Option Pricing for a Jump-Diffusion Model with General Discrete Jump-Size Distributions

Michael C. Fu
Robert H. Smith School of Business and Institute for Systems Research, University of Maryland, College Park, Maryland 20742, mfu@rhsmith.umd.edu

Bingqing Li, Guozhen Li
School of Finance, Nankai University, 300350 Tianjin, China [libq@nankai.edu.cn, jennylea152@gmail.com]

Rongwen Wu
Capital One Financial Corporation, McLean, Virginia 22102, rongwen_wu@hotmail.com

We obtain a closed-form solution for pricing European options under a general jump-diffusion model that can incorporate arbitrary discrete jump-size distributions, including nonparametric distributions such as an empirical distribution. The flexibility in the jump-size distribution allows the model to better capture leptokurtic features found in real-world data. The model uses a discrete-time framework and leads to a pricing formula that is provably convergent to the continuous-time price as the discretization is increased. The solution is easy to implement with fast convergence properties. Numerical results illustrate the efficiency and accuracy of the proposed model and highlight its robustness and flexibility.

Keywords: jump-diffusion process; option pricing; European option; generating function; lattice path

History: Received November 11, 2015; accepted April 4, 2016, by Noah Gans, stochastic models and simulation. Published online in Articles in Advance September 29, 2016.

1. Introduction

Jump-diffusion processes are commonly used to model the price dynamics for the underlying asset in option pricing, because they are able to capture leptokurtic features and volatility smiles observed in real-world data. The jump-diffusion process was first proposed for option pricing by Merton (1976) to address important characteristics observed in actual market data that cannot be captured by the Black–Scholes model (Black and Scholes 1973), such as negative skewness and excess kurtosis. Since the seminal paper of Merton (1976), jump-diffusion processes have become one of the main models in option pricing for capturing jumps in an underlying asset. Surveys of critical developments in the jump-diffusion literature, as well as research challenges, can be found in Eraker et al. (2003), Bates (2003), Broadie and Detemple (2004), and Kou (2008a); see also Schoutens (2003) for more general Lévy processes. Jump-diffusion models complement stochastic volatility models, which are better able to capture volatility clustering effects; combining the two features (jumps and stochastic volatility) leads to more generals models (see Garcia et al. 2010) but at the cost of tractability, as concluded in Kou (2008a, p. 88):

jump-diffusion models attempt to strike a balance between reality and tractability, especially for short maturity options and short term behavior of asset pricing.

Further motivation for our work is summarized in Cai and Kou (2011, p. 2067):

a key question for jump diffusion models is what jump size distribution will be used.

For general jump-size distributions, Merton (1976) provided a European option pricing formula written in the form of an infinite sum of Black–Scholes-type terms, which is independent of the specific jump-size distribution; however, each term involves complex computationally intensive multidimensional integration, which generally depends heavily on the actual jump-size distribution, requiring a case-by-case analysis for each distribution. Merton (1976) considered two special jump-size distributions—a discrete distribution described by Samuelson (1973), where there is a positive probability of immediate ruin, and the log-normal distribution—and provided corresponding pricing formulas. The log-normal assumption makes estimation and hypothesis testing tractable, and it has become the most important representation of the jump-diffusion models (e.g., see Ball and Torous 1983, Jarrow and Rosenfeld 1984, Bates 1991).

However, a large body of empirical studies shows that the log-normal jump-size distribution fits actual returns data rather poorly, primarily because empirical log-return distributions exhibit excess kurtosis and skewness relative to the normal distribution (e.g., see...
Bookstaber and McDonald 1987, Madan and Seneta 1990, Tucker 1992, and references therein). As a result, there is a vast literature on analyzing and developing more accurate models for option pricing. We briefly survey some representative streams.

Ramezani and Zeng (1998) conducted a maximum likelihood estimation on security prices and showed that allowing for a mixture of distributions for the upward jumps and downward jumps proves to be a better fit to the data than having a common distribution spawning both. They proposed an asymmetric jump-size distribution with the upward-jump and downward-jump magnitude following Pareto and Beta distributions, respectively.

Kou (2002) proposed a separate model with jump sizes following a (asymmetric) log-double-exponential distribution, for which he was able to derive closed-form pricing formulas for European options, later extended to path-dependent options such as American options, barrier options, and look-back options (Kou and Wang 2004, Kou 2008b). More recently, the log-double-exponential distribution of jump size has been generalized to log-phase-type-exponential (Asmussen et al. 2004), log-hyperexponential (Cai 2009) and log-mixed-exponential (Cai and Kou 2011) distributions.

As pointed out by Bollerslev and Todorov (2011), the jump-size distributions are distinct for different underlying assets and different sampling frequencies. Kaeck (2013) demonstrated that improved option pricing accuracy comes mainly from a better fit to the jumps, which highlights the importance of the choice of jump-size distribution for option pricing. After investigating seven kinds of alternative jump-size distributions, he concluded that the log-double-Gamma distribution outperforms the other jump-size distributions.

The literature review makes clear that the vast majority of jump-diffusion models have focused on continuous jump-size distributions. The main advantage of continuous distributions is their ability to be described by a small set of parameters. However, one could argue that the effect of jumps is more critical when they are large in magnitude, and these often occur with a relative infrequency that is not well captured by continuous distributions and might be best represented, for example, by an empirical distribution. Discrete jump-size distributions provide flexibility and robustness that could serve as alternative to continuous distributions in some settings.

Thus, our proposed model incorporates discrete jump-size distributions in a discrete-time framework that naturally gives rise to a multinomial lattice framework to model the jump-diffusion process. The resulting multinomial lattice is a generalization of the multinomial lattice developed in Amin (1993), who priced options using backwards induction. Under our framework, the option price, which is an expectation under the martingale measure, requires the calculation of a summation. For most options, the summation is an implicit enumeration problem of the probability-weighted lattice paths for the underlying asset price. The use of generating functions from enumerative combinatorics provides an efficient method to carry out such calculations, and Li and Zhao (2009) used the generating function method to price Parisian options under the binomial lattice framework. Here, we apply the method to the multinomial lattice framework, which is significantly more computationally burdensome than a recombining binomial tree. By calculating the cardinal number and probability weight of all the possible price paths, we derive the price distribution of the underlying asset on the expiration date. Then we derive a unified pricing formula that leads to a closed-form solution for European options on the proposed discrete jump-diffusion model. We prove that the resulting formula converges to Merton’s continuous-time pricing formula as the number of time steps N goes to infinity, where the computation complexity is of order $O(N^{1.5})$.

Our proposed solution is very flexible and robust, since the same valuation formula applies to European options with any jump-size distribution. For example, discrete jump-size distributions (see Rachev et al. 2010), such as binomial and negative binomial distributions (Benninga and Wiener 1998) and hypergeometric distributions (Humpage 1997, Santos and Guerra 2015), have been widely used in financial modeling but not so much in option pricing, as a result of the challenge of deriving a separate analytical option pricing formula for each different jump-size distribution, but they could be easily incorporated in our framework. Numerical experiments on valuing options under different jump-size distributions using our pricing formula indicate that the shape of the jump-size distribution could have a significant impact on the price of an option. With the flexibility of being able to incorporate any jump-size distribution, our model allows one to easily compare the pricing performance resulting from different distributions, which can help improve price accuracy by virtue of better distribution choices, e.g., to model rare events in a financial crisis.

In sum, our work contributes to the option pricing research literature as follows:

- We provide a closed-form solution for European options on jump-diffusion processes with general discrete jump-size distributions, which includes nonparametric distributions such as empirical distributions that can be used to model actual data directly and better capture leptokurtic features.
• We prove that the analytical solution under the discrete-time framework converges to Merton’s continuous-time formula as the number of time steps $N$ goes to infinity, with a computation complexity of order $O(N^{1.5})$.

• We develop a general combinatorial approach for derivatives pricing on a multinomial lattice that is significantly more computationally efficient than previous dynamic programming-based backward recursion methods.

The rest of the paper is organized as follows. In Section 2, the multinomial lattice model and lattice path option pricing method are presented. In Section 3, we derive the European option pricing formula for discrete-time jump-diffusion processes and prove convergence of the formula to the price under the continuous-time process. Section 4 presents numerical experiments both to demonstrate the computational properties of the pricing formula and to empirically investigate the impact of the shape of jump-size distributions on the option prices. Section 5 concludes this paper and provides some directions for future research.

2. Model Setting

The theoretical model for option pricing in discrete time is constructed in this section.

2.1. Lattice Construction

In a discrete-time framework, Cox et al. (1979) provided an option pricing formula using the binomial lattice model. Here, we generalize the binomial lattice model to the multinomial lattice model, where the constraint of asset price dynamics being discrete and moving among lattice points only is retained. In the multinomial lattice model, the asset price lattice has more than two—possibly infinite—lattice points at each time step.

We consider the price dynamics of an asset $S_t$ over the interval $t \in [0, T]$, which is divided into $N$ equal intervals of length $\tau = T/N$, where $S_0$ denotes the initial price state of the asset. Then we have the asset price path of an asset $\{S_0, S_1, S_2, \ldots, S_T\}$ over the $N$ time steps. Defining the (proportional) price change at time step $k$ by $X_k$, then the price relationship between time steps $k-1$ and $k$ is given by

$$X_k = \frac{S_k}{S_{k-1}},$$

where $X_k$ is a random variable that takes value in the state space $\{u^i, i \in \mathbb{Z}\}$, $u > 1$.

Therefore, for a fixed initial price $S_0$, each sample asset price path can be defined by a sequence of (one-step) moves on the lattice that we will call a lattice path. The concept of lattice path was first introduced by Li and Zhao (2009) in the binomial lattice framework.

2.2. Probability Measure

We express the probability of a lattice path in terms of one-step move probabilities, which are assumed to be mutually independent and given by

$$P(X_k = u^i) = p_i, \quad i \in \mathbb{Z}. \quad (2)$$

As a result, we can define the lattice path probability as below.

**Definition 2.** For lattice path $\alpha = (x_1, x_2, \ldots, x_n)$, the lattice path probability is defined by

$$P(\alpha) \doteq \prod_{k=1}^{n} P(X_k = x_k), \quad (3)$$

where $P(X_k)$ is defined as above.

Using the lattice path $\alpha_0$ in Figure 1 as an example, we have

$$P(\alpha_0) = p_{+1}p_{-1}p_{-1}p_{+1}p_{h}p_{-1}p_{+1}p_{-1}p_{h} = p_{+1}p_{-1}p_{+1}p_{-1}p_{h}p_{h}.$$

\begin{itemize}
  \item • We develop a general combinatorial approach for derivatives pricing on a multinomial lattice that is significantly more computationally efficient than previous dynamic programming-based backward recursion methods.
  \item • We prove that the analytical solution under the discrete-time framework converges to Merton’s continuous-time formula as the number of time steps $N$ goes to infinity, with a computation complexity of order $O(N^{1.5})$.
\end{itemize}
2.3. Pricing Technique
Here, we provide an overview on how we will use the lattice path technique defined for any given probability measure to price an option. From arbitrage-free option pricing theory, under a specific martingale probability measure \( \mathbb{P} \), the value of a European option is equal to the expectation of its discounted future payoffs (see Harrison and Kreps 1979, Harrison and Pliska 1981); i.e.,

\[
V = e^{-rT} E_\mathbb{P} [F_T],
\]

where \( F_T \) denotes the European option payoff at the expiration date \( T \). Common examples are \( F_T = (S_T - K)^+ \) for a European call option and \( F_T = (K - S_T)^+ \) for a European put option, where \( K \) is the strike price. In Section 3.1, we will specify \( \mathbb{P} \) for the jump-diffusion model.

In the lattice framework, the underlying asset price \( S_T \) at expiration date \( T \) can be written as

\[
S_T = S_{N'} e^{\alpha},
\]

where \( \alpha = (x_1, x_2, \ldots, x_N) \) is the lattice path (see Definition 1). Hence the option value can be calculated by the weighted-sum of the payoff along all feasible lattice paths. For payoffs that only depend on the lattice height, we can count all possible paths ending with height \( h \) and then compute the probability weight of each \( h \).

Taking a European call option as an example,

\[
V_{\text{call}} = e^{-rT} \mathbb{E}_\mathbb{P} [(S_T - K)^+] = e^{-rT} \mathbb{E}_\mathbb{P} [(S_{N'} - K)^+] = e^{-rT} \sum_{\text{lattice paths } \alpha = (x_1, x_2, \ldots, x_N) \text{ to } h}(\mathbb{P}(\alpha)(S_{N'} - K)^+)
\]

\[
= e^{-rT} \sum_{h} \sum_{\alpha: l(\alpha) = N', |\alpha| = h} \mathbb{P}(\alpha)(S_{N'} - K)^+)
\]

\[
= e^{-rT} \sum_{h} (S_{N'} - K) \mathbb{P}_N(h),
\]

where

\[
\mathbb{P}_N(h) \triangleq \sum_{\alpha: l(\alpha) = N', |\alpha| = h} \mathbb{P}(\alpha) = \sum_{i_1 + \cdots + i_N = h} p_{i_1} \cdots p_{i_N}. \tag{4}
\]

Therefore, we have the following pricing formula for an arbitrary discrete stochastic process on the general lattice model.

**Proposition 1.** For a European option with strike price \( K \), expiration date \( T \), and initial underlying asset value \( S_0 \), the values of the call and put options under a multinomial lattice with grid size \( u \) can be written as

\[
V_{\text{call}} = e^{-rT} \sum_{h \geq K} \mathbb{P}_N(h)(S_{N'} - K) \quad \text{and} \quad V_{\text{put}} = e^{-rT} \sum_{h \geq K^\text{put}} \mathbb{P}_N(h)(K - S_{N'}), \tag{5}
\]

respectively, where \( \mathbb{P}_N(h) \) is given by Equation (4), \( r \) is the riskless rate, and

\[
h_{\text{call}} = \lfloor \log_u (K/S_0) \rfloor, \quad h_{\text{put}} = \lceil \log_u (K/S_0) \rceil. \tag{6}
\]

Note that \( \lfloor \cdot \rfloor \) and \( \lceil \cdot \rceil \) denote the (greatest/least integer) ceiling and floor functions, respectively, and \( h_{\text{call}} \) and \( h_{\text{put}} \) are the least and greatest integers guaranteeing \((S_{N'} - K)^+ = S_{N'} - K \) and \((K - S_{N'})^+ = K - S_{N'}\), respectively.

Thus, we just need to calculate the payoff and probability weight for each possible lattice path to obtain the option value or, more precisely, the payoff and probability weight \( \mathbb{P}_N(h) \) of each height \( h \) on the \( N \)-length lattice. We have converted a European option pricing problem into a lattice path enumeration problem in combinatorics mathematics.

It is relatively straightforward to calculate the number of lattice paths leading to the same lattice height if the underlying asset price follows a binomial lattice model, but for a multinomial lattice model, the task becomes much more challenging, especially in the setting where one-step moves can take the price to any point in the lattice, which itself could in principle be unbounded.

3. Pricing European Options for the Jump-Diffusion Model
In this section, we derive the closed-form pricing formula for a European option where the underlying asset price dynamics follows a discrete-time jump-diffusion process using the lattice model introduced in Section 2. Then we prove that the formula converges to the corresponding continuous-time formula when the number of time steps \( N \) goes to infinity (for a fixed \( T \)), and we demonstrate that the formula has an efficient computational complexity of \( O(N^{1.5}) \). We also generalize the formula to the setting where the asset price volatility is time dependent but deterministic.

3.1. Pricing the Discrete Jump-Diffusion Model
Section 2 introduced the general lattice model for an arbitrary discrete stochastic process. Here, we specify the general lattice model for jump-diffusion processes.

A jump-diffusion process is made up of a diffusion process and a jump process. Following the lattice model with one-step moves \( X_0 \) defined by (1), we define the discrete-time jump-diffusion model by a one-step move that is a random walk \( W = u^{\pm 1} \) (up or down) or a jump \( Y \) of size \( |w_j| \), as illustrated in Figure 2.

As in Merton (1976) and Amin (1993), we assume that the jump component is a Poisson
process with arrival rate $\lambda$. If the jump risks are systematic, we can obtain a corresponding martingale process $\tilde{\mathbb{P}}$ by means of the discrete Radon–Nikodym derivative, from which the option can be priced, as in Amin (1993). If the volatility of the diffusion process under the physical probability measure is $\sigma$, and the jump-size distribution is $\mathcal{Y}$, then the one-step move probabilities under the martingale probability measure $\tilde{\mathbb{P}}$ are given by

$$
P(X_k) = \begin{cases} 
1-q & X_k = W, \\
q & X_k = Y, \\
(1-q)p & X_k = W = u, \\
q(p)(1-p) & X_k = W = d, \\
q\rho_j & X_k = Y = u, \quad j \in \mathbb{Z}\setminus\{\pm1\},
\end{cases}
$$

where

$$
q = \lambda \tau, \quad p = \frac{(e^{\sigma^2} - \lambda \tau \mathbb{E}_y[Y]/(1-\lambda \tau)) - d}{u - d},
$$

and the jump-size distribution is written as

$$
Y \sim \mathcal{Y}; \quad \mathbb{P}(Y = u) = \rho_j, \quad j \in \mathbb{Z}\setminus\{\pm1\}, \quad \sum_j \rho_j = 1.
$$

Also, to ensure that $p > 0$, the length of time subinterval $\tau$ must be taken sufficiently small. Under the assumption of diversifiability, the jump-size distribution under the martingale probability measure is the same as under physical probability measure (Amin 1993).

For ease of enumeration, we standardize the multinomial lattice by considering the logarithm of the asset price $\log_u S_t$. The previously defined concepts on the lattice model can be naturally translated to the standard lattice: lattice path $\alpha = (x_1, x_2, \ldots, x_n)$, $x_k \in \{i, i \in \mathbb{Z}\}$, lattice height $|\alpha| = \sum_{k=1}^{n} x_k$, and the log jump-size distribution

$$
J \sim \mathcal{Y}; \quad \mathbb{P}(J = j) = \mathbb{P}(Y = u^j) = \rho_j, \quad j \in \mathbb{Z}\setminus\{\pm1\}.
$$

Figure 3 shows the lattice path $\alpha_0$ from Figure 1 on the standard lattice. Note that this standardization procedure does not change the one-step move probabilities nor the resulting lattice path probabilities.

Now we calculate the probability weight $\mathbb{P}_N(h)$ of lattice height for the discrete jump-diffusion model to obtain a closed-form pricing formula for European options.

**Theorem 1.** For the discrete jump-diffusion process defined on a multinomial lattice with one-step move probabilities,

$$
P(X_k) = \begin{cases} 
1-q & X_k = W, \\
q & X_k = J, \\
(1-q)p & X_k = W = 1, \\
q(p)(1-p) & X_k = W = -1, \\
q\rho_j & X_k = J = j, \quad j \in \mathbb{Z}\setminus\{\pm1\};
\end{cases}
$$

with $p$ and $q$ defined by (7), the probability weight is given by

$$
\mathbb{P}_N(h) = \sum_{I=0}^{N-1} \sum_{U=0}^{N-1} \binom{N}{I} q^{I}(1-q)^{N-I} p^{U} \cdot (1-p)^{N-I-U} \rho_{N+h-I-2U},
$$

For simplification, we retain almost all the same notation after standardization, such as $X_k$, $W$, $\alpha$, with the meaning understood by context.
where \( (N_{i,j}) = N!/I!U!(N - I - U)! \) is the multinomial
coefficient, and \( \rho_{j}^{(l)} = \sum_{i+j=1} \rho_{i} \cdots \rho_{j} \) is the
conditional probability of a cumulative jump size of \( L \) lattice
points over \( I \) jumps.

**Proof.** For an \( h \)-height lattice path \( \alpha = (x_{1}, x_{2}, \ldots, x_{N}) \),
letting \( U, D, \) and \( I \) denote the number of random
walk up, random walk down, and jump moves in the first \( N \) time steps,
respectively, and letting \( j_{i} \) denote the jump size as a result of the \( i \)th jump,
we have
\[
h = |\alpha| = \sum_{k=1}^{N} x_{k} = U(1) + D(1) + \sum_{i=1}^{I} j_{i} = U - D + L = 2U + I + L - N,
\]
where \( L = \sum_{i=1}^{I} j_{i} \).

From Lemma 3 of Appendix B.1, we know that under the discrete jump-diﬀusion model, the generating function of \( \mathbb{P}_{N}(h) \) is
\[
G_{N}(z; \bar{\mathcal{H}}) = \sum_{l=0}^{N} \sum_{I=0}^{N-l} \sum_{U=0}^{\infty} \frac{N!}{I!U!(N - I - U)!} q^{I}(1-q)^{N-I} \cdot p^{U}(1-p)^{N-I-U} \cdot \rho_{l}^{(I)} \cdot z^{2U+I+L-N}.
\]
Since \( \mathbb{P}_{N}(h) \) is the coeﬃcient of the term \( z^{h} \), (11) follows.

Combining Proposition 1 and Theorem 1, we obtain the closed-form pricing formula for European options.

**Corollary 1.** For a European option with strike price \( K \),
expiration date \( T \), and initial underlying asset value \( S_{0} \),
where the underlying asset price process \( \{S_{t}\} \) follows a
discrete-time jump-diﬀusion process with one-step move probabilities given by (10),
the values of the call and put options under a multinomial lattice with grid size \( u \) are given by
(5), (6), and (11).

### 3.2. Continuous-Time Limit

Now we prove that as the number of time steps \( N \) goes to infinity,
the discrete-time call option pricing formula given by Corollary 1 converges to the general
continuous-time jump-diﬀusion call option formula\(^2\) (Merton 1976) given by
\[
V_{c} = \sum_{l=0}^{\infty} e^{-\lambda T} \left( \frac{\lambda T}{I!} \right) \cdot \mathbb{E}_{\bar{\mathcal{H}}} \left[ \text{BS} \left( e^{-(\lambda-1)T} S_{0} \prod_{i=1}^{I} y_{i}; T; K, \sigma^{2}, r \right) \right],
\]
where \( \lambda \) is the riskless rate, \( \bar{\mathcal{H}} \) is the continuous jump-size distribution (subscript \( c \) is added to differentiating from the discrete jump-size distribution \( \bar{\mathcal{H}} \) defined earlier), \( \mu = \mathbb{E}_{\bar{\mathcal{H}}} [Y] \) is the conditional expectation of

\(^2\) This formula has no requirement on the form of the jump-size distribution,
differing from the usual version found in the literature.

the return from one jump, and \( \text{BS} \) denotes the Black-
Scholes formula given by
\[
\text{BS}(S_{0}, T; K, \sigma^{2}, r) = S_{0}N(d_{1}) - e^{-rT} K N(d_{2}),
\]
where \( N(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-t^{2}/2} dt \),
\[
d_{1} = \frac{\ln(S_{0}/K) + (r + \sigma^{2}/2)T}{\sigma\sqrt{T}},
\]
\[
d_{2} = \frac{\ln(S_{0}/K) + (r - \sigma^{2}/2)T}{\sigma\sqrt{T}}.
\]

For notational convenience, we denote
\[
q(I) = \frac{e^{-\lambda T} (\lambda T)^{l}}{I!}
\]
and
\[
V_{c}(I) = \mathbb{E}_{\bar{\mathcal{H}}} \left[ \text{BS} \left( e^{-(\lambda-1)T} S_{0} \prod_{i=1}^{I} y_{i}; T; K, \sigma^{2}, r \right) \right].
\]

then the continuous-time pricing formula is written as
\[
V_{c} = \sum_{l=0}^{\infty} q(l) V_{c}(l) = \mathbb{E}_{\bar{\mathcal{H}}} [V_{c}(\xi)], \quad \xi \sim \mathbb{P}(\lambda T), \quad (12)
\]
where \( \mathbb{P} \) denotes the Poisson distribution.

Before presenting the convergence result, we introduce the following notation to simplify the expressions of the discrete-time formula:
\[
q_{N}(I) = \left( \frac{N}{I} \right) q^{I}(1-q)^{N-I}, \quad \text{where} \quad \left( \frac{N}{I} \right) = \frac{N!}{I!(N-I)!},
\]
\[
V_{N}(l) = e^{-\lambda T} \sum_{I=0}^{\infty} \left( S_{0}u^{I} K \right) \sum_{I=0}^{N-I} \left( \frac{N-I}{I} \right) \cdot p^{U}(1-p)^{N-I-U} \cdot \rho_{l}^{(I)} \cdot \mathbb{P}_{N+h-2l}. \quad (13)
\]

It follows that we can rewrite the value of European call option \( V_{\text{call}} \) as
\[
V_{N} \approx V_{\text{call}} = \sum_{l=0}^{N} q_{N}(I) V_{N}(l) = \mathbb{E}_{\mathbb{P}} [V_{N}(\xi_{N})],
\]
\[
\xi_{N} \sim \mathbb{B}(N, q), \quad (14)
\]
where \( \mathbb{B} \) denotes the binomial distribution.

Notice that in Equation (8), we assume the jump size follows a discrete distribution. If the jump-size distribution is continuous, we approximate the possible values of \( Y \) by integer powers of \( u \) (see Appendix D.1 for details):
\[
Y \sim \hat{Y}: \mathbb{P}(Y = u^{j}) = \hat{\rho}_{j}.
\]

Convergence of the proposed discrete-time analytical pricing solution given by Equations (13) and (14) to the continuous-time formula (12) is established by the following theorem, whose proof is provided in Appendix C.2.
The European call and put option pricing formulas involving function (JDGF) algorithm for implementing numerical examples later). Our proposed jump-diffusion generating function (JDGF) algorithm for implementing the European call and put option pricing formulas of the numerical examples, is given in Appendix D. Explicit pseudocode in MATLAB and Mathematica can be found in Appendix E. A method for obtaining the discrete jump-size distribution $\tilde{\mathcal{Y}}: \{\rho_i\}_{i=0}^{M^+}$ from $\tilde{\mathcal{Y}}$ or $\tilde{\mathcal{Y}}_c$, which is used in all of the numerical examples, is given in Appendix D.

We now show that in terms of computational complexity, the pricing formulas given by (5) and (15)—as implemented with the additional truncations parameters $M_b$ and $R$—represent a dramatic improvement over DP-based recursion. Specifically, we prove that the computational complexity of our proposed JDGF algorithm is $O(N^{1.5})$, whereas the complexity of the DP-based recursive algorithm on the same lattice is $O(N^3)$.³

Theorem 3. The computational complexity of the JDGF option pricing algorithm implemented using (5), (15), and (16) is $O(N^{1.5})$.

Proof. During the numerical estimation of the price of a European call option, as demonstrated in Appendix D, we only need to calculate

$$V_N = \sum_{l=0}^{l_0} q_N(I) V_N(I),$$

where $V_N(I)$ is given by Equation (13) for a call option (the put option expression is similar). Note that the value of $l_0$ defined by (16) is independent of $N$, so the computational complexity of $V_N(I)$ is the same as that of $V_N(I)$.

To compute the complexity of $V_N(I)$, we note that, as shown in Section 3.1 and Appendix B.1, the coefficient of $V_N(I)$ is exactly the coefficient of the term $Z^h$

³The special truncation procedure for the multinomial lattice described in Amin (1993, p. 1852) results in an improved $O(N^{1.5})$ complexity, but this truncation procedure is less accurate and may exhibit numerical instabilities.
in the generating function \( G_N(z; \bar{\varphi}; I) \) given by Equation (B2) in Appendix B.1. Since \( V_N(I) \) only includes terms with \( h \geq h^{\text{call}} \), the computational complexity of \( V_N(I) \) is not more than \( G_N(z; \bar{\varphi}; I) \), which we now show has complexity \( O(N^{1.5}) \).

Consider the form of \( G_N(z; \bar{\varphi}; I) \) in the line prior to Equation (B2) in Appendix B.1:

\[
G_N(z; \bar{\varphi}; I) = \sum_{u=0}^{N-1} \binom{N-1}{u} p^u (1-p)^{N-1-u} z^{2u-I-N} \sum_{j+m-z}^{\infty} \rho_j z^j;
\]

and therefore, for a fixed \( I \), the product of two polynomials having \( O(N^{0.5}) \) monomials, so the complexity of \( G_N(z; \bar{\varphi}; I) \) is \( O(N) \); thus, the complexity of the JDF option pricing algorithm is at most \( O(N^{1.5}) \). □

The next result establishes the computational complexity of DP-based recursion on the same multinomial lattice constructed in Section 2.

**Proposition 2.** The computational complexity of the DP-based recursive algorithm on the multinomial lattice with one-step move probabilities given by (10) is \( O(N^3) \).

**Proof.** Since \( N \) is the number of time steps, at time step \( k \), there are \( kO(N^{0.5}) \) points in the multinomial lattice. For each point, the price could reach \( O(N^{0.5}) \) possible points at the next time step; thus the holding value for each point is calculated on \( O(N^{0.5}) \) possible points using backward induction. Therefore, the complexity for estimating the expected option holding value is \( \sum_{k=0}^{N-1} kO(N^{0.5})O(N^{0.5}) = O(N^3) \). □

### 3.4. Time-Varying Volatility Case

We now extend the analysis to the setting of time-dependent volatility, defined by

\[
\sigma(t) = \sigma_s, \quad t \in [t_{s-1}, t_s), \quad s = 1, \ldots, S,
\]

where \( 0 = t_0 < t_1 < \cdots < t_S = T \); i.e., the volatility is piecewise constant over each time interval \([t_{s-1}, t_s)\).

We divide each time interval \([t_{s-1}, t_s)\) into \( N_s \) equal subintervals:

\[
\tau_s = \frac{t_s - t_{s-1}}{N_s}, \quad s = 1, \ldots, S.
\]

Note that \( \tau_s \) is chosen in each time interval \([t_{s-1}, t_s)\) to guarantee that \( u_s = e^{\mu \sqrt{\tau_s}} \), \( s = 1, \ldots, S \) remains a constant (which is denoted as \( u \)) throughout the entire time horizon.

As a result, the one-step move probability in each subinterval \([t_{s-1}, t_s)\) retains the same form as Equation (2):

\[
P(X_i = u^i) = p_i^{(0)}, \quad i \in \mathbb{Z},
\]

where \( k \in (N_1 + \cdots + N_{s-1}, N_1 + \cdots + N_{s-1} + N_s) \) is an integer representing the time step in the time interval \([t_{s-1}, t_s)\). Thus, for the general lattice model of an arbitrary discrete stochastic process, we have the following pricing formula.

**Theorem 4.** On the multinomial lattice where the underlying asset price has time-varying volatility (17) and one-step move probabilities (18), the probability weight is given by

\[
P_N(h) = \sum_{h = h_1 + \cdots + h_s} \prod_{s=1}^S P_N(h_s),
\]

where \( \tilde{N} \equiv (N_1, N_2, \ldots, N_s) \), and \( P_N(h_s) \) represents the probability weight in time interval \([t_{s-1}, t_s)\) as defined in (4).

The proof is provided in Appendix B.2.

**Proposition 3.** For a European option with strike price \( K \), expiration date \( T \), and underlying asset initial value \( S_0 \), where the underlying asset price has time-varying volatility (17) and one-step move probabilities (18), the values of the call and put options under a multinomial lattice with grid size \( u \) can be written as

\[
\begin{align*}
V_N^{\text{call}} &= e^{-rT} \sum_{h \geq h^{\text{call}}} \mathbb{P}(h)(S_0 u^h - K) \quad \text{and} \\
V_N^{\text{put}} &= e^{-rT} \sum_{h \geq h^{\text{put}}} \mathbb{P}(h)(K - S_0 u^h),
\end{align*}
\]

respectively, where \( r \) is the riskless rate, and \( h^{\text{call}} \) and \( h^{\text{put}} \) are defined in (6).

Next we focus on the setting where the underlying asset dynamics follows a jump-diffusion process. The one-step move probabilities are given by

\[
\begin{align*}
P(X_k) = \begin{cases} 
(1 - q_j) q_s, & X_k = W = 1, \\
(1 - q_j) (1 - p_s), & X_k = W = -1, \\
q_j p_s, & X_k = j,
\end{cases}
\]

where \( k \in (N_1 + \cdots + N_{s-1}, N_1 + \cdots + N_{s-1} + N_s) \) is an integer representing the time step in the time interval \([t_{s-1}, t_s)\), and \( q_s, p_s \) are defined as

\[
q_s = \lambda \tau_s, \quad p_s = \frac{e^{r \tau_s} - \lambda \tau_s \mathbb{E}[Y] / (1 - \lambda \tau_s) - d}{u - d}.
\]

Combining Theorem 1, Theorem 4, and Proposition 3, we have the following corollary.
Corollary 2. For a European option with strike price $K$, expiration date $T$, and underlying asset initial value $S_0$, where the underlying asset price follows a discrete-time jump-diffusion process with time-varying volatility (17) and one-step move probabilities (19), the values of the call and put options under a multinomial lattice with grid size $u$ can be written as

$$V^c_{h} = e^{-rT} \sum_{h \in h^c} P^c_{h}(S_0 u^h - K)$$

$$V^p_{h} = e^{-rT} \sum_{h \in h^p} P^p_{h}(K - S_0 u^h),$$

respectively, where $r$ is the riskless rate, $h^c$ and $h^p$ are defined in (6), and

$$P^c_{h}(h) = \prod_{i=0}^{s} \sum_{h_{i+1}+\cdots+h_s=0}^{N_s} \frac{\left( N_s \right)_{i}}{s!} q^i (1 - q_u)^{N_s-i}$$

$$\cdot p^i_u (1 - p_u)^{N_s-i-u} p_u^{j-i} \cdot p_u^{j-i}$$

where $q_u$ and $p_u$ are defined in (20).

4. Numerical Results

In this section, we price European options in our discrete-time framework for the underlying asset price following a jump-diffusion process. We first consider log-normal and log-double-exponential jump-size distributions, where analytical results exist for the continuous-time setting. The results demonstrate that the proposed JDGF option pricing algorithm converges to the continuous-time model for large enough $N$ and is computationally efficient, especially when compared with dynamic programming. In particular, the numerical experiments validate the theoretical computational complexity. We then investigate the impact of different jump-size distributions on option prices using parameters estimated from daily data on the stock price of IBM. All numerical results were obtained using a MATLAB implementation of the JDGF European option pricing algorithm in Figure 4 (see Appendix E for pseudocode in both MATLAB and Mathematica), with CPU times reported using a 2.90 GHz 8 GB RAM desktop computer.

4.1. Merton’s Log-Normal Distribution

We price a European put option with diffusion parameters $r = 0.08$ and $\sigma^2 = 0.05$, Poisson rate $\lambda = 5$, jump-size distribution $\ln N(-0.025, 0.05)$, initial asset price $S_0 = 40$, strike price $K = 45$, and expiration $T = 0.5$. The jump truncation parameter is $I_0 = 12$, calculated using Equation (16) for $\varepsilon = 10^{-5}$. Under different time steps $N$, we calculate the truncation parameters $M_0$ and $R$ using the procedure in Appendix D.

European put option prices under different time steps $N$ are shown in Table 1, where the option price under Merton’s pricing formula is given in the last column.

<table>
<thead>
<tr>
<th>$N$</th>
<th>500</th>
<th>1,000</th>
<th>1,500</th>
<th>2,000</th>
<th>2,500</th>
<th>3,000</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_0$</td>
<td>95</td>
<td>135</td>
<td>165</td>
<td>190</td>
<td>213</td>
<td>233</td>
</tr>
<tr>
<td>$R$</td>
<td>10</td>
<td>14</td>
<td>17</td>
<td>19</td>
<td>21</td>
<td>23</td>
</tr>
<tr>
<td>Price</td>
<td>7.9028</td>
<td>7.9037</td>
<td>7.9044</td>
<td>7.9039</td>
<td>7.9040</td>
<td>7.9044</td>
</tr>
<tr>
<td>CPU (secs.)</td>
<td>0.2</td>
<td>0.3</td>
<td>0.6</td>
<td>0.9</td>
<td>1.3</td>
<td>1.8</td>
</tr>
</tbody>
</table>

4.2. Kou’s Log-Double-Exponential Distribution

We price European call options with diffusion parameters $r = 0.05$ and $\sigma = 0.3$; Poisson rate $\lambda = 3$; jump size distribution $\ln \mathcal{C}(0.6, 0.4, 20, 20)$ with probability density function

$$f_{\ln Y}(y) = 0.6 \cdot 20 e^{-20y} \cdot I_{[y \geq 0]} + 0.4 \cdot 20 e^{20y} \cdot I_{[y < 0]};$$

initial asset price $S_0 = 100$; strike prices $K = 90, 100, 110$; and expiration $T = 1$.

We choose $M_0$, $R$, and $I_0$ based on the rules described in Appendix D. The prices using our pricing formula for different time steps $N$ are shown in Table 2, where the corresponding values under Kou’s pricing formula are displayed in the last column.

<table>
<thead>
<tr>
<th>$N$</th>
<th>500</th>
<th>1,000</th>
<th>2,000</th>
<th>3,000</th>
<th>5,000</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_0$</td>
<td>61</td>
<td>85</td>
<td>121</td>
<td>148</td>
<td>190</td>
</tr>
<tr>
<td>$R$</td>
<td>955</td>
<td>194</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>CPU (secs.)</td>
<td>0.13</td>
<td>0.32</td>
<td>0.95</td>
<td>1.85</td>
<td>4.59</td>
</tr>
</tbody>
</table>

4.3. Computational Complexity

The theoretical computational complexity results in Section 3.3 indicate a substantial improvement over the DP-based recursive approach. Here, we test empirical performance on several numerical examples, comparing the computation time of our JDGF algorithm with DP. We fix $I_0 = 12$ and price the European put option with riskless rate $r = 0.08$, strike price $K = 45$, and expiration time $T = 0.5$. The underlying asset has initial price $S_0 = 40$, volatility $\sigma^2 = 0.05$, and jump frequency parameter $\lambda = 5$, with the same log-normal jump-size distribution used in Section 4.1. Table 3 shows the option price and CPU time with different time steps $N$ under both algorithms. The results indicate that the JDGF algorithm is both more accurate and computationally more efficient. CPU time as a function of the number of time steps $N$ for both algorithms is plotted in Figures 5.
and 6, and the results support the theoretical computational complexity results.

4.4. Sensitivity of Option Prices to Jump-Size Distributions

To investigate the sensitivity of option prices to the jump-size distribution, we compare the option prices with various jump-size distributions by applying the JDGF algorithm to an example where the diffusion component is identical and the first two moments of the jump-size distribution are matched.

The underlying asset price parameters are fitted using daily data on the stock price of IBM from January 1962 to December 2015. To fit the discrete jump-diffusion process and mean $\alpha = 0.0011$ and standard deviation $\delta = 0.05$ for the log-jump distribution. In addition to the discrete empirical distribution, three other (continuous) distributions are considered.

- Normal $\mathcal{N}$:
  \[
  f_{\ln Y}(y) = \frac{1}{\sqrt{2\pi}0.05} e^{-y^2/(2\times0.05^2)}
  \]

- Double-exponential $\mathcal{E}$:
  \[
  f_{\ln Y}(y) = 0.335 \times 16.5 e^{-16.5\frac{y}{10}} + 0.665 \times 34.5 e^{34.5\frac{y}{10}} 
  \]

- Double-Gamma $\mathcal{G}$:
  \[
  f_{\ln Y}(y) = 0.48 \times 35.3 e^{-35.3\frac{y}{10}} + 0.52 \frac{(-y)^{0.21-1}}{\Gamma(0.21)0.1145^{0.21}} e^{y/0.1145} \mathbb{I}_{|y|<0}
  \]

The density functions of the three continuous distributions and the discrete empirical distribution are shown in Figure 7.

Then the parameters for the discrete jump-diffusion model estimated from the data are volatility $\sigma = 0.19$ and Poisson arrival rate $\lambda = 11.5$ for the jump-diffusion process and mean $\alpha = 0.0011$ and standard deviation $\delta = 0.05$ for the log-jump distribution. In addition to the discrete empirical distribution, three other (continuous) distributions are considered.

- Normal $\mathcal{N}$:
  \[
  f_{\ln Y}(y) = \frac{1}{\sqrt{2\pi}0.05} e^{-y^2/(2\times0.05^2)}
  \]

- Double-exponential $\mathcal{E}$:
  \[
  f_{\ln Y}(y) = 0.335 \times 16.5 e^{-16.5\frac{y}{10}} + 0.665 \times 34.5 e^{34.5\frac{y}{10}} 
  \]

- Double-Gamma $\mathcal{G}$:
  \[
  f_{\ln Y}(y) = 0.48 \times 35.3 e^{-35.3\frac{y}{10}} + 0.52 \frac{(-y)^{0.21-1}}{\Gamma(0.21)0.1145^{0.21}} e^{y/0.1145} \mathbb{I}_{|y|<0}
  \]

The density functions of the three continuous distributions and the discrete empirical distribution are shown in Figure 7.

Table 3 Pricing European Put Options Under Log-Normal Jump-Size Distribution $\ln r \sim (-0.025, 0.05); \sigma^2 = 0.05, S_0 = 40, r = 0.08, T = 0.5, \lambda = 5, K = 45, \tau = 12$

<table>
<thead>
<tr>
<th>$N$</th>
<th>DP recursive</th>
<th>JDGF</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Option price</td>
<td>Option price</td>
</tr>
<tr>
<td>100</td>
<td>7.8961</td>
<td>7.8890</td>
</tr>
<tr>
<td>200</td>
<td>7.9006</td>
<td>7.8982</td>
</tr>
<tr>
<td>500</td>
<td>7.9036</td>
<td>7.9028</td>
</tr>
<tr>
<td>800</td>
<td>7.9043</td>
<td>7.9021</td>
</tr>
<tr>
<td>1000</td>
<td>7.9045</td>
<td>7.9037</td>
</tr>
<tr>
<td>1500</td>
<td></td>
<td>7.9044</td>
</tr>
<tr>
<td>2000</td>
<td></td>
<td>7.9039</td>
</tr>
<tr>
<td>5000</td>
<td></td>
<td>7.9046</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>CPU time (secs.)</th>
<th>Price range</th>
<th>CPU time (secs.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>1.90</td>
<td>7.8844 to 7.9605</td>
<td>196 (3.4)</td>
</tr>
<tr>
<td>200</td>
<td>11.73</td>
<td></td>
<td></td>
</tr>
<tr>
<td>500</td>
<td>127</td>
<td></td>
<td></td>
</tr>
<tr>
<td>800</td>
<td>468</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>953</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1500</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5000</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note. The Monte Carlo results are based on 10 macroreplications of 100,000 paths with $N = 1,000$, where the entries show the option price range and CPU time mean (standard errors in parentheses).
5. Conclusions and Future Research

Using the generating function technique from enumerative combinatorics, we derive an analytic pricing formula for European options under a discrete-time jump-diffusion framework. In contrast to the continuous setting, where the pricing methodology is usually tailored to a specific jump-size distribution, our pricing formula can be applied to general jump-size distributions. As the number of time steps $N$ goes to infinity, the formula converges to the corresponding continuous-time formula of Merton (1976), with a computational complexity of $O(N^{1.5})$. Numerical experiments demonstrate the flexibility and efficiency of the JDGF algorithm, as well as the sensitivity of option prices to different jump-size distributions. Computationally, the method is orders of magnitude faster than Monte Carlo simulation for European options, although it would worthwhile to investigate how it performs against various transform methods (e.g., Carr and Madan 1999, Cai et al. 2014, Feng and Linetsky 2009, Feng and Lin 2013).

The empirical results in Section 4.4 used asset prices to estimate the parameters of the jump-diffusion model for the purpose of demonstrating sensitivity of option prices to different types of jump-size distributions. In practice, market data on actual option prices are used to calibrate any option pricing model. Although not a focus of this work, a calibration procedure based on actual option prices is required to make the JDGF algorithm implementable for a practitioner. We briefly suggest one possible approach here, which adopts the regularization method of Cont and Tankov (2004); however, determining a good procedure is definitely a critical need for further research.

The calibration approach in Cont and Tankov (2004) is a nonparametric method for fitting a (finite activity) jump-diffusion process to a finite set of observed option prices. Their approach appears to be well suited to our model, since it can be applied to an arbitrary discrete jump distribution, such as an empirical distribution, and is especially effective for compound

---

### Table 4 Prices and Implied Volatilities of European Put Option Under Different Jump-Size Distributions Based on IBM Daily Asset Price Data from January 1962 to December 2015

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>110</td>
<td>10.6935</td>
<td>0.2548</td>
<td>10.5909</td>
<td>0.2487</td>
<td>10.4231</td>
<td>0.2385</td>
<td>11.2193</td>
<td>0.2854</td>
</tr>
<tr>
<td></td>
<td>(+0.96%)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(-2.53%)</td>
<td></td>
<td>(+4.92%)</td>
</tr>
<tr>
<td>100</td>
<td>4.4394</td>
<td>0.2543</td>
<td>4.3489</td>
<td>0.2497</td>
<td>4.2149</td>
<td>0.2429</td>
<td>4.7885</td>
<td>0.2720</td>
</tr>
<tr>
<td></td>
<td>(-2.04%)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(-5.06%)</td>
<td></td>
<td>(+7.86%)</td>
</tr>
<tr>
<td>90</td>
<td>1.1658</td>
<td>0.2561</td>
<td>1.1119</td>
<td>0.2571</td>
<td>1.1814</td>
<td>0.2574</td>
<td>1.2751</td>
<td>0.2649</td>
</tr>
<tr>
<td></td>
<td>(-4.62%)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(+1.34%)</td>
<td></td>
<td>(+8.38%)</td>
</tr>
</tbody>
</table>

**Notes.** Here, $S_0 = 100$, $r = 0.05$, $T = 0.25$, $\sigma = 0.19$, and $\lambda = 11.5$. The percent difference relative to the empirical distribution price is reported in parentheses.
Poisson processes such as the jump process. Moreover, in numerical experiments they reported that the calibration procedure did a good job of recovering option prices generated (artificially) from Kou’s model.

Because jump-diffusion models have nonunique martingale measures for option pricing, additional criteria are needed to determine a unique measure. The main idea of the approach in Cont and Tankov (2004) is to add to the commonly used least-squares criterion a convex penalization term that yields a unique and stable solution to the inverse problem and for which a gradient-based optimization algorithm can be applied to find the optimal solution. Basically, the additional term minimizes the relative entropy (or Kullback–Leibler divergence) from a prior distribution, which must be specified, in addition to some weights and regularization parameters, for which they provide some guidelines. In terms of the prior distribution, in our setting one possible candidate could be obtained using the box-plot method in Section 4.4.

Although we derived the explicit pricing formula for the jump-diffusion setting, the proposed method can be applied to more general independent increments processes that can be modeled using a multinomial lattice. Furthermore, the methodology developed here should also be applicable in other contexts beyond option pricing, e.g., interest rate derivatives. Path-dependent options present another challenge, and future research could tackle barrier options extending processes that can be modeled using a multinomial lattice framework, which is either a (1) step or a (−1) step.

The following lemma is the basic tool, and it explains the combinatorial meaning of the product of generating functions.

**Lemma 1.** Let $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{L}$ be three sets of lattice paths. If any path $γ$ in $\mathcal{L}$ can be uniquely factored as $α·β = (a_1, a_2, \ldots, a_n)$, then $\mathcal{L}(x) = \mathcal{A}(x)\mathcal{B}(x)$, and for any $α ∈ \mathcal{A}$ and $β ∈ \mathcal{B}$, we have $α·β ∈ \mathcal{L}$, then $\mathcal{L}(x) = \mathcal{A}(x)\mathcal{B}(x)$.

**Proof.** The generating functions for the set $\mathcal{A}$ and $\mathcal{B}$ are, respectively,

$$\mathcal{A}(x) = \sum_{n=0}^{\infty} \#\{\mathcal{A}_n\}x^n \quad \mathcal{B}(x) = \sum_{n=0}^{\infty} \#\{\mathcal{B}_n\}x^n,$$

and where $\mathcal{A}_n = \{α ∈ \mathcal{A} : \ell(α) = n\}$, and $\mathcal{B}_n = \{β ∈ \mathcal{B} : \ell(β) = n\}$. Since any path $γ$ in $\mathcal{L}$ is uniquely factored as $α·β$, $α ∈ \mathcal{A}$, and $β ∈ \mathcal{B}$, and for any $α ∈ \mathcal{A}$ and $β ∈ \mathcal{B}$, we have $α·β ∈ \mathcal{L}$, then we can obtain that

$$\#\{\mathcal{L}_n\} = \sum_{k=0}^{n} \#\{\mathcal{A}_k\}\#\{\mathcal{B}_{n-k}\},$$

where $\mathcal{L}_n = \{γ ∈ \mathcal{L} : \ell(γ) = n\}$. By the usual Cauchy product rule of power series, we have

$$\mathcal{L}(x) = \sum_{n=0}^{\infty} \#\{\mathcal{L}_n\}x^n = \left(\sum_{n=0}^{\infty} \#\{\mathcal{A}_n\}x^n\right)\left(\sum_{n=0}^{\infty} \#\{\mathcal{B}_n\}x^n\right) = \mathcal{A}(x)\mathcal{B}(x). \quad \square$$

Given a set $\mathcal{L}$ of lattice paths, if there exists a subset $\mathcal{A}$ of $\mathcal{L}$ such that every path $α ∈ \mathcal{L}$ can be uniquely factored as $α_1α_2\cdotsα_m$ with $α_i ∈ \mathcal{A}$, then we say that $\mathcal{A}$ is the prime of $\mathcal{L}$.

**Lemma 2.** Let $\mathcal{L}$ be a set of lattice paths. If $\mathcal{A}$ is the prime of $\mathcal{L}$, then

$$\mathcal{L}(x) = \frac{1}{1 − \mathcal{A}(x)}.$$

**Proof.** Let $\hat{\mathcal{L}}_m$ be the set of lattice paths that are uniquely factored as $α_1α_2\cdotsα_m$ with $α_i ∈ \mathcal{A}$. By Lemma 1, it is easy to see $\hat{\mathcal{L}}_m(x) = (\mathcal{A}(x))^m$, where $\hat{\mathcal{L}}_0(x) = 1$ corresponds to the void path. So the generating function for the set $\mathcal{L}$ is

$$\mathcal{L}(x) = \sum_{m=0}^{\infty} \hat{\mathcal{L}}_m(x) = \sum_{m=0}^{\infty} (\mathcal{A}(x))^m = \frac{1}{1 − \mathcal{A}(x)}. \quad \square$$
Appendix B. Support for Derivation of the Discrete Pricing Formula

B.1. Generating Function for Lattice Height Probability

**Lemma 3.** On a multinomial lattice with one-step move probabilities given by (10), the generating function of N-length lattice path probability weight is

\[ G_N(z; \tilde{\mathcal{P}}) = \sum_{i=0}^{N} \binom{N}{i} q^i (1-q)^{N-i} G_N(z; \tilde{\mathcal{P}}; I), \tag{B1} \]

where

\[ G_N(z; \tilde{\mathcal{P}}; I) = \sum_{U=0}^{N-1} \sum_{L=-\infty}^{+\infty} \binom{N-I}{U} p^U (1-p)^{N-I-U} \rho_L^U z^{2U+I-L-N} \tag{B2} \]

is the generating function of N-length lattice path probability weight with I jumps.

**Proof.** Using the generating function techniques of Appendix A, the generating function of the jump-diffusion multinomial lattice is

\[ G(z, x; \tilde{\mathcal{P}}) = \frac{1}{1 - (1-q)pz^1 + (1-q)(1-p)z^{-1} + qz^\infty x}, \]

where x is a variable whose power indicates the length of paths, z’s power indicates the height of paths, and p, q, and \( \rho_L \)'s represent related probability of \( \tilde{\mathcal{P}} \) in (10).

For paths of length N, the generating function is the coefficient of term \( x_N \) in the function above, written as

\[ G_N(z; \tilde{\mathcal{P}}) = \left[ (1-q)pz^1 + (1-q)(1-p)z^{-1} + qz^\infty \sum_{-\infty}^{+\infty} \rho_L z^L \right]^N. \]

We expand this N-steps generating function as follows:

\[ G_N(z; \tilde{\mathcal{P}}) = \sum_{i=0}^{N} \binom{N}{i} [q^i (1-q)^{N-i}] \left( \sum_{-\infty}^{+\infty} \rho_L z^L \right)^i \]

which is Equation (B1), and

\[ G_N(z; \tilde{\mathcal{P}}; I) = [pz^1 + (1-p)z^{-1}]^{N-I} \left( \sum_{-\infty}^{+\infty} \rho_L z^L \right)^I \]

which is Equation (B2). \( \square \)

B.2. Proof of Time-Varying Lattice Height Probability

From the proof of Theorem 1 and Lemma 3, we know that for paths with length N, and probability measure \( \tilde{\mathcal{P}}_{N_1} \) the lattice probability weight \( \mathbb{P}_{N_1}(h_i) \) is the coefficient of the term \( z^{h_i} \) of its generating function \( G_{N_1}(z; \tilde{\mathcal{P}}_{N_1}) \). Therefore, from Lemma 1, during the time interval \( [0, T] \), the generating function under the probability measure \( \tilde{\mathcal{P}} \) is

\[ G_N(z; \tilde{\mathcal{P}}) = \prod_{i=1}^{N} G_N(z; \tilde{\mathcal{P}}_i), \quad \tilde{N} = (N_1, N_2, \ldots, N_t). \]

As a result, the probability weight \( \mathbb{P}_{\tilde{N}}(h) \), which is the coefficient of the \( z^h \) term, is

\[ \mathbb{P}_{\tilde{N}}(h) = \left[ \prod_{i=1}^{N} G_N(z; \tilde{\mathcal{P}}_i) \right] \cdot \prod_{i=1}^{N} \mathbb{P}_{\tilde{N}_i}(h_i). \]

Appendix C. Proof of the Convergence of Discrete Pricing Formula

C.1. Support for the Proof

In Section 3.2, we want to prove Theorem 2:

\[ V_N = \sum_{l=0}^{N} V_l(l)q(l) = \mathbb{E}_\mathbb{P}[V_N(\xi_N)] \xrightarrow{N \to \infty} V_t \]

\[ = \sum_{l=0}^{\infty} V_l(l)q(l) = \mathbb{E}_\mathbb{P}[V_t(\xi)]. \]

Here, we lay the foundation for the proof.

First we take some transforms to make the relationship clearer, and \( h_0 \cong h^\text{call} \). The coefficient of \( e^{-rT}K \) in \( V_N(l) \) is given by

\[ \sum_{h_0 \geq 0} \sum_{U=0}^{N-I} \binom{N-I}{U} p^U (1-p)^{N-I-U} \mathbb{E}[\phi(N, I, p)]. \]

where the superscript L indicates that the expectation \( \mathbb{E} \) is under the distribution of random variable \( L \).

Similarly, the coefficient of \( S_0 \) of \( V_S(l) \) is given by

\[ e^{-\gamma T} \sum_{l=0}^{N-I} \binom{N-I}{U} p^U (1-p)^{N-I-U} \mathbb{E}[\phi(N, I, p)]. \]

\[ = e^{-\gamma T} \mathbb{E}[\phi(N, I, p)]. \]

Downloaded from informs.org by [212.238.245.62] on 31 October 2016, at 07:00. For personal use only, all rights reserved.
We substitute
\[
\frac{\ln(K/S_0)}{\ln u} = L \leq \sum_{i=1}^{I} \ln y_i / \ln u \quad , \quad u = e^{rT}
\]
in the right-hand side of the above and then transform it identically to get
\[
\frac{(h_0 + (N - I) - L)/2 - 1 - (N - I)p}{\sqrt{(N - I)p(1-p)}}
\]
\[
\leq \left[ \frac{h_0 + (N - I) - L}{2} - 1 - (N - I)p \right] / \sqrt{(N - I)p(1-p)}.
\]

We substitute
\[
h_0 = \left[ \frac{\ln(K/S_0)}{\ln u} \right] , \quad L = \sum_{i=1}^{I} \ln y_i / \ln u \quad , \quad u = e^{rT}
\]
in the right-hand side of the above and then transform it identically to get
\[
\frac{(h_0 + (N - I) - L)/2 - 1 - (N - I)p}{\sqrt{(N - I)p(1-p)}}
\]
\[
\leq \left[ \frac{(h_0 + (N - I) - L)/2 - 1 - (N - I)p}{\sqrt{(N - I)p(1-p)}} \right]
\]
\[
\leq \left[ \frac{h_0 + (N - I) - L}{2} - 1 - (N - I)p \right] / \sqrt{(N - I)p(1-p)}.
\]
since
\[ 2\bar{p} - 1 = \sqrt{r - \lambda (\mu - 1) + \sigma^2/2} + o(\tau). \]

Analogous to (i), by applying the dominated convergence theorem,
\[ \mathbb{E}^N\{u^I\psi(N, I, \bar{p})\} \xrightarrow{N \to \infty} \mathbb{E}^1\left[ \prod_{i=1}^I y_i \Psi(I, \sigma^2/2) \right]. \]

On the other hand, by taking a Taylor series expansion in \( \tau \), it is easy to see that
\[ e^{-r\tau} \mathbb{E}^N\{u^I\} \xrightarrow{N \to \infty} e^{-r(N-1)\tau}, \]

establishing the convergence of (ii).

Combining (i) and (ii), we conclude that \( V_n(I) \xrightarrow{N \to \infty} V(I) \).

**Lemma 5.** \( V_n(I) \) and \( V(I) \) are bounded functions for fixed \( I \). More specifically, there exists \( \exists n \in \mathbb{Z}^+ \) such that for all \( N > n \), there exists \( A > 0 \) such that
\[ |V_n(I)| < A \mu^I, \quad |V(I)| < A \mu^I. \]

**Proof.** (i) Boundedness of \( V_n(I) \): Because of inequalities \( 0 \leq \Psi(I, \sigma^2/2) \leq 1 \) and \( 0 \leq \Psi(I, -\sigma^2/2) \leq 1 \), we have
\[ 0 \leq \mathbb{E}^1\{\Psi(I, -\sigma^2/2)\} \leq 1, \]

\[ 0 \leq \mathbb{E}^1\left[ \prod_{i=1}^I y_i \Psi(I, \sigma^2/2) \right] \leq \mathbb{E}^1\left[ \prod_{i=1}^I y_i \right], \]

( noticing that \( y_i \) is positive).

Therefore,
\[ |V_n(I)| = \left| S_0 e^{-\lambda(N-1)I} \mathbb{E}^1\left[ \prod_{i=1}^I y_i \Psi(I, \sigma^2/2) \right] e^{-r(N-1)I} \mathbb{E}^1\{\Psi(I, -\sigma^2/2)\} \right| \leq |S_0 e^{-\lambda(N-1)I} \mathbb{E}^1\left[ \prod_{i=1}^I y_i \right] |. \]

(ii) Boundedness of \( V(I) \): As a result of \( 0 \leq \psi(N, I, p) \leq 1 \), \( 0 \leq \psi(N, I, \bar{p}) \leq 1 \), we have
\[ 0 \leq \mathbb{E}^1\{\psi(N, I, p)\} \leq 1, \]

\[ 0 \leq \mathbb{E}^1\left[ u^I\psi(N, I, \bar{p})\right] \leq \mathbb{E}^1\{u^I\}. \]

Therefore,
\[ |V_n(I)| = |S_0 e^{-r\tau(N-1)} \mathbb{E}^1\{u^I\psi(N, I, p)\} - e^{-r\tau N} \mathbb{E}^1\{\psi(N, I, p)\}| \leq |S_0 e^{-r\tau(N-1)} \mathbb{E}^1\{u^I\}| \leq |S_0 e^{-r\tau(N-1)} + \theta| (\mu^I + \theta^I)|, \]

(from the proof of Lemma 4, we know that \( u^I \to \prod_{i=1}^I y_i, \quad e^{-r\tau(N-1)} \to e^{-\lambda(N-1)\tau} \))

Hence there exists \( n \in \mathbb{Z}^+ \) such that when \( N > n \), there exists \( A > 0 \) such that
\[ |V_n(I)| < A \mu^I, \quad |V(I)| < A \mu^I. \]

**Lemma 6.** For any \( \varepsilon > 0 \), there exists \( N_0 \in \mathbb{Z}^+ \) such that for all \( N > N_0 \),
\[ \sum_{I=0}^N V_n(I)q_n(I) < \varepsilon. \]

**Proof.** Pick \( \varepsilon > 0 \) arbitrary and fixed. First, from Lemma 5, there exists \( N_1 \in \mathbb{Z}^+ \) and \( A \in \mathbb{R}^+ \) such that for any \( N > N_1 \),
\[ V_n(I) < A \mu^I. \]

Second, as \( qN = \lambda T \) is constant, we have \( q_n(I) \to q(I) \), so there exists \( N_2 \in \mathbb{Z}^+ \) such that for any \( N > N_2 \),
\[ q_n(I) < 2q(I) = 2e^{-\lambda T} \frac{(\lambda T)^I}{I!}. \]

Next, from the Taylor’s series expansion of \( e^x \), we know the series \( \sum_{I=0}^\infty ((\lambda T)^I/I!) \) is convergent. Hence for \( \varepsilon > 0 \), there exist \( N_3 \in \mathbb{Z}^+ \) such that for any \( N > N_3 \),
\[ \sum_{I=0}^N \frac{N!}{(\lambda T)^I} < \sum_{I=0}^N \frac{(\lambda T)^I}{I!} < 2Ae^{-\lambda T}. \]

Therefore, for \( \varepsilon > 0 \), there exists \( N_0 = \max\{N_1, N_2, N_3\} \) such that \( N > N_0 \),
\[ \sum_{I=0}^N V_n(I)q_n(I) < \sum_{I=0}^N A \mu^I \cdot 2e^{-\lambda T} \frac{(\lambda T)^I}{I!} = 2Ae^{-\lambda T} \sum_{I=0}^N \frac{(\lambda T)^I}{I!} < \varepsilon. \]

**Lemma 7.** \( \mathbb{E}_n\{V_n(\xi_n)\} \xrightarrow{N \to \infty} \mathbb{E}_n\{V(\xi)\}. \)

**Proof.** First, we know that \( q_n(I) \to q(I) \), as \( qN = \lambda T \) is constant; thus we have \( \xi_n \to \xi \).

And \( V_n(I) \) is a function whose domain is \( \mathbb{N} \). We can expand its domain to \( \mathbb{R}^+ \) just by naturally adjoining the points \((I, V_n(I)) \) and \((I+1, V_n(I+1))\) on the graph of \( V_n(I) \). After this intuitive operation, we obtain a continuous function \( V_n(x) \). Hence for a continuous function \( V(x) \),
\[ \mathbb{E}_n\{V_n(\xi_n)\} \xrightarrow{N \to \infty} \mathbb{E}_n\{V(\xi)\}. \]

**C.2. Proof of Theorem 2**

**Proof.** Pick \( \varepsilon > 0 \) arbitrary but fixed.

From Lemma 7, \( \mathbb{E}_n\{V(\xi_n)\} \xrightarrow{N \to \infty} \mathbb{E}_n\{V(\xi)\} \), so there exists \( n_1 \in \mathbb{Z}^+ \) such that for any \( N > n_1 \),
\[ |\mathbb{E}[V(\xi_n) - V(\xi)]| < \varepsilon/3. \]

From Lemma 6, there exists \( n_2 \in \mathbb{Z}^+ \) such that for any \( N > n_2 \),
\[ \left| \sum_{I=0}^N V_n(I)q_n(I) \right| < \varepsilon/3. \]
And from Lemma 4, \( V_n(I) \rightarrow_{N \to \infty} V(I) \) for fixed \( I \in \mathbb{N} \), so there exists \( N(I) \in \mathbb{N} \) such that for all \( N \geq N(I) \),

\[
|V_n(I) - V(I)| < \frac{\varepsilon}{3}.
\]

Set \( n_3 = \max_{2 \leq I \leq N}[N(I)] \); then for \( N > n_3 \),

\[
\frac{\sum_{l=0}^{n_2} |V_n(I) - V(I)| q_n(I)}{\sum_{l=0}^{n_2} q_n(I)} \leq \frac{\varepsilon}{3} \sum_{l=0}^{n_2} q_n(I) \leq \frac{\varepsilon}{3}.
\]

Let \( n = \max\{n_1, n_2, n_3\} \); then for \( N > n \),

\[
|E[V_n(\xi_N) - V_\xi(\xi)]| \leq |E[V_n(\xi_N) - V_\xi(\xi)]| + |E[V_\xi(\xi_N) - V_\xi(\xi)]| \leq \frac{\varepsilon}{3} + \frac{\varepsilon}{3} + \frac{\varepsilon}{3} = \varepsilon.
\]

Therefore, \( E[V_n(\xi_N)] \rightarrow_{N \to \infty} E[V_\xi(\xi)] \), \( V_n \rightarrow_{N \to \infty} V_\xi \). \( \square \)

### Appendix D. Computational Implementation

Implementation of the pricing formulas requires specifying the truncation parameter \( I_0 \) on the maximum number of jumps in a lattice path. In the latter case, if the original jump-size distribution is continuous or discrete with infinite support, then an approximation procedure is required. Furthermore, the size of the lattice must also be bounded, which is controlled by two additional parameters, denoted by \( M_0 \) and \( R \). This appendix discusses these two issues and offers various implementation suggestions that were used in the numerical experiments reported in the paper.

#### D.1. Approximation of Jump-Size Distribution

The algorithm requires a finite discrete jump-size distribution. Here, we describe methods for discretizing a continuous distribution and for truncating an infinite discrete distribution.

1. **Discretizing a continuous distribution**

Partition the range for the jump size into a set of small intervals to obtain the discrete distribution

\[
P(Y = y^i) = P(J = j) = F\left(j + \frac{1}{2}\right) - F\left(j - \frac{1}{2}\right).
\]

Without loss of generality, we write the discrete distribution as \( \{q_j\}_{j=-\infty}^{\infty} \), where the value of \( q_j \) could be 0 if the distribution is finite. To be consistent with the model setting, we require the probability of a small jump (±1) to be 0 and move the original mass at \( q_{+1} \) to \( q_0 \) to get the distribution \( \tilde{Y} : \{\tilde{\rho}_j\}_{j=-\infty}^{\infty} \), where \( \rho_{+1} = 0, \rho_0 = 0, \rho_{-1} = \rho_{-2} = \rho_2 = \rho_3 = \cdots, \rho_j = q_j, \forall j \neq 0, \pm 1. \)

2. **Truncating an infinite distribution**

Approximate the left tail and/or right tail of the distribution \( \hat{Y} : \{\rho_j\}_{j=-\infty}^{\infty} \) using a single bucket as follows:

\[
\tilde{Y} : \tilde{\rho}_j = \begin{cases} \sum_{j=0}^{\lfloor N/\ln u \rfloor - M_0 - 1} \rho_j & f = M^- \leq \lfloor N/\ln u \rfloor - M_0 - 1, \\ [\alpha/\ln u] - M_0 \leq j & \sum_{j=0}^{\lfloor N/\ln u \rfloor + M_0} \rho_j & 0, \\ \sum_{j=M^+}^{\lfloor N/\ln u \rfloor + M_0} \rho_j & f = M^+ \leq \lfloor N/\ln u \rfloor + M_0 + R, \\ 0 & \text{otherwise;} \end{cases}
\]

where \( \alpha \) is the expectation of log jump for the original jump-size distribution \( \hat{Y} \) or \( \tilde{Y} \). The truncation parameter \( M_0 \) is chosen to guarantee that the shape of \( \tilde{Y} \) is close enough to the original jump-size distribution \( \hat{Y} \) or \( \tilde{Y} \), and the integer \( R > 0 \) minimizes \( |\tilde{\delta} - \hat{\delta}| \) such that the variance \( \tilde{\delta} \) of \( \tilde{Y} \) is close enough to \( \hat{\delta} \), the variance of \( \hat{Y} \) or \( \tilde{Y} \).

As an example, for a normal distribution \( N(\alpha, \delta^2) \), we use the 3\( \delta \)-principle to derive

\[
M_0 = \left\lceil \frac{3\delta}{\ln u} \right\rceil,
\]

where the chance of jump size greater than 3\( \delta \) under a normal distribution case is less than \( 10^{-3} \). For the double-exponential distribution, we can also calculate the corresponding \( M_0 \) for each \( N \).

#### D.2. Determination of the Maximum Number of Jumps \( I_0 \)

From Lemma 6, for an appropriate \( I_0 \), we have

\[
\sum_{l=0}^{N} V_n(l)q_n(l) \rightarrow 0;
\]

i.e., lattice paths that have more than \( I_0 \) jumps have negligible impact on the option value. Thus we turn to the determination of \( I_0 \).

For the Poisson process, we have that \( qN = \lambda T \) is a constant, and \( \beta_r(N, q) \rightarrow 1(\lambda T) \). Therefore, we choose the maximum number of jumps \( I_0 \) by bounding the tail probability of the Poisson distribution \( 1(\lambda T) \); i.e., \( I_0 \) is the smallest integer satisfying

\[
\sum_{l=I_0+1}^{\infty} e^{-\lambda T}(\lambda T)^l/l < \varepsilon,
\]

which leads to (16). Note that by (16), the value of the maximum number of jumps \( I_0 \) is determined by \( \lambda \) and \( T \) only; i.e., it is independent of \( N \).

We show an example under the jump-diffusion process described in Section 4.1 for a particular time step \( N \) to see the computational implementation.

1. Taking \( \varepsilon = 10^{-3} \) in (16) gives \( I_0 = 12 \).
2. For each time step \( N \), do the following operations. Here, we illustrate for \( N = 2,000 \).
3. By definition of \( \tau \) and relationship (7), calculate \( u = e^{\sigma \sqrt{\tau}} = 1.00354 \).
Table D.1  European Put Option Prices with Different $l_0$ Under Log-Normal Jump-Size Distribution in $\lambda(-0.025, 0.05)$
$\sigma^2 = 0.05, S_0 = 40, r = 0.08, T = 0.5, \lambda = 5, and K = 45$ (Time Steps $N = 2,000$)

<table>
<thead>
<tr>
<th>$l_0$</th>
<th>Option price</th>
<th>$l_0$</th>
<th>Option price</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>6.7392</td>
<td>5</td>
<td>7.4283</td>
</tr>
<tr>
<td>6</td>
<td>7.7351</td>
<td>7</td>
<td>7.8511</td>
</tr>
<tr>
<td>8</td>
<td>7.8892</td>
<td>9</td>
<td>7.9002</td>
</tr>
<tr>
<td>10</td>
<td>7.9031</td>
<td>11</td>
<td>7.9038</td>
</tr>
<tr>
<td>12</td>
<td>7.9039</td>
<td>13</td>
<td>7.9039</td>
</tr>
<tr>
<td>14</td>
<td>7.9040</td>
<td>15</td>
<td>7.9040</td>
</tr>
</tbody>
</table>

Figure D.1  (Color online) European Put Option Prices with a Different Maximum Number of Jumps, Under Jump-Size Distribution in $\lambda(-0.025, 0.05)$, and Parameters $\sigma^2 = 0.05, S_0 = 40, r = 0.08, T = 0.5, \lambda = 5, and K = 45$ (Time Steps $N = 2,000$)

4. Following Appendix D.1 gives $M_0 = 190, R = 19, and -8 < \alpha/\ln u < -7$; then the corresponding discrete jump-size distribution is written as follows:

$$\tilde{\nu} = \begin{cases} 
F((-198 - \frac{j}{2}) \ln u) & j = -217, \\
F((j + \frac{1}{2}) \ln u) - F((j - \frac{1}{2}) \ln u) & -198 \leq j < -1, \\
F((1 + \frac{1}{2}) \ln u) - F((-1 - \frac{1}{2}) \ln u) & j = 0, \\
F((j + \frac{1}{2}) \ln u) - F((j - \frac{1}{2}) \ln u) & 1 \leq j \leq 183, \\
1 - F((183 + \frac{1}{2}) \ln u) & j = 202, \\
0 & \text{otherwise}.
\end{cases}$$

5. Compute using the main pricing formulas after calculation of $q = 0.00125$ and $p = 0.501948$ by Equation (7).

To see the effect of a different maximum number of jumps on the option price, we show the resulting option price with different $l_0$ under this set of parameters in Table D.1 and Figure D.1.

It can be seen from Table D.1 and Figure D.1 that the option price converges within $10^{-4}$ precision for $l_0 = 12$.

Appendix E. Pseudocode

In MATLAB

```matlab
h1 = min(l0 * (-M + 1) - N, -N);
h2 = max(l0 * (M - 1) + N, N);
tm = (h2 - h1 + 2) / 2;
I = [1:10];
G = zeros(l0 + 1, h2 - h1 + 1);
for i = 1: length(I)
    U[I] = [0: N - I(i)];
end
Q = ComBinN(N, I, q);
```

for $i = 1$: length(I)
    temp = ComBinN(N - I(i), cell2mat(U[I]), p);
    temp0 = zeros(1, length(temp));
    P = [temp; temp0];
    P = reshape(P, [1, numel(P)]);
    P(end) = [];
    if I(i) == 0
        G(1, tm - N: tm + N) = P;
        y = Y;
    else
        G(I(i) + 1, tm - (M - 1) * I(i) - N: tm + (M - 1) * I(i) + N) = conv(y, P);
        y = conv(y, Y);
    end
end
PX = Q * G;
su = exp((hcall * 1: h2)) * (log(u) * ones(1, h2 - hcall + 1));
Vcall = exp(-r * T) * (FX(1, hcall - h1 + 1: h2 - h1 + 1) * (SO * su - K));

In Mathematica

```mathematica
J[z_] := Sum[r[i]*z^i, {i, -M0 - R, M0 + R}];
qu[i] := Table[Binomial[n, i]*q[i - 1 - q]^n - i, {i, 0, l0}];
pu[i] := Table[Binomial[n - i, u]*p[i - 1 - p]^n - i, {u, 0, n - i}];
nu[i] := Table[Total[pu[i]*Table[r[i]*z[i], {z, 1, 0, l0}]], {i, 0, l0}];
```

5. References


