in pricing single-barrier options. In double-barrier options, $F(V_{i\Delta t}, i)$ is set to 0 if $V_{i\Delta t} \le \ln(L/S_0)$ or $V_{i\Delta t} \ge \ln(H/S_0)$. The backward induction for the CRR lattice is a special case by setting $m = 0$ and $q_0 = 1$. Hilliard and Schwartz argue that their lattice is efficient since a small m, says 3 or 4, results in accurate pricing results for vanilla options. Later we will show that it is not accurate for barrier ones.

3.3 Complexity Analysis

In the HS lattice, according to Eq. (14), the stock prices at time step i can be represented by

$$\begin{aligned} S_0 e^{V_{i\Delta t}} &= S_0 e^{V_{(i-1)\Delta t} + c_1\sigma\sqrt{\Delta t} + j_1 h}, \\ &= S_0 e^{V_{(i-2)\Delta t} + (c_1+c_2)\sigma\sqrt{\Delta t} + (j_1+j_2)h}, \\ &= \quad\quad\quad \vdots \\ &= S_0 e^{V_{0\Delta t} + (c_1+c_2+\cdots+c_i)\sigma\sqrt{\Delta t} + (j_1+j_2+\cdots+j_i)h}, \end{aligned}$$

where $c_k = \pm 1$, $j_k = 0, \pm 1, \pm 2, \ldots, \pm m$, and $k \in \{1, 2, \ldots, i\}$. Recall that $V_{0\Delta t} \equiv \ln(S_0/S_0) = 0$. Therefore, the stock prices can be represented by $S_0 e^{\alpha\sigma\sqrt{\Delta t} + \beta h}$, where α and β are integers. At time step i, the value of the variable α ranges from $-i$ to i and that of β ranges from $-mi$ to mi. Note that the diffusion nodes and the jump nodes do not occupy the same grid point because the log-distance between two adjacent diffusion nodes, $2\sigma\sqrt{\Delta t}$, is different from $h = \sqrt{\gamma^2 + \delta^2}$. As a result, the node count of the lattice at time step i equals $(2i+1)(2mi+1) = O(i^2)$; therefore, the total node count is $\Sigma_{i=0}^{n}(2i+1)(2mi+1) = O(n^3)$. This is also the time complexity of pricing vanilla options and barrier options on the HS lattice.

3.4 The Oscillation Problem

Theoretically, the pricing results generated by the lattices converge to theoretical option value as $n \to \infty$ [11]; however, the pricing results may oscillate significantly. Figlewski and Gao argue that this phenomenon is mainly due to the nonlinearity error — the error introduced by the nonlinearity of the option value function [12]. The nonlinearity error can be significantly reduced by making a node or a price level of the lattice coincide with the location where the option value function is highly nonlinear. For example, the oscillation can be suppressed by making a node coincide with the strike price at the maturity date or by making a price level coincide with the barrier for pricing barrier options [25].

4 Lattice Construction

We first explain the trinomial structure. Next, we describe how to use this structure to fit derivatives' specifications and, later, how to use it on the jump nodes to reduce the node count of our lattice. Finally, we analyze the time complexities of the HS and our lattices.

4.1 The Trinomial Structure

We shall construct a valid trinomial structure that can be easily added in front of a truncated CRR lattice which has Δt as the duration of one time step. A trinomial structure is valid if it

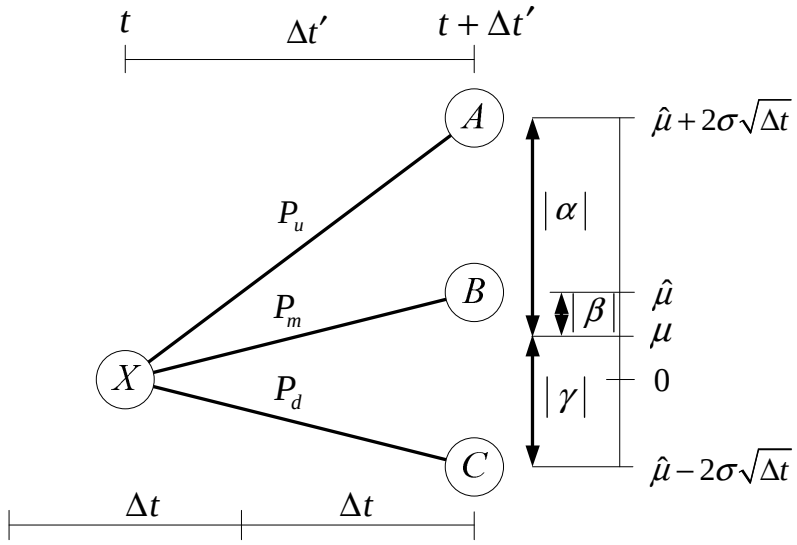

Figure 5: **The Trinomial Structure.** The stock price can move from node X to node A with probability P_u, node B with probability P_m, and node C with probability P_d. Above, μ denotes the mean of the $s(X)$-log-price of $S_{t+\Delta t'}$ (see Eq. (16)). The $s(X)$-log-price of $s(X)$ is 0, and those of $s(B)$, $s(A)$, and $s(C)$ are $\hat{\mu}$, $\hat{\mu}+2\sigma\sqrt{\Delta t}$, and $\hat{\mu}-2\sigma\sqrt{\Delta t}$, respectively. The numbers β, α, and γ are defined in Eqs. (21)–(23). The length of the time step for the trinomial structure, $\Delta t'$, satisfies the constraint $\Delta t \leq \Delta t' < 2\Delta t$.

matches the mean and variance of the stock return; in addition, the branching probabilities P_u, P_m, and P_d are between 0 and 1.

Let the stock price of node Z be $s(Z)$ for convenience. Given a node X at time t and a CRR lattice with the length of each time step equal to Δt beginning at time $t + \Delta t'$, the mean (Eq. (13)) and variance (Eq. (8)) of the $s(X)$-log-price of $S_{t+\Delta t'}$ are adjusted thus:

$$\mu \equiv \left(r - \lambda\bar{k} - 0.5\sigma^2\right)\Delta t', \tag{16}$$

$$\text{Var} \equiv \sigma^2 \Delta t'. \tag{17}$$

In Fig. 5, for example, the log-distance between two adjacent nodes at time $t + \Delta t'$ is $2\sigma\sqrt{\Delta t}$. Let node B be the node whose $s(X)$-log-price ($\hat{\mu}$) is closest to μ among all the nodes at time $t + \Delta t'$. We select two nodes A and C, which are adjacent to node B, and then obtain three branching probabilities P_u, P_m, and P_d through matching the mean and variance of the $s(X)$-log-price of $S_{t+\Delta t'}$. That this procedure works is guaranteed by the next theorem.

Theorem 4.1 *Given a node X at time t and a CRR lattice with the length of each time step equal to Δt beginning at time $t + \Delta t'$, there is a valid trinomial structure from the node X whose $s(X)$-log-price of the central node B lies in the interval $[\mu - \sigma\sqrt{\Delta t}, \mu + \sigma\sqrt{\Delta t})$. Furthermore, the valid branching probabilities can be found by matching the mean and variance of the $s(X)$-log-price of $S_{t+\Delta t'}$.*

Proof:

The branching probabilities for the node X (i.e., P_u, P_m, P_d) can be derived by solving the following three equalities:

$$\begin{aligned} P_u\alpha + P_m\beta + P_d\gamma &= 0, && (18)\\ P_u(\alpha)^2 + P_m(\beta)^2 + P_d(\gamma)^2 &= \text{Var}, && (19)\\ P_u + P_m + P_d &= 1, && (20) \end{aligned}$$

where

$$\begin{aligned} \beta &\equiv \hat{\mu} - \mu, && (21)\\ \alpha &\equiv \hat{\mu} + 2\sigma\sqrt{\Delta t} - \mu = \beta + 2\sigma\sqrt{\Delta t}, && (22)\\ \gamma &\equiv \hat{\mu} - 2\sigma\sqrt{\Delta t} - \mu = \beta - 2\sigma\sqrt{\Delta t}, && (23)\\ \hat{\mu} &\equiv \ln\left(s(B)/s(X)\right). \end{aligned}$$

In the above equations, α, β, and γ are the difference between μ and the $s(X)$-log-prices of $s(A)$, $s(B)$, and $s(C)$, respectively, where node B is bracketed by nodes A and C as in Fig. 5. Equations (18) and (19) match the mean and variance of the $s(X)$-log-price of $S_{t+\Delta t'}$ (see Eqs. (16) and (17)), respectively. Define

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

Cramer's rule applied to Eqs. (18) to (20) gives $P_u = \det_u/\det$, $P_m = \det_m/\det$, and $P_d = \det_d/\det$. Note that $\det < 0$ because $\alpha > \beta > \gamma$. To ensure that the branching probabilities are valid, it suffices to show that P_u, P_m, $P_d \geq 0$. As $\det < 0$, it is sufficient to show $\det_u$, $\det_m$, and $\det_d \leq 0$ instead. Finally, as $\alpha > \beta > \gamma$, it suffices to show that $\beta\gamma + \text{Var} \geq 0$, $\alpha\gamma + \text{Var} \leq 0$, and $\alpha\beta + \text{Var} \geq 0$ under the premise $\beta \in [\,-\sigma\sqrt{\Delta t}, \sigma\sqrt{\Delta t}\,)$. Indeed,

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

as desired. ■

4.2 Fitting the Derivative's Specifications

After the construction of a valid trinomial structure, we next explain how to use it and several techniques to make the lattice align with the derivatives' specifications. Pricing a

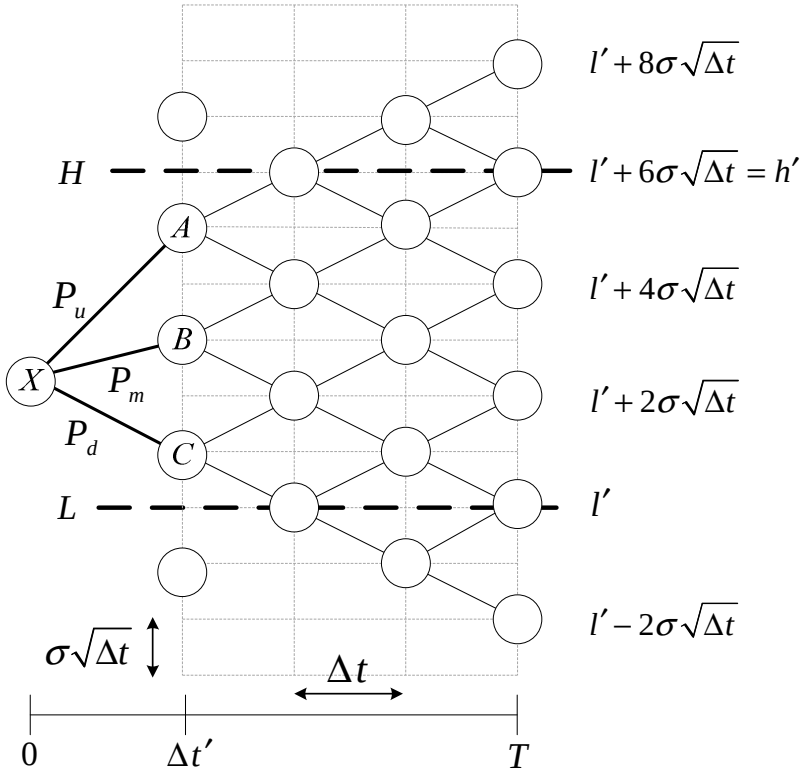

Figure 6: **Fitting Two Barriers.** The truncated CRR lattice is lying on the grid that partitions the plane into cells. The cell width of the grid is Δt and the cell height of the grid is $\sigma\sqrt{\Delta t}$. Nodes A, B, and C are selected for constructing the trinomial branch. P_u, P_m, and P_d denote the branching probabilities from node X, H denotes the high barrier, L denotes the low barrier, and the log-distance between two adjacent nodes at the same time step of the truncated CRR lattice is $2\sigma\sqrt{\Delta t}$. The $s(X)$-log-prices of the nodes at time T are listed next to the nodes.

double-barrier option without regards to the alignment of the barriers causes the oscillation problem mainly because of the nonlinearity error. To suppress it, we truncate a CRR lattice, make two price levels of the truncated CRR lattice coincide with the barriers, and then adjoin a trinomial structure to it. Numerical experiments show that only the diffusion part needs to fit the barriers to yield stable pricing results. For the adjustment of the lattice structure, the following two requirements must be satisfied: (1) the lattice is aligned with the barriers, i.e., the barriers coincide with certain levels of nodes on the lattice; (2) at the first time step, the trinomial structure are valid.

We use Fig. 6 to explain how to make a lattice align with the barriers (i.e., H and L) and how to find the nodes on the CRR lattice to adjoin a trinomial structure. In this figure, the first two time steps of the CRR lattice are truncated. The truncated CRR lattice is laid on the grid that partitions the plane into cells. The cell width of the grid is Δt, and the cell height of the gird is $\sigma\sqrt{\Delta t}$. The truncated CRR lattice emanates from three nodes: A, B, and C at time $\Delta t'$ and They are connected to the root node X at time 0 with branching probabilities P_u, P_m, P_d. We now show how to choose the length of a time step, Δt, to make two levels of nodes on the truncated CRR lattice coincide with H and L. In order to make the two barrier H and L coincide with levels of nodes on the lattice, $\frac{h'-l'}{2\sigma\sqrt{\Delta t}}$ needs to be some integer k, where $h' \equiv \ln(H/s(X))$ and $l' \equiv \ln(L/s(X))$. For example, $k = 3$ in Fig. 6. Although $\Delta\tau \equiv T/n$ is a natural choice for an n-time-step lattice, the problem is $\frac{h'-l'}{2\sigma\sqrt{\Delta\tau}}$ may not be an integer. Therefore, we pick a Δt that is close to, but does not exceed, $\Delta\tau$ and this Δt makes $\frac{h'-l'}{2\sigma\sqrt{\Delta t}}$ an integer. This means $\Delta t = (\frac{h'-l'}{2\kappa\sigma})^2$, where $\kappa \equiv \left\lceil \frac{h'-l'}{2\sigma\sqrt{\Delta\tau}} \right\rceil$. The number of time steps is $\lfloor \frac{T}{\Delta t} \rfloor$ because the truncated CRR lattice has $\lfloor \frac{T}{\Delta t} \rfloor - 1$ time steps. As a result, the length of the first time step, $\Delta t'$, is the remaining amount of time to make the whole lattice span T, i.e.,

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

Clearly, $\Delta t \leq \Delta t' < 2\Delta t$.

Now, lay out the grid from barrier L upward. Automatically, barrier H coincides with one level of nodes because of the above condition. Each node at time $\Delta t'$ is represented by

$$\begin{cases} l' + 2j\sigma\sqrt{\Delta t}, & \text{if the truncated CRR lattice has an even number of time steps,} \\ l' + (2j+1)\sigma\sqrt{\Delta t}, & \text{otherwise,} \end{cases}$$

for some integer j. For example, the truncated CRR lattice in Fig. 6 has 3 time steps, so the nodes at time $\Delta t'$ are represented by $l' + (2j+1)\sigma\sqrt{\Delta t}$.

To create a valid trinomial structure from the root node X, Theorem 4.1 says that at time $\Delta t'$, there exists a node B whose corresponding $s(X)$-log-price is in the interval $[\,\mu - \sigma\sqrt{\Delta t}, \mu + \sigma\sqrt{\Delta t}\,)$; moreover, we can also obtain two adjacent nodes, A and C, and three valid branching probabilities P_u, P_m, P_d from node X.

By means of a similar strategy mentioned above, we can also modify the lattice for a single-barrier option with lower barrier L. There is no need to adjust the length of the

time step (Δt) because only one barrier (L) needs to coincide with a level of nodes on the truncated CRR lattice. (Recall that the reason we needed to adjust Δt for a double-barrier option is because $\frac{h'-l'}{2\sigma\sqrt{\Delta t}}$ may not be an integer without the adjustment.) So we can simply set $\Delta t = T/n$. Then we lay out the grid form barrier L upward and obtain the trinomial structure from root node according to Theorem 4.1.

4.3 Jump Nodes

The node count of our lattice can be significantly reduced by connecting the jump nodes to the diffusion nodes after one time step via the aforementioned trinomial structure. In our lattice, each time step is decomposed into two phases: the diffusion phase (for the diffusion component Eq. (3)) and the jump phase (for the jump component Eq. (4)). This decomposition is feasible due to the fact that the diffusion and the jump components are assumed to be independent. Define $d(\ell)$ as the number of diffusion nodes at time step ℓ; the total number of nodes (including the diffusion and the jump nodes) at time step ℓ is $(2m+1)d(\ell)$. Our proposed lattice construction method will make the total number of nodes at each time step grows at a milder rate than that of the HS lattice.

Figure 7 illustrates our two-phase lattice construction procedure. In the diffusion phase, a diffusion node X at time step $\ell - 1$ with stock price S will move upward to Su with probability P_u and downward to Sd with probability P_d. In the ensuing jump phase, the node with stock price Su (like node A) can jump to Sue^{jh}, and the node with stock price Sd (like node B) can jump to Sde^{jh}, where $-m \le j \le m$ and h is from Eq. (15). Note that at $j = 0$, no jump event occurs and the stock price is unchanged (like nodes a and b). Then, in the diffusion phase of the next time step, we construct valid trinomial structures (dashed lines) for the jump nodes (like the gray nodes at time step ℓ) by Theorem 4.1 and connect these jump nodes to the appropriate diffusion nodes (like nodes C, D, E, F, and G). The diffusion nodes at time step ℓ (like nodes a and b) follow the CRR lattice structure.

4.4 Complexity Analysis

We now prove that the node count of our lattice is $O(n^{2.5})$ by showing that the number of nodes at time step ℓ, $(2m+1)d(\ell)$, grows at a milder rate of $O(n^{1.5})$. In Fig. 8, assume that node d is the highest diffusion node at time step ℓ. Then the log-distance between node X, the highest jump nodes at time step ℓ, and node d is mh. Note that the trinomial structure from node X is determined by Theorem 4.1 (recall Fig. 5). Thus, the log-distance between node X and its middle successor node (node B in both Figs. 5 and 8) is $\mu + \beta$. Moreover, the log-distance between node X's two successor nodes A and B is $2\sigma\sqrt{\Delta t}$. Thus, the log-distance between node A and node Y, which is connected by the upper branch of node d, is $mh + \mu + \beta + 2\sigma\sqrt{\Delta t} - \sigma\sqrt{\Delta t} \le mh + 3\sigma\sqrt{\Delta t}$ because both μ and β are at most $\sigma\sqrt{\Delta t}$ (see Theorem 4.1 and Eq. (12)). Therefore, the number of extra diffusion nodes (the white nodes above Y) is at most $\left\lceil \frac{mh+3\sigma\sqrt{\Delta t}}{2\sigma\sqrt{\Delta t}} \right\rceil$. Similarly, the number of extra diffusion nodes

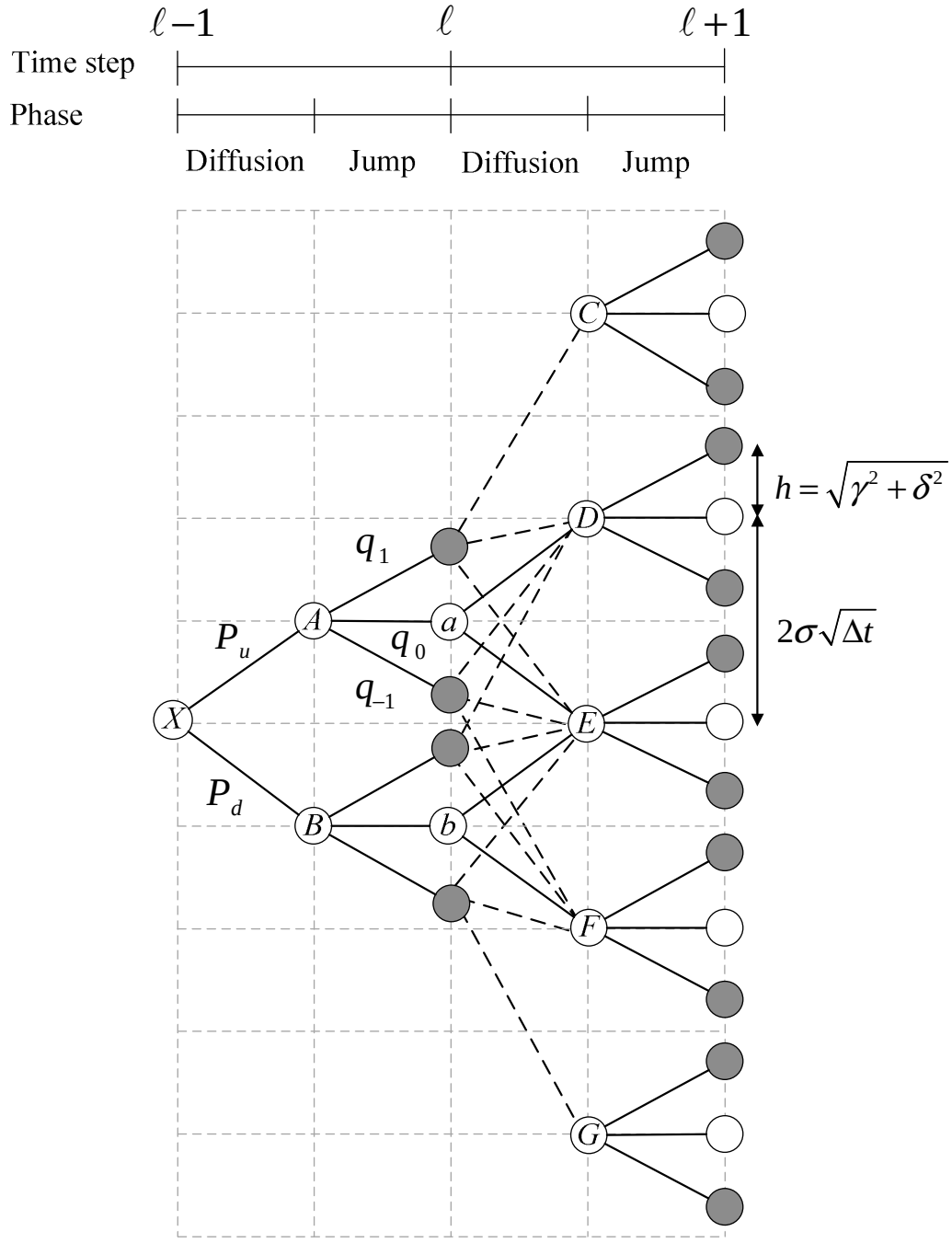

Figure 7: **Two-Phase Lattice Construction.** The white nodes and gray nodes represent the diffusion nodes and the jump nodes, respectively. Each time step Δt is divided into two phases: diffusion and jump. The dashed lines represent the trinomial structures that are used to connect the jump nodes to the diffusion nodes after one time step, and m is set to one in this figure for simplicity.

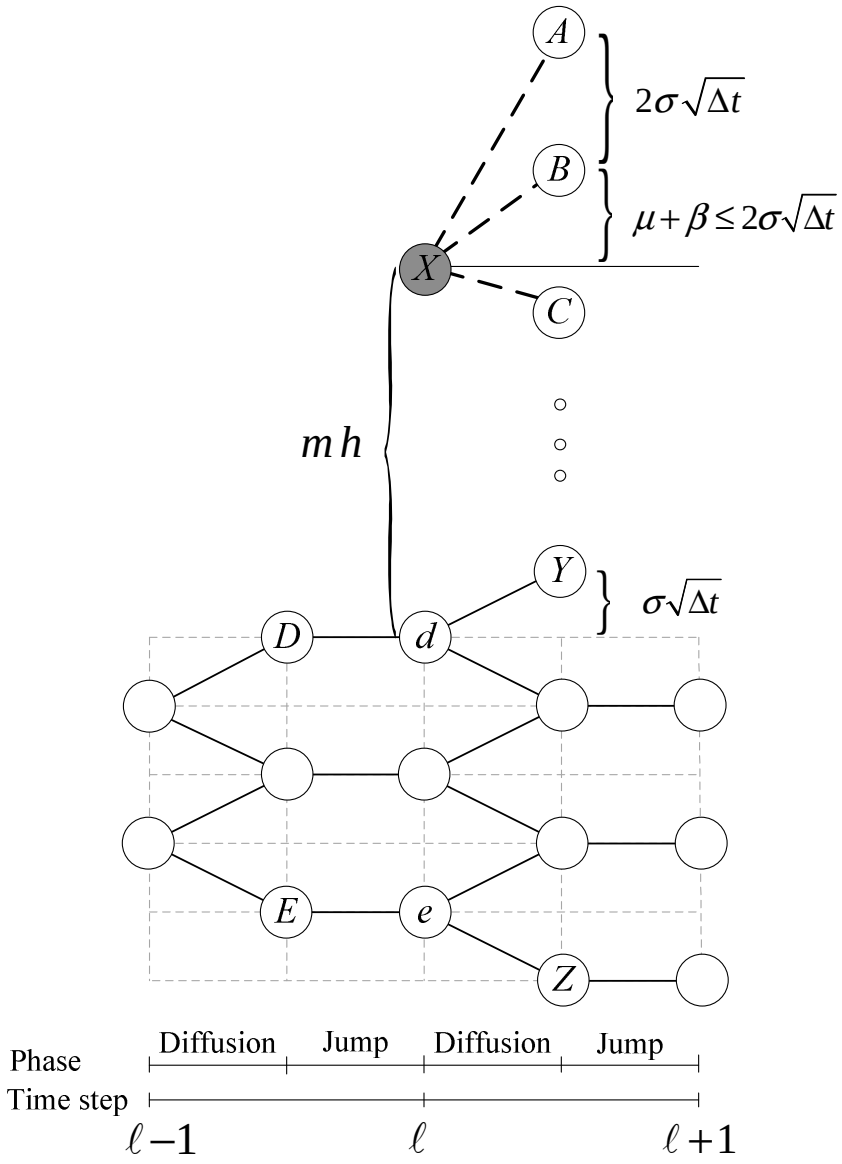

Figure 8: **The Node Count of Our Lattice.** Node X represents the highest jump node from node D. The log-distance between two adjacent diffusion nodes is $2\sigma\sqrt{\Delta t}$, m is the number of the jump nodes on each side of a diffusion node, $h = \sqrt{\gamma^2 + \delta^2}$, and the definitions of μ and β can be found in Fig. 5.

below node Z, which is connected by the lower branch of the lowest diffusion node e at time step ℓ, is at most $\left\lceil \frac{mh+3\sigma\sqrt{\Delta t}}{2\sigma\sqrt{\Delta t}} \right\rceil$. As a result, $d(\ell)$ satisfies the following recurrence relation:

$$\begin{aligned} d(\ell+1) &= d(\ell) + 2 \times \left\lceil \frac{mh}{2\sigma\sqrt{\frac{T}{n}}} + 1.5 \right\rceil + 1, \\ &= d(\ell-1) + 2 \times 2 \times \left\lceil \frac{mh}{2\sigma\sqrt{\frac{T}{n}}} + 1.5 \right\rceil + 2, \\ &= \quad \vdots \\ &= d(1) + 2 \times \ell \times \left\lceil \frac{mh}{2\sigma\sqrt{\frac{T}{n}}} + 1.5 \right\rceil + \ell, \\ &= O(n^{1.5}), \end{aligned}$$

because $d(1) = 2$, $d(0) = 1$, $\Delta t = T/n$ and $\ell \leq n$. Consequently, the total node count is $(2m+1)\sum_{\ell=0}^{n} d(\ell) = O(n^{2.5})$. This is also the time complexity of pricing vanilla options and barrier options on our lattice.

5 Numerical Results

This section evaluates the numerical performance of our lattice. First, we compare the time complexity of our lattice with that of the HS lattice. After that, numerical results for vanilla options and barrier options are analyzed. For the experimental settings, the first 6 local moments of the lognormal jumps, which approximate the stochastic price jumps, are matched in both our lattice as the HS lattice. The parameters for the vanilla options are from Hilliard and Schwartz [16]. The experiments are obtained by running programs on a PC with the Intel Pentium D 2.8GHz CPU.

5.1 Time Complexity

We first compare the time complexity of our lattice with that of the HS lattice. As confirmed in Fig. 9, the time complexity of our lattice is $O(n^{2.5})$, whereas that of the HS lattice is $O(n^3)$. This efficiency of our lattice is mainly due to the fact that the node count of our lattice is reduced by connecting the jump nodes to the diffusion nodes via the trinomial structure. In contrast, the HS lattice connects the jump nodes to the diffusion nodes via the CRR lattice structure.

5.2 Vanilla Option

Table 1 tabulates the prices of European put options under different strike prices and relative jump volatilities (i.e., δ/σ). It compares the European put option prices computed by our

	European puts			
Strike	Merton	Ours	HS	Amin
Panel A. $\gamma = -0.025$, $\delta = \sqrt{0.05}$, and $\sigma = \sqrt{0.05}$				
30	2.621	2.626	2.621	2.623
35	4.412	4.416	4.414	4.415
40	6.696	6.698	6.698	6.701
45	9.422	9.422	9.427	9.426
50	12.524	12.520	12.526	12.528
Panel B. $\gamma = -0.045$, $\delta = 0.3$, and $\sigma = 0.1$				
30	3.918	3.920	3.915	3.894
35	5.982	5.994	5.993	5.972
40	8.458	8.460	8.465	8.461
45	11.302	11.287	11.299	11.318
50	14.460	14.466	14.483	14.487
Panel C. $\gamma = -0.025$, $\delta = \sqrt{0.05}$, and $\sigma = 0.05$				
30	2.172	2.194	2.189	1.837
35	3.810	3.793	3.788	3.553
40	5.980	6.002	6.004	5.783
45	8.650	8.630	8.638	8.501
50	11.756	11.773	11.787	11.646
	MAE	**0.010**	0.011	0.076
	RMSE	**0.013**	0.015	0.130

Table 1: **Comparisons under Different Relative Jump Volatilities in Pricing European Puts.** The benchmarks are generated by Merton's model. The parameters are $S = 40$, $r = 0.08$, $\lambda = 5$, and $T = 1$ (year). The relative jump volatility is defined as δ/σ. All lattices are implemented with 200 time steps.

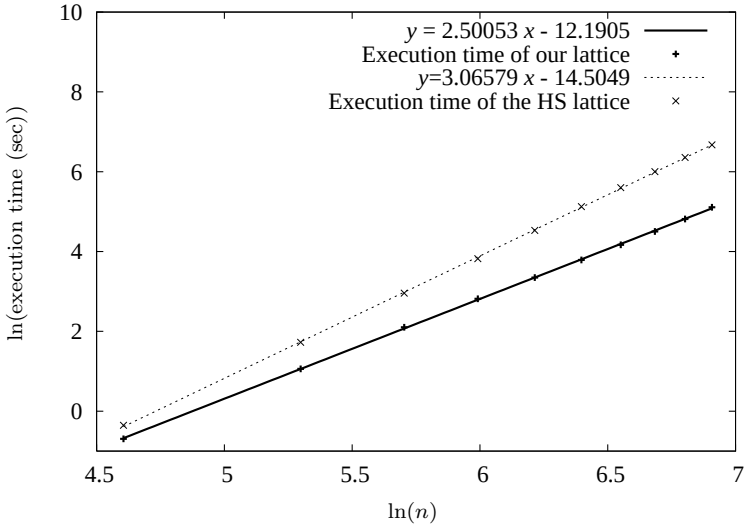

Figure 9: **Time complexity.** The points represent the log-log plot of the time steps versus the execution time. The lines represent the regression lines of these points. The time complexity of our lattice is $O(n^{2.5})$ and that of the HS lattice is $O(n^3)$. The order of the time complexities can be obtained by observing the coefficients of the variable x of the regression lines. The parameters are $S = 40$, $X = 40$, $r = 0.08$, $\lambda = 5$, $\gamma = -0.025$, $\delta = \sqrt{0.05}$, $\sigma = \sqrt{0.05}$, and $T = 1$ (year).

lattice, the HS lattice, Amin's lattice with Merton's model as the benchmark. In Panels A and B of the table, the prices generated by these three lattice algorithms are close to the benchmarks. However, in Panel C, which lists the results for large relative jump volatilities, Amin's lattice gives less accurate prices when the strike price is lower, whereas our lattice and the HS lattice both provide accurate results. Moreover, the mean absolute error (MAE) and root-mean-squared error (RMSE) of our lattice are smaller than those of the other two lattices.

Table 2 lists the results of European put option prices with different maturities: one year, five years, and ten years. Observe that the MAE and RMSE of the HS lattice are larger than those of our lattice. With increasing maturities, the percentage errors of the HS lattice are also much larger than those of our lattice. These results indicates that our lattice is more accurate and efficient than the other two lattices.

Figure 10 demonstrates the convergence behavior of our lattice in pricing a vanilla option. Since the errors of the option values converge at a rate of $O(1/n)$ [14], we then utilize extrapolation as in Fig. 10. In this figure, the extrapolated result 5.6177 accurately approximates the benchmark value 5.617851, which is generated by Merton's model . Figure 11 compares the convergence of the HS lattice with that of our lattice. It shows that given the same computational time, our lattice obtains more accurate option values than the HS lattice.

	European puts			Percent errors	
Strike	Merton	Ours	HS	Ours	HS
Panel A. Maturity (T) = one year					
30	2.621	2.626	2.622	0.179	0.033
35	4.412	4.416	4.414	0.102	0.054
40	6.696	6.698	6.698	0.036	0.031
45	9.422	9.421	9.427	−0.007	0.051
50	12.524	12.520	12.526	−0.028	0.017
Panel B. Maturity (T) = 5 years					
30	5.618	5.621	5.622	0.056	0.074
35	7.437	7.441	7.442	0.054	0.067
40	9.412	9.416	9.418	0.050	0.068
45	11.519	11.524	11.526	0.046	0.061
50	13.741	13.747	13.749	0.042	0.059
Panel C. Maturity (T) = 10 years					
30	5.283	5.286	5.288	0.042	0.088
35	6.653	6.656	6.659	0.043	0.095
40	8.093	8.096	8.100	0.043	0.090
45	9.593	9.597	9.601	0.042	0.085
50	11.145	11.150	11.154	0.041	0.081
	MAE	**0.004**	0.005		
	RMSE	**0.004**	0.006		

Table 2: **Comparisons under Different Maturities.** The parameters are $S = 40$, $r = 0.08$, $\lambda = 5$, $\gamma = -0.025$, $\delta = \sqrt{0.05}$, and $\sigma = \sqrt{0.05}$. Both our and the HS lattices are implemented with 200 time steps.

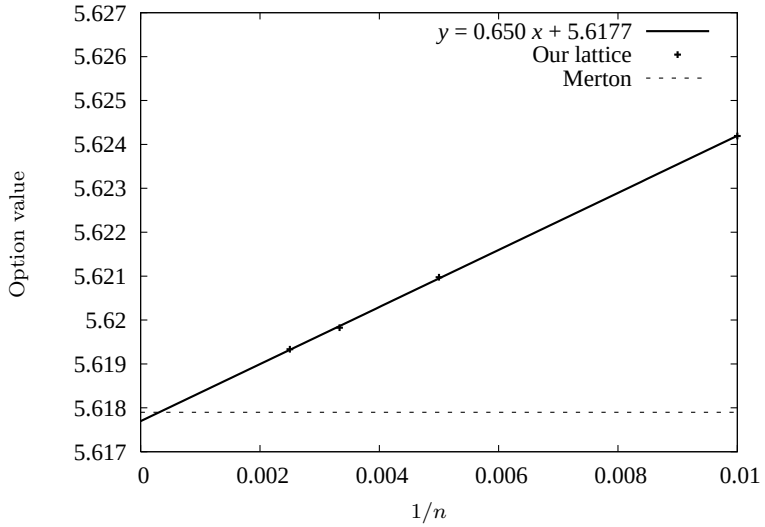

Figure 10: **Convergence Property.** The points and solid line represent the option values at various time steps and the regression line, respectively. The dashed line illustrates the benchmark generated by Merton's model. The parameters are $S = 40$, $X = 30$, $r = 0.08$, $\lambda = 5$, $\gamma = -0.025$, $\delta = \sqrt{0.05}$, $\sigma = \sqrt{0.05}$, and $T = 5$ (years).

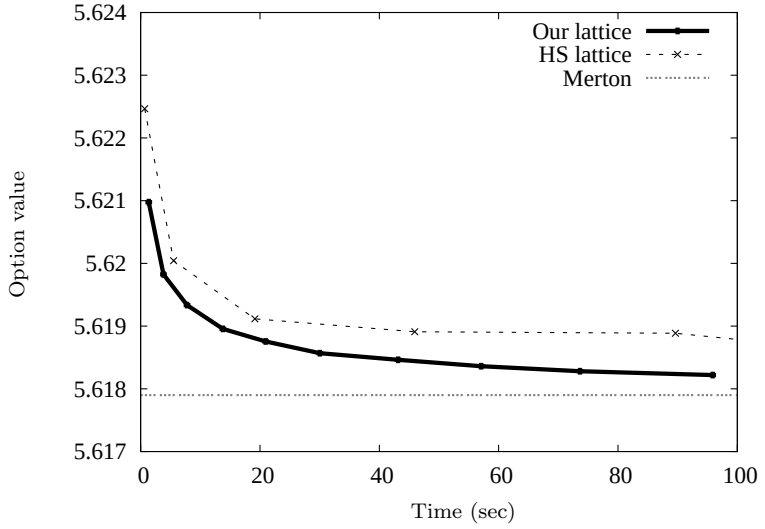

Figure 11: **Convergence Performance of Our and the HS Lattices.** The solid curve and the dashed one trace the option values of our lattice and the HS lattice as functions of the computational time, respectively. The dotted line illustrates the benchmark generated by Merton's model. The parameters are $S = 40$, $X = 30$, $r = 0.08$, $\lambda = 5$, $\gamma = -0.025$, $\delta = \sqrt{0.05}$, $\sigma = \sqrt{0.05}$, and $T = 5$ (years).

	European single-barrier call			Percent errors	
Time steps	Simulation value	Ours	HS	Ours	HS
67		3.887	4.508	−0.079	15.896
268	3.890	3.871	4.473	−0.487	14.987
604	(3.868, 3.912)	3.869	3.951	−0.541	1.563
1075		3.867	4.080	−0.581	4.892

Table 3: **Our and the HS Lattices in Pricing a Single-Barrier Call Option.** Simulation values serve as the benchmarks and are the results of 1,000,000 simulations with 100,000 time steps. The values in the parentheses denote the 95% confident interval of the simulation value. The parameters are $S = 102$, $X = 105$, $L = 100$, $r = 0.05$, $\lambda = 3$, $\gamma = -0.005$, $\delta = 0.1$, $\sigma = 0.1$, and $T = 1$ (year).

5.3 Barrier Options

We next compare our lattice with the HS lattice in pricing barrier options. For both single-barrier and double-barrier options, simulation values serve as the benchmarks and are the results of 1,000,000 simulations with 100,000 time steps.

Table 3 tabulates the numerical results for a European single-barrier call option. In this table, three points are worth mentioning. First, we observe that the percentage errors of our lattice are significantly lower than those of the HS lattice. Second, the option prices of our lattice are more stable than those of the HS lattice, whose errors range from 1.6% to over 15%. Third, the prices generated by the HS lattice are always much higher than the benchmarks. Figure 12 demonstrates that for pricing single-barrier options, the results computed by the HS lattice do not converge smoothly as the number of time steps increases, whereas those by our lattice converge smoothly.

Table 4 lists the numerical results of our lattice and those of the HS lattice for a European double-barrier call option. As shown in this table, the percentage errors of our lattice are again much lower than those of the HS lattice, and the HS lattice always overestimates the prices of the double-barrier option.

According to the above observations, the numerical results for both single- and double-options generated by our lattice are always better than those of the HS lattice. As a result, our lattice is more accurate than the HS lattice in pricing barrier options. The main reason is that our lattice can fit derivatives' specifications to suppress the oscillation problem.

6 Conclusions

This paper proposes a novel, accurate, and efficient lattice to price derivatives whose underlying assets follow the jump-diffusion process. To our best knowledge, this work is not only the first one that deals with the oscillation problem under the jump-diffusion process but also the first attempt to reduce the time complexity of the lattice for the jump-diffusion

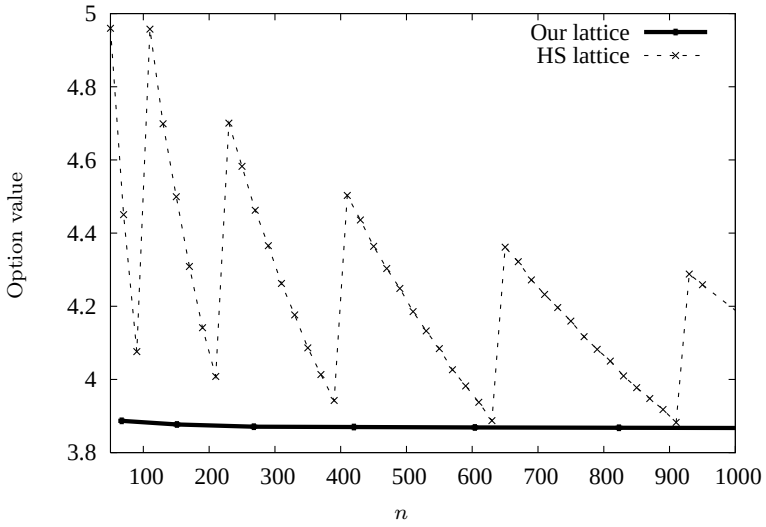

Figure 12: **Comparison of the Oscillation Phenomenon between the HS and Our Lattices.** The numerical settings are the same as those in Table 3. The solid line plots the prices generated by our lattice. The dashed line plots the prices generated by the HS lattice.

	European double-barrier call			**Percent errors**	
Time steps	Simulation value	Ours	HS	Ours	HS
100		2.593	2.653	−0.103	2.211
401	2.595	2.598	2.635	0.097	1.518
701	(2.584, 2.606)	2.599	2.621	0.125	0.989
1002		2.599	2.630	0.146	1.356

Table 4: **Our and the HS Lattices in Pricing a Double-Barrier Call Option.** Simulation values serve as the benchmarks and are the results of 1,000,000 simulations with 100,000 time steps. The values in the parentheses denote the 95% confident interval of the simulation value. The parameters are $S = 90$, $X = 100$, $H = 130$, $L = 70$, $r = 0.05$, $\lambda = 3$, $\gamma = -0.005$, $\delta = 0.1$, $\sigma = 0.1$, and $T = 1$ (year).

process from earlier works of $O(n^3)$ to $O(n^{2.5})$. According to the numerical results, our lattice is confirmed to be superior to the existing methods in terms of accuracy, speed, and generality.

References

[1] D. Aingworth, R. Motwani, and J. D. Oldham. Accurate Approximations for Asian Options. *Proceedings of the eleventh annual ACM-SIAM symposium on Discrete algorithms*, pages 891–900, 2000.

[2] K. I. Amin. Jump Diffusion Option Valuation in Discrete Time. *Journal of Finance*, 48(5):1833–1863, 1993.

[3] J. Barraquand and T. Pudet. Pricing of American Path-dependent Contingent Claims. *Mathematical Finance*, 6(1):17–51, 1996.

[4] F. Black and M. Scholes. The Pricing of Options and Corporate Liabilities. *Journal of Political Economy*, 81(3):637–654, 1973.

[5] P. P. Boyle and S. H. Lau. Bumping Up Against the Barrier with the Binomial Method. *Journal of Derivatives*, 1(4):6–14, 1994.

[6] P. Chalasani. Approximate Option Pricing. *Algorithmica*, 25(1):2–21, 1999.

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

[8] T.-S. Dai, G.-S. Huang, and Y.-D. Lyuu. An Efficient Convergent Lattice Algorithm for European Asian Options. *Applied Mathematics and Computation*, 169(2):1458–1471, 2005.

[9] T.-S. Dai and Y.-D. Lyuu. An Exact Subexponential-Time Lattice Algorithm for Asian Options. *Acta Informatica*, 44(1):23–39, 2007.

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

[11] D. Duffie. *Dynamic Asset Pricing Theory*. Princeton University Press Princeton, NJ, 2001.

[12] S. Figlewski and B. Gao. The Adaptive Mesh Model: A New Approach to Efficient Option Pricing. *Journal of Financial Economics*, 53(3):313–351, 1999.

[13] J. M. Harrison and S. Pliska. Martingales and Stochastic Integrals in the Theory of Continuous Trading. *Stochastic Processes and Their Applications*, 11(3):215–260, 1981.

[14] S. Heston and G. Zhou. On the Rate of Convergence of Discrete-Time Contingent Claims. *Mathematical Finance*, 10(1):53–75, 2000.

[15] S. L. Heston. A Closed-form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options. *Review of Financial Studies*, 6(2):327–343, 1993.

[16] J. E. Hilliard and A. Schwartz. Pricing European and American Derivatives Under a Jump-Diffusion Process: A Bivariate Tree Approach. *Journal of Financial and Quantitative Analysis*, 40(3):671–691, 2005.

[17] J. R. M. Hosking, G. Bonti, and D. Siegel. Beyond the Lognormal. *Risk*, 13(5):59–62, 2000.

[18] R. Huisman, K. G. Koedijk, and R. A. J. Pownall. VaR-x: Fat Tails in Financial Risk Management. *Journal of Risk*, 1(1):47–61, 1998.

[19] J. Hull. *Options, Futures, and Other Derivatives*. 2003.

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

[21] Y.-D. Lyuu. *Financial Engineering and Computation: Principles, Mathematics, Algorithms*. Cambridge University Press, 2002.

[22] Y.-D. Lyuu and C.-N. Wu. On Accurate and Provably Efficient GARCH Option Pricing Algorithms. *Quantitative Finance*, 5(2):181–198, 2005.

[23] R. C. Merton. Option Pricing When Underlying Stock Returns are Discontinuous. *Journal of Financial Economics*, 3(1–2):125–144, 1976.

[24] P. Ritchken. On Pricing Barrier Options. *Journal of Derivatives*, 3(2):19–28, 1995.

[25] P. Ritchken, L. Sankarasubramanian, and A. M. Vijh. The Valuation of Path Dependent Contracts on the Average. *Management Science*, 39(10):1202–1213, 1993.