Open Journal of Statistics, 2017, 7, 446-458
http://www.scirp.org/journal/ojs
ISSN Online: 2161-7198
ISSN Print: 2161-718X

Numerical Methods for Discrete Double Barrier Option Pricing Based on Merton Jump Diffusion Model

Mingjia Li

Jinan University, Guangzhou, China
Email: limingjia51@163.com

How to cite this paper: Li, M.J. (2017) Numerical Methods for Discrete Double Barrier Option Pricing Based on Merton Jump Diffusion Model. *Open Journal of Statistics*, **7**, 446-458.
https://doi.org/10.4236/ojs.2017.73032

Received: May 8, 2017
Accepted: June 9, 2017
Published: June 12, 2017

Abstract

As a kind of weak-path dependent options, barrier options are an important kind of exotic options. Because the pricing formula for pricing barrier options with discrete observations cannot avoid computing a high dimensional integral, numerical calculation is time-consuming. In the current studies, some scholars just obtained theoretical derivation, or gave some simulation calculations. Others impose underlying assets on some strong assumptions, for example, a lot of calculations are based on the Black-Scholes model. This thesis considers Merton jump diffusion model as the basic model to derive the pricing formula of discrete double barrier option; numerical calculation method is used to approximate the continuous convolution by calculating discrete convolution. Then we compare the results of theoretical calculation with simulation results by Monte Carlo method, to verify their efficiency and accuracy. By comparing the results of degeneration constant parameter model with the results of previous models we verified the calculation method is correct indirectly. Compared with the Monte Carlo simulation method, the numerical results are stable. Even if we assume the simulation results are accurate, the time consumed by the numerical method to achieve the same accuracy is much less than the Monte Carlo simulation method.

Keywords

Discrete Double Barrier Option, Merton Jump Diffusion Model, Discrete Convolution, Monte Carlo Method

1. Introduction

Options as a kind of important financial derivatives have been developed rapidly in recent decades. With the increase of investment demands and the progress of theoretical level, all kinds of exotic options were born. As an important path dependent exotic option, barrier option was accounted for nearly half of the share. Their research has a very important practical significance. So, new exotic option pricing problem has become an important research topic.

Black and Scholes [1] set up the B-S model and get the pricing formula of the call option. Common barrier options pricing has a long history. It was first studied by Merton [2] on the down and out call option pricing. After this period, Reimer and Sandmann [3] use the binary tree model research barriers options. Then Gao, Huang and Subrahmanyam [4] pushed forward the pricing analysis of the American barrier option by using the decomposition technique, and Dai and Kwok [5] has got the pricing formula of American down and in call option. The model of the underlying asset is also developed from the original B-S model to a variety of more complex models. Such as Wang, Du [6] obtained pricing formula in the class of barrier options under jump diffusion model, and Xie [7] put forward the constant elasticity of variance (CEV) pricing barrier option process. However, these developments are based on the constant parameter model. Farnoosh, Sobhani, Rezazadeh [8] proposed a method based on the time dependent parametric B-S model of the barrier options to extend the research of this topic to another direction.

Looking at the development of barrier option pricing, the barrier option pricing theory is from a study based on the continuous observation idealized model to the effective discrete observation model, and the stochastic model of the target stock is also developed from B-S model with constant parameters to model with time-dependent parameter, or constant parameter jump diffusion model, CEV model, SVJD model and so on. In order to make the scope of application more extensive, the goal of this paper is the further development for the numerical algorithm of the option pricing based on Merton jump diffusion model.

In the numerical calculation section, having a wide range of applications, Monte Carlo simulation method is certainly one of the alternative calculation methods. But it is often accompanied by a wide range of accuracy and low computational efficiency. As traditional numerical algorithm performs badly in non-smooth solutions, Golbabai, Ballestraand Ahmadian used finite element method (FEM) to improve orders of convergence in [9]. In [10], Ahmadian and Ballestra found it also performs well under a constant elasticity of variance model with jump diffusion. For some models, the other way to get numerical method is directly starting from the analytic solution. In [8], Farnoosh, Sobhani, Rezazadeh is proposed that for time dependent numerical parameters of barrier options based on B-S model an analytical solution can be obtained. They started from solving partial differential equations. By constructing a heat conduction equation, they finally got the analytical solution. However, it still can't avoid calculate a multiple integral of a high dimension. Using the numerical solution of the Romberg extrapolation algorithm to calculate is taking time and precision into account. But consider from the efficiency of the integral calculation, it is not as good as quadrature method which been used by Milev and Tagliani in general

B-S model [11]. The difference is probably cause by the influence of boundary. The quadrature method used on general B-S model is able to extend to more complex model. Furthermore, for time dependent parameters, it also can achieve good results.

In the second section, we will consider knock-out call option as an example to introduce risk neutral pricing based on Merton jump diffusion model. Finally, we obtain an analytical formula of the discrete double barrier option.

In the third section, we will introduce the application of quadrature method to calculate the high dimensional integrals generation generated in the second section. We also introduce Monte Carlo simulation method to compute the result for comparison.

In the fourth section, the computational efficiency of the numerical method is compared with samples in literatures and results of simulation method.

In the conclusion, we note that this calculation method is significantly better in accuracy and efficiency, and there is a certain promotion potential.

2. Discrete Double Barrier Options Model

We assume the European discrete knock-out double barrier option has the following conditions: 1) European options, *i.e.*, option can only be exercised at maturity; 2) The time for observation has is set in advance, $\sigma_J^2 \quad = T$ in the subsequent derivation, we assume observation time interval is consistent. For different observation intervals, the subsequent theory still holds; 3) All the observation time have the same U and L as barrier. When the stock price at the observation time higher than U or lower than L, the option knock-out; 4) Ignore the transaction costs and taxes. It means if the option hasn't knocked out at observation time, at the time of maturity, the option value is $\max\left(S_T - K, 0\right)$, $L \le K \le U, L \le S_0 \le U$; 5) the stock price can change continuously.

Merton jump diffusion model assumes that the stock price S_t is in accordance with the following stochastic differential equation,

$$\frac{\mathrm{d}S_t}{S_{t-}} = \left(\mu - \lambda E\left[J_t\right]\right)\mathrm{d}t + \sigma \mathrm{d}W_t + J_t \mathrm{d}N_t, \tag{1}$$

where $\ln\left(1+J_t\right) \sim N\left(\mu_J, \sigma_J^2\right)$, W_t is standard Wiener process, N_t is Poisson process with intensity λ. We denote $Y_t = \ln\left(1+J_t\right)$, $\gamma = \mu - \lambda E\left[J_t\right] - \frac{1}{2}\sigma^2$. By Itô Lemma, we obtain

$$\mathrm{d}\ln S_t = \gamma \mathrm{d}t + \sigma \mathrm{d}W_t + Y_t \mathrm{d}N_t. \tag{2}$$

We denote $A_i = \left\{S_{T_i} \in \left[L, U\right]\right\}, i = 1, 2, \cdots, m$. In risk-neutral measure, if we assume risk-free interest rate r is constant. (Thus, the drift μ is replaced by the interest rate r.) By the definition of the option its price at the beginning time is equal to

$$\mathrm{e}^{-rT} E\left[\max\left(S_T - K, 0\right), 1_{A_1} 1_{A_2} \cdots 1_{A_m}\right]. \tag{3}$$

We can calculate this formula to get the analytical formula of the discrete double barrier option.

Theorem 2.1. The value of option with above assume is given by the following integral

$$V(S,T)=\int_{(x_1,x_2,\cdots,x_m)\in D}\left(L\mathrm{e}^{x_1+x_2+\cdots+x_m}-K\right)\tilde{f}(x_1,x_2,\cdots,x_m)\mathrm{d}x_m\cdots\mathrm{d}x_2\mathrm{d}x_1, \quad (4)$$

where

$$D=\left\{x_1,x_2,\cdots,x_m \mid x_1,x_1+x_2,\cdots,x_1+x_2+\cdots+x_{m-1}\in\left[0,\ln\frac{U}{L}\right],\right.$$
$$\left.x_1+x_2+\cdots+x_m\in\left[\ln\frac{K}{L},\ln\frac{U}{L}\right]\right\},$$

$$\tilde{f}(x_1,x_2,\cdots,x_m)=h\left(x_1-\ln\frac{L}{S_0}\right)\prod_{i=2}^{m}h(x_i),$$

$$h(x)=\sum_{i=0}^{\infty}\frac{(\lambda T/m)^i}{i!}\mathrm{e}^{-\frac{\lambda T}{m}}\phi\left(\frac{\lambda T}{m}+i\mu_J,\frac{\sigma^2T}{m}+i\sigma_J^2\right),$$

$\phi(\mu,\sigma^2)$ is the p.d.f. of $N(\mu,\sigma^2)$.

Proof. Step 1. Assuming that the p.d.f. of random variable is known, we calculate the analytical result.

We denote $\tilde{\xi}_i=\ln S_{T_i}-\ln S_{T_{i-1}}, i=1,2,\cdots,m$, we have $S_{T_i}=S_0\mathrm{e}^{\sum_{j=1}^{i}\tilde{\xi}_j}$. Let $\xi_1=\ln S_0-\ln L+\tilde{\xi}_1, \xi_i=\tilde{\xi}_i, i=2,3,\cdots,m$, We get $S_{T_i}=L\mathrm{e}^{\sum_{j=1}^{i}\xi_j}$. For $i=1,2,\cdots,m$,

$$A_i=\left\{L\le S_{T_i}\le U\right\}=\left\{\ln L\le\ln S_{T_i}\le\ln U\right\}=\left\{0\le\sum_{j=1}^{i}\xi_j\le\ln\frac{U}{L}\right\}.$$

From Equation (4), the value of this barrier option is,

$$\begin{aligned}
&V(S,T)\\
&=\mathrm{e}^{-rT}E\left[\max(S_T-K,0)1_{A_1}1_{A_2}\cdots1_{A_m}\right]\\
&=\mathrm{e}^{-rT}E\left[\max\left(L\mathrm{e}^{\sum_{j=1}^{m}\xi_j}-K,0\right)1_{\left\{0\le\sum_{j=1}^{1}\xi_j\le\ln\frac{U}{L}\right\}}1_{\left\{0\le\sum_{j=1}^{2}\xi_j\le\ln\frac{U}{L}\right\}}\cdots1_{\left\{0\le\sum_{j=1}^{m}\xi_j\le\ln\frac{U}{L}\right\}}\right] \quad (5)\\
&=\mathrm{e}^{-rT}E\left[\left(L\mathrm{e}^{\sum_{j=1}^{m}\xi_j}-K\right)1_{\left\{0\le\sum_{j=1}^{1}\xi_j\le\ln\frac{U}{L}\right\}}1_{\left\{0\le\sum_{j=1}^{2}\xi_j\le\ln\frac{U}{L}\right\}}\cdots1_{\left\{\ln\frac{K}{L}\le\sum_{j=1}^{m}\xi_j\le\ln\frac{U}{L}\right\}}\right]\\
&=\mathrm{e}^{-rT}\int_{(x_1,x_2,\cdots,x_m)\in D}\left(L\mathrm{e}^{x_1+x_2+\cdots+x_m}-K\right)\prod_{i=1}^{m}h_i(x_i)\mathrm{d}x_m\cdots\mathrm{d}x_2\mathrm{d}x_1,
\end{aligned}$$

where

$$D=\left\{x_1,x_2,\cdots,x_m \mid x_1,x_1+x_2,\cdots,x_1+x_2+\cdots+x_{m-1}\in\left[0,\ln\frac{U}{L}\right],\right.$$
$$\left.x_1+x_2+\cdots+x_m\in\left[\ln\frac{K}{L},\ln\frac{U}{L}\right]\right\},$$

$h(x_i)$ is p.d.f. of $\xi_i, i=2,3,\cdots,m$.

Step 2. Calculate $h(x_i)$.

For $i=2,3,\cdots,m$ from Equation (2), We know ξ_i is sum of independent random variables including a normal random variable $\eta \sim N\left(\frac{\gamma T}{m}, \frac{\sigma^2 T}{m}\right)$ and a series of normal random variables $\eta_k \sim N\left(\mu_J, \sigma_J^2\right)$. The length of the series is follow a Poisson distribution with intensity $\lambda T/m$. As we know while the length is *j*, sum of these variables $\xi_i \sim N\left(\frac{\gamma T}{m} + j\mu_J, \frac{\sigma^2 T}{m} + j\sigma_J^2\right)$, we get the p.d.f. of ξ_i,

$$h_i = \sum_{j=0}^{\infty} \frac{(\lambda T/m)^j}{j!} \mathrm{e}^{-\frac{\lambda T}{m}} \phi\left(\frac{\gamma T}{m} + j\mu_J, \frac{\sigma^2 T}{m} + j\sigma_J^2\right), \tag{6}$$

where $\phi\left(\mu, \sigma^2\right)$ is the p.d.f. of $N\left(\mu, \sigma^2\right)$. Because $h_i(x)$ has no relation with *i*, in $i = 2, 3, \cdots, m$, we denote $h_i(x)$ with $h(x)$. In particular, as there is a constant shift in ξ_1, $h_1(x) = h\left(x - \ln \frac{L}{S_0}\right)$.

Plug $h(x)$ into Equation (5), we finally obtain (4). □

Specially, if parameters is change to different constant in different monitor interval, $h_i(x)$ will be different while they can be calculate in the same method by different parameters.

3. Numerical Valuation Algorithm

From section 2 we know random variables $\xi_i, i = 1, 2, \cdots, m$ decide the option value at $t = T$. As we have already known the distribution of ξ_i, we can use Monte Carlo method to simulate ξ_i and get some possible option value at $t = T$. By weak law of large numbers, the means of the Discounting of these price is converging to the option price at $t = 0$. However, we can't expect it as an efficient selection.

Since we have got the analytical formula of the discrete double barrier option. We can also calculate the numerical result by calculate the analytical formula. While there is a High dimensional multiple integral, we should anticipate some tricks. As the integration has a break at the boundary of the integral area, we compare and find that Milev's method to discrete the integration, solve directly perform a better precision than the extrapolation method, such as the Romberg extrapolation algorithm.

We use discrete method to calculate the High dimensional multiple integral by calculate an approximate discrete convolution. Monte Carlo method is also used for compared.

Firstly, we introduce the discrete method. Let $d = \ln \frac{U}{L} \Big/ n$, where *n* is an integer number. We used discrete random variable $\xi_{i,n}$ to approximate ξ_i, $\xi_{i,n}$ satisfy

$$P\left(\xi_{i,n} = kd\right) = p_{k,n}, \tag{7}$$

where

$$p_{k,n} = P\left(\left(k-0.5\right)d \le \xi_2 < \left(k+0.5\right)d\right), \tag{8}$$

$k = \cdots,-1,0,1,2,\cdots, i = 2,3,\cdots,m$. Specially, we discrete random variable $\xi_{1,n}$ to approximate ξ_1, $\xi_{1,n}$ satisfy

$$P\left(\xi_{i,n} = kd\right) = \tilde{p}_{k,n}, \tag{9}$$

where

$$\tilde{p}_{k,n} = P\left(\left(k-0.5\right)d \le \xi_1 < \left(k+0.5\right)d\right). \tag{10}$$

From Equation (5), we know

$$V\left(S,T\right) = \mathrm{e}^{-rT} E\left[\max\left(L\mathrm{e}^{\sum_{j=1}^{m}\xi_j} - K,0\right) 1_{\left\{0\le\sum_{j=1}^{1}\xi_j\le\ln\frac{U}{L}\right\}} 1_{\left\{0\le\sum_{j=1}^{2}\xi_j\le\ln\frac{U}{L}\right\}} \cdots 1_{\left\{0\le\sum_{j=1}^{m}\xi_j\le\ln\frac{U}{L}\right\}}\right]. \tag{11}$$

So we can use $\tilde{V}_n\left(S,T\right)$ to approximate $V\left(S,T\right)$, where

$$\tilde{V}_n\left(S,T\right) = \mathrm{e}^{-rT} E\left[\max\left(L\mathrm{e}^{\sum_{j=1}^{m}\xi_{j,n}} - K,0\right) 1_{\left\{0\le\sum_{j=1}^{1}\xi_{j,n}\le\ln\frac{U}{L}\right\}} 1_{\left\{0\le\sum_{j=1}^{2}\xi_{j,n}\le\ln\frac{U}{L}\right\}} \cdots 1_{\left\{0\le\sum_{j=1}^{m}\xi_{j,n}\le\ln\frac{U}{L}\right\}}\right]. \tag{12}$$

We denote $\zeta_{k,n} = \sum_{j=1}^{k}\xi_{j,n} 1_{\left\{0<\sum_{j=1}^{1}\xi_{j,n}\le\ln\frac{U}{L}\right\}} \cdots 1_{\left\{0<\sum_{j=1}^{k}\xi_{j,n}\le\ln\frac{U}{L}\right\}}$, $k = 1,2,\cdots,m$. From Equation (12), we know

$$\tilde{V}_n\left(S,T\right) = \mathrm{e}^{-rT} E\left(\max\left(L\mathrm{e}^{\zeta_{m,n}} - K,0\right)\right). \tag{13}$$

So we can calculate $\tilde{V}_n\left(S,T\right)$ by calculate $\zeta_{m,n}$. From the definition of $\zeta_{1,n}$, we know

$$\zeta_{1,n} = \xi_{1,n} 1_{\left\{0<\xi_{j,n}\le\ln\frac{U}{L}\right\}}. \tag{14}$$

So the value of $\zeta_{1,n}$ may be $0,d,2d,\cdots,nd$, and satisfy

$$P\left(\zeta_{1,n} = kd\right) = \begin{cases} \tilde{p}_{k,n}, & k = 1,2,\cdots,n \\ \hat{p}_{0,n}, & k = 0 \end{cases} \tag{15}$$

where $\hat{p}_{0,n} = 1 - \sum_{k=1}^{n}\tilde{p}_{k,n}$. Notice that $\zeta_{1,n} \ge 0$. $\zeta_{1,n} > 0$ is equivalent to $1_{\left\{0<\xi_{1,n}\le\ln\frac{U}{L}\right\}} = 1$. $\zeta_{1,n} = 0$ is equivalent to $1_{\left\{0<\xi_{1,n}\le\ln\frac{U}{L}\right\}} = 0$. In fact, we have the following lemma:

Lemma 3.1. For $k = 1,2,\cdots,m, \zeta_{k,n} \ge 0$, and

$$\zeta_{k,n} = 0 \Leftrightarrow 1_{\left\{0<\sum_{j=1}^{1}\xi_{j,n}\le\ln\frac{U}{L}\right\}} 1_{\left\{0<\sum_{j=1}^{2}\xi_{j,n}\le\ln\frac{U}{L}\right\}} \cdots 1_{\left\{0<\sum_{j=1}^{k}\xi_{j,n}\le\ln\frac{U}{L}\right\}} = 0, \tag{16}$$

$$\zeta_{k,n} > 0 \Leftrightarrow 1_{\left\{0<\sum_{j=1}^{1}\xi_{j,n}\le\ln\frac{U}{L}\right\}} 1_{\left\{0<\sum_{j=1}^{2}\xi_{j,n}\le\ln\frac{U}{L}\right\}} \cdots 1_{\left\{0<\sum_{j=1}^{k}\xi_{j,n}\le\ln\frac{U}{L}\right\}} = 1. \tag{17}$$

If $\zeta_{k,n} > 0$, then $\zeta_{k,n} = \sum_{j=1}^{k} \xi_{j,n}$.

Proof. From the definition of $\zeta_{k,n}$, it can be easily proved.□

For $k = 2, 3, \cdots, m$, by lemma 3.1, we have

$$\zeta_{k,n} = \begin{cases} \sum_{j=1}^{k} \xi_{j,n} 1_{\left\{0<\sum_{j=1}^{k}\xi_{j,n} \le \ln\frac{U}{L}\right\}}, & 1_{\left\{0<\sum_{j=1}^{1}\xi_{j,n} \le \ln\frac{U}{L}\right\}} \cdots 1_{\left\{0<\sum_{j=1}^{k-1}\xi_{j,n} \le \ln\frac{U}{L}\right\}} = 1 \\ 0, & 1_{\left\{0<\sum_{j=1}^{1}\xi_{j,n} \le \ln\frac{U}{L}\right\}} \cdots 1_{\left\{0<\sum_{j=1}^{k-1}\xi_{j,n} \le \ln\frac{U}{L}\right\}} = 0 \end{cases}$$
$$= \begin{cases} \left(\zeta_{k-1,n} + \xi_{k,n}\right) 1_{\left\{0<\sum_{j=1}^{k}\xi_{j,n} \le \ln\frac{U}{L}\right\}}, & 1_{\left\{0<\sum_{j=1}^{1}\xi_{j,n} \le \ln\frac{U}{L}\right\}} \cdots 1_{\left\{0<\sum_{j=1}^{k-1}\xi_{j,n} \le \ln\frac{U}{L}\right\}} = 1 \\ 0, & 1_{\left\{0<\sum_{j=1}^{1}\xi_{j,n} \le \ln\frac{U}{L}\right\}} \cdots 1_{\left\{0<\sum_{j=1}^{k-1}\xi_{j,n} \le \ln\frac{U}{L}\right\}} = 0 \end{cases}. \tag{18}$$

So we know $0 \le \zeta_{k,n} \le \ln\frac{U}{L} = nd$. As the possible value of $\zeta_{k,n}$ is an integer multiple of *d*, the value of $\zeta_{k,n}$ may be $0, d, 2d, \cdots, nd$. For $i = 0, 1, 2, \cdots, n$, let $q_{i,k,n} = P\left(\zeta_{k,n} = id\right)$. Notice that $\zeta_{k,n} > 0 \Leftrightarrow 1_{\left\{0<\sum_{j=1}^{1}\xi_{j,n} \le \ln\frac{U}{L}\right\}} \cdots 1_{\left\{0<\sum_{j=1}^{k}\xi_{j,n} \le \ln\frac{U}{L}\right\}} = 1$, from Equation (18),

$$\begin{aligned} q_{i,k,n} &= P\left(\zeta_{k,n} = id\right) \\ &= P\left(\left(\zeta_{k,n-1} + \xi_{k,n}\right) 1_{\left\{0<\sum_{j=1}^{k}\xi_{j,n} \le \ln\frac{U}{L}\right\}} 1_{\left\{\zeta_{k,n-1}\right\}} = id\right) \\ &= \sum_{j=1}^{n} P\left(\zeta_{k,n-1} = jd\right) P\left(\xi_{k,n} = (i-j)d\right) \\ &= \sum_{j=1}^{n} q_{j,k-1,n}\, p_{i-j,n}, \end{aligned} \tag{19}$$

$$q_{0,k,n} = 1 - \sum_{i=1}^{n} q_{i,k,n}. \tag{20}$$

Using Equation (19) and Equation (20) we can calculate all $q_{i,k,n}$ step by step. Notice that $-(n-1) \le i - j \le n-1$, we just need to precompute $p_{k,n}$ for $k = -n+1, -n+2, \cdots, n-1$. Plug it into Equation (13), we have

$$\begin{aligned} \tilde{V}_n(S,T) &= \mathrm{e}^{-rT} E\left(\max\left(L\mathrm{e}^{\zeta_{m,n}} - K, 0\right)\right) \\ &= \mathrm{e}^{-rT} \sum_{i=1}^{n} q_{i,m,n} \max\left(L\mathrm{e}^{id} - K, 0\right). \end{aligned} \tag{21}$$

It means we could calculate $\tilde{V}_n(S,T)$ by Equation (19), Equation (20) and Equation (21) after we got $p_{k,n}$ and $\tilde{p}_{k,n}$. Actually, as we don't need $q_{0,k,n}$ to calculate $\tilde{V}_n(S,T)$ by Equation (21), there is no need to calculate Equation (20). Equation (19) is a discrete convolution we have to calculate *m* times.

We use the p.d.f. of ξ_i To get $p_{k,n}$ and $\tilde{p}_{k,n}$. Denote

$$g_j(x) = \frac{(\lambda T/m)^j}{j!} \mathrm{e}^{-\frac{\lambda T}{m}} \phi\left(\frac{\gamma T}{m} + j\mu_J, \frac{\sigma^2 T}{m} + j\sigma_J^2\right)(x).$$

Because $\sum_{j=0}^{\infty} g_j(x)$ is uniform convergence, for $i = 2, 3, \cdots, m$, by the definition of $p_{k,n}$, we have

$$\begin{aligned}
p_{k,n} &= P\left(\left(k-0.5\right)d \le \xi_2 < \left(k+0.5\right)d\right) = \int_{(k-0.5)d}^{(k+0.5)d} h\left(x\right)\mathrm{d}x \\
&= \int_{(k-0.5)d}^{(k+0.5)d} \sum\nolimits_{j=0}^{\infty} g_j\left(x\right)\mathrm{d}x = \sum\nolimits_{j=0}^{\infty}\int_{(k-0.5)d}^{(k+0.5)d} g_j\left(x\right)\mathrm{d}x \\
&= \sum\nolimits_{j=0}^{\infty}\frac{\left(\lambda T/m\right)^j}{j!}\mathrm{e}^{-\frac{\lambda T}{m}}\left(\Phi\left(\frac{\gamma T}{m}+j\mu_J,\frac{\sigma^2 T}{m}+j\sigma_J^2\right)\left(\left(k+0.5\right)d\right)\right. \\
&\quad \left.-\Phi\left(\frac{\gamma T}{m}+j\mu_J,\frac{\sigma^2 T}{m}+j\sigma_J^2\right)\left(\left(k-0.5\right)d\right)\right) \\
&= \sum\nolimits_{j=0}^{\infty}\frac{\left(\lambda T/m\right)^j}{j!}\mathrm{e}^{-\frac{\lambda T}{m}}\left(\Phi\left(0,1\right)\left(\frac{\left(k+0.5\right)d-\gamma T/m-j\mu_J}{\sqrt{\sigma^2 T/m+j\sigma_J^2}}\right)\right. \\
&\quad \left.-\Phi\left(0,1\right)\left(\frac{\left(k-0.5\right)d-\gamma T/m-j\mu_J}{\sqrt{\sigma^2 T/m+j\sigma_J^2}}\right)\right),
\end{aligned} \tag{22}$$

where $\Phi\left(\mu,\sigma^2\right)$ is c.d.f. of $N\left(\mu,\sigma^2\right)$, and $\Phi\left(0,1\right)$ is c.d.f. of standard normal distribution. Similarly, from the p.d.f. of ξ_1, we can get

$$\begin{aligned}
\tilde{p}_{k,n} &= \sum\nolimits_{j=0}^{\infty}\frac{\left(\lambda T/m\right)^j}{j!}\mathrm{e}^{-\frac{\lambda T}{m}}\left(\Phi\left(0,1\right)\left(\frac{\left(k+0.5\right)d-\ln\left(L/S_0\right)-\gamma T/m-j\mu_J}{\sqrt{\sigma^2 T/m+j\sigma_J^2}}\right)\right. \\
&\quad \left.-\Phi\left(0,1\right)\left(\frac{\left(k-0.5\right)d-\ln\left(L/S_0\right)-\gamma T/m-j\mu_J}{\sqrt{\sigma^2 T/m+j\sigma_J^2}}\right)\right).
\end{aligned} \tag{23}$$

Unfortunately, we can't calculate $p_{k,n}$ and $\tilde{p}_{k,n}$ directly although we have got their analytic expression. Let

$$\begin{aligned}
p_{k,n,n'} &= \sum\nolimits_{j=0}^{n'}\frac{\left(\lambda T/m\right)^j}{j!}\mathrm{e}^{-\frac{\lambda T}{m}}\left(\Phi\left(0,1\right)\left(\frac{\left(k+0.5\right)d-\gamma T/m-j\mu_J}{\sqrt{\sigma^2 T/m+j\sigma_J^2}}\right)\right. \\
&\quad \left.-\Phi\left(0,1\right)\left(\frac{\left(k-0.5\right)d-\gamma T/m-j\mu_J}{\sqrt{\sigma^2 T/m+j\sigma_J^2}}\right)\right),
\end{aligned} \tag{24}$$

$$\begin{aligned}
\tilde{p}_{k,n,n'} &= \sum\nolimits_{j=0}^{n'}\frac{\left(\lambda T/m\right)^j}{j!}\mathrm{e}^{-\frac{\lambda T}{m}}\left(\Phi\left(0,1\right)\left(\frac{\left(k+0.5\right)d-\ln\left(L/S_0\right)-\gamma T/m-j\mu_J}{\sqrt{\sigma^2 T/m+j\sigma_J^2}}\right)\right. \\
&\quad \left.-\Phi\left(0,1\right)\left(\frac{\left(k-0.5\right)d-\ln\left(L/S_0\right)-\gamma T/m-j\mu_J}{\sqrt{\sigma^2 T/m+j\sigma_J^2}}\right)\right),
\end{aligned} \tag{25}$$

from above, we know $\lim\limits_{n'\to\infty} p_{k,n,n'} = p_{k,n}, \lim\limits_{n'\to\infty}\tilde{p}_{k,n,n'} = \tilde{p}_{k,n}$, and $p_{k,n,n'},\tilde{p}_{k,n,n'}$ can be directly calculate. By using $p_{k,n,n'},\tilde{p}_{k,n,n'}$ to approximate $p_{k,n},\tilde{p}_{k,n}$, we finally get an approximation of $\tilde{V}_n\left(S,T\right)$, and denote it with $\tilde{V}_{n,n'}\left(S,T\right)$, who is a computable approximation of $V\left(S,T\right)$. If parameter is different in different monitor interval, we should just refresh $g_j\left(x\right)$ in each iteration to keep the algorithm run correctly.

Secondly, we introduce our Monte Carlo simulation method. The main idea Monte Carlo simulation method is generating thousands of simulations of S_t. Then we can use the mean price of option base on these S_t to estimate $V\left(S,T\right)$

according to the law of large numbers. While the price of option just bases on the value of S_t on monitor date, we just need to simulation these values on monitor date. From Equation (2) we can conveniently simulate $\ln S_t$ then get S_t indirectly. In other words, we simulate $\xi_i, i = 1, 2, \cdots, m$. As we know ξ_i is sum of independent random variables including a normal random variable $\eta \sim N\left(\frac{\gamma T}{m}, \frac{\sigma^2 T}{m}\right)$ and a series of normal random variables $\eta_k \sim N\left(\mu_J, \sigma_J^2\right)$. The length of the series is follow a Poisson distribution with intensity $\lambda T/m$. We simulate η, and use a while loop to simulate η_k, and exit the loop with probability $e^{-\lambda T/m}$ each round. As usual, estimation error of Moto Carlo simulation is Proportional to $1/\sqrt{N}$ while estimated standard deviation is $\sigma_V/\sqrt{N}$, σ_V is standard deviation of $V(S,T)$, N is the number of simulation path. Because we can know the option value if S_t touch the barrier in early monitor date, so in simulation, we can also stop a simulation round in advance. However, the probability to stop is different for different parameters. So we use no-stop method to simulate and record the time in worst situation.

4. Numerical Results

We use some cases to compare the estimation accuracy of two methods introduced above.

Firstly, we use parameters that Poisson intensity $\lambda = 0$ in case 1, case 2 and case 3 to compare the result with other literatures. It shows when λ tends to 0, results tends to the results base on B-S model. Using the estimate value and estimate standard error calculate by Monte Carlo Simulation as the real value, the t statistic of estimate error of numerical algorithm with n = 1000 is around 1. It means its estimate error is at the same level with the estimate standard error of Monte Carlo Simulation, *i.e.*, the accuracy of numerical algorithm with n = 1000 is like the accuracy of Monte Carlo Simulation with simulation times = 100000. However, CPU time cost by the latter is thousands times of the former. To estimate the numerical error on the option price, we will compare the numerical result with the accurate result with n = 10000 and calculate the error.

Case 1: Prices of discrete double knock-out call option in 5 monitoring dates. The current price of the underlying asset is S_0, the strike price is 100, the volatility is 20% per annum, the call option has six months remaining to maturity, the risk-free rate is 10% per annum (compounded continuously), the lower barrier is placed at 95, and the upper barrier is imposed at 110. Numerical method estimate numerical error (written in brackets). Monte Carlo Simulation also estimate standard error (written in brackets), Milev's result is copy from his paper. Because they are generated by different computers, so CPU time in present methods and CPU time in Milev's paper cannot be compared. Table 1 shows the result.

Case 2: Prices of discrete double knock-out call option monitored daily (125 times) for different values of the underlying asset S_0 and parameters $K = 100$, $\sigma = 0.25$, $T = 0.5$, $r = 0.05$, $L = 95$, $U = 110$, see Table 2.

Table 1. 5 monitoring dates, S_0, $K = 100$, $\sigma = 0.25$, $T = 0.5$, $r = 0.05$, $L = 95$, $U = 110$.

S_0	Numerical algorithm $n = 200$	Numerical algorithm $n = 1000$	Monte Carlo simulation Times = 10^6	Milev's Num. algorithm $n = 1000$	Milev's M.C. simulation times = 10^7
95	0.172487 (0.00197)	0.174095 (0.00036)	-	0.174498	-
95.0001	0.172489 (0.00197)	0.174096 (0.00036)	0.174872 (0.00103)	0.174499	0.17486 (0.00064)
100	0.229986 (0.00247)	0.232003 (0.00045)	0.231914 (0.00118)	0.232508	0.23263 (0.00036)
109.9999	0.165450 (0.00191)	0.167005 (0.00035)	0.167474 (0.00101)	0.167394	0.16732 (0.00062)
110	0.165449 (0.00191)	0.167004 (0.00035)	-	0.167393	-
CPU	27 ms	78 ms	418 s	39 s	hundred sec.

Table 2. 125 monitoring dates, S_0, $K = 100$, $\sigma = 0.25$, $T = 0.5$, $r = 0.05$, $L = 95$, $U = 110$.

S_0	Numerical algorithm $n = 200$	Numerical algorithm $n = 1000$	Monte Carlo simulation Times = 10^6	Milev's Num. algorithm $n = 1000$	Milev's M.C. simulation times = 10^7
95	0.002536 (1.619e−4)	0.002668 (3.020e−5)	-	0.0027003	-
95.0001	0.002536 (1.619e−4)	0.002668 (3.020e−5)	0.002737 (0.00012)	0.0027005	0.002673 (0.00007)
100	0.010918 (4.915e−4)	0.011319 (9.055e−5)	0.011167 (0.00025)	0.011414	0.011394 (0.00015)
109.9999	0.002427 (1.550e−4)	0.002553 (2.890e−5)	0.002697 (0.00012)	0.00258415	0.002664 (0.00007)
110	0.002427 (1.550e−4)	0.002553 (2.890e−5)	-	0.0025843	-
CPU	408 ms	2025 ms	7836 s	39 s	hundreds sec.

Case 3: Prices of discrete double knock-out call option monitored weekly (25 times) for different values of the underlying asset S_0 and parameters $K = 100$, $\sigma = 0.25$, $T = 0.5$, $r = 0.05$, $L = 95$, $U = 110$, see Table 3.

Secondly, we compare the efficiency of numerical method and simulation method in case 4 and case 5. Poisson intensity $\lambda > 0$, *i.e.*, jump is considered. We set $\lambda = 5$, $\mu_J = 0.05, \sigma_J^2 = 0.05$, other parameters except *U*, *L* and S_0 are the same with case 2. We could learn that while *U* and *L* are quite close, option is Worthless. So standard deviation of option value at maturity is so large relative to the option value at now. Both of the two methods have a large relative error. While *U* and *L* are not quite close, although calculate result's accuracy is a bit worse than Simulation result's, it is still much better in efficiency as it cost much less CPU time.

Case 4: *U* and *L* are close so that the option has a high probability to knock-out, see Table 4.

Table 3. 25 monitoring dates, S_0, $K = 100$, $\sigma = 0.25$, $T = 0.5$, $r = 0.05$, $L = 95$, $U = 110$.

S_0	Numerical algorithm $n = 200$	Numerical algorithm $n = 1000$	Monte Carlo simulation Times = 10^6	Milev's Num. algorithm $n = 1000$	Milev's M.C. simulation times = 10^7
95	0.018879 (6.359e−4)	0.019397 (1.178e−4)	-	0.019528	-
95.0001	0.018879 (6.359e−4)	0.019397 (1.178e−4)	0.018508 (0.00033)	0.019528	0.019515 (0.00021)
100	0.041751 (1.182e−3)	0.042716 (2.184e−4)	0.043521 (0.00051)	0.042957	0.042736 (0.00031)
109.9999	0.018067 (6.086e−4)	0.018563 (1.128e−4)	0.018853 (0.00033)	0.018688	0.018676 (0.00019)
110	0.018067 (6.086e−4)	0.018563 (1.128e−4)	-	0.018688	-
CPU	87 ms	407 ms	1643 s	9 s	hundred sec.

Table 4. $\lambda = 5$, $\mu_J = 0.05$, $\sigma_J^2 = 0.05$, $L = 95$, $U = 105$.

S_0	Numerical algorithm $n = 200$	Numerical algorithm $n = 1000$	Numerical algorithm $n = 2000$	Monte Carlo Simulation Times = 10^6
95.1	1.08754e−6 (1.131e−7)	1.17917e−6 (2.149e−8)	1.19107e−6 (9.588e−9)	0(−)
96	1.77950e−6 (1.769e−7)	1.922880e−6 (3.353e−8)	1.94145e−6 (1.496e−8)	3.33815e−6 (2.41542e−6)
100	3.93182e−6 (3.626e−7)	4.22590e−6 (6.850e−8)	4.22590e−6 (6.850e−8)	9.290592e−7 (8.43256e−7)
104	2.18637e−6 (2.171e−7)	2.36232e−6 (4.114e−8)	2.38510e−6 (1.835e−8)	0(−)
104.9	1.44651e−6 (1.499e−7)	1.56798e−6 (2.847e−8)	1.58375e−6 (1.269e−8)	4.638891e−6 (3.373127e−6)
CPU	397 ms	2072 ms	7114 ms	7984 s

Case 5: U and L are not quite close so that the option has a lower probability to knock-out, see Table 5.

Finally, if we predict parameters are not constants, we could set different value of parameters in different monitor interval (interval between two adjacent monitor date). In case 6, parameters are the same with case 5, except σ and r. σ will increase from 0.1 to 0.25 in linearly step in monitor intervals. r will decrease from 0.05 to 0.02 in linearly step in monitor intervals. In this case, as we should refresh $g_j(x)$ in each iteration, while $n = 2000$, CPU time mainly cost by generate $p_{k,n,n'}$ whose time cost is $O(mnn')$. From the result we could learn that while $n = 2000$, the accuracy of numerical algorithm is a bit worse than Monte Carlo simulation. But while it just cost 1.4% CPU time of simulation method, as CPU time cost is proportional to n which means calculate result's convergence rate is faster than simulation while $n = 2000$. It still has a much better efficiency.

Case 6: see Table 6.

Table 5. $\lambda = 5$, $\mu_J = 0.05$, $\sigma_J^2 = 0.05$, $L = 80$, $U = 140$.

S_0	Numerical algorithm $n = 200$	Numerical algorithm $n = 1000$	Numerical algorithm $n = 2000$	Monte Carlo Simulation Times = 10^6
80.1	0.255603 (0.01708)	0.269405 (0.00328)	0.271218 (0.00146)	0.275555 (0.00224)
81	0.461714 (0.01941)	0.477431 (0.00369)	0.479476 (0.00165)	0.484087 (0.00296)
100	5.042788 (0.01736)	5.028913 (0.00349)	5.026992 (0.00156)	5.020779 (0.00857)
139	0.951235 (0.06628)	1.005237 (0.01228)	1.012056 (0.00546)	1.022783 (0.00458)
139.9	0.660173 (0.05794)	0.707246 (0.01087)	0.713274 (0.00484)	0.716428 (0.00386)
CPU	376 ms	1891 ms	6496 ms	7378 s

Table 6. σ, r are not constants.

S_0	Numerical algorithm $n = 200$	Numerical algorithm $n = 1000$	Numerical algorithm $n = 2000$	Monte Carlo Simulation Times = 10^6
80.1	0.109371 (0.01748)	0.123303 (0.00355)	0.125255 (0.00160)	0.124955 (0.00146)
81	0.329240 (0.01826)	0.343901 (0.00360)	0.345886 (0.00161)	0.346085 (0.00240)
100	5.349461 (0.04085)	5.315899 (0.00730)	5.311831 (0.00323)	5.296170 (0.00847)
139	1.613736 (0.17757)	1.758731 (0.03257)	1.776845 (0.01447)	1.789104 (0.00615)
139.9	0.777533 (0.16118)	0.907367 (0.03135)	0.924693 (0.01402)	0.948932 (0.00457)
CPU	9942 ms	49 s	102 s	7414 s

5. Conclusion

While mainstream methods to price European discrete knock-out double barrier option are not better than Monte Carlo simulation method in estimated error order, the advantage of the presented numerical algorithm is that it costs less compute time than simulation method. It is benefit from its direct calculate thought. Because of the main idea of this method that is not complicated, it can be extended to the case stock price obeys Merton jump diffusion model and work well on it. However, as this numerical method is based on the analytical solution, it is applicable only to those stochastic models for which the transition probability can be computed in closed-form.

References

[1] Black, F. and Scholes, M. (1973) The Pricing of Options and Corporate Liabilities.

Journal of Political Economy, **81**, 637-659. https://doi.org/10.1086/260062

[2] Merton, R.C. (1973) Theory of Rational Option Pricing. *The Bell Journal of Economics*, **4**, 141-183. https://doi.org/10.2307/3003143

[3] Reimer, M. and Sandmann, K. (1995) A Discrete Time Approach for European and American Barrier Options. *Ssrn Electronic Journal*. https://doi.org/10.2139/ssrn.6075

[4] Gao, B., Huang, J. and Subrahmanyam, M. (2000) The Valuation of American Barrier Options Using the Decomposition Technique. *Journal of Economic Dynamics & Control*, **24**, 1783-1827.

[5] Dai, M. and Yue, K.K. (2004) Knock-in American Options. *Journal of Futures Markets*, **24**, 179-192. https://doi.org/10.1002/fut.10101

[6] Wang, L. and Du, X. (2008) Pricing of European Barrier Option on Jump-Diffusion Model. *Mathematics in Economics*, **25**, 248-253.

[7] Xie, C. (2001) Pricing Barrier Options under CEV Process. *Journal of Management Sciences in China*, **4**, 13-20.

[8] Farnoosh, R., Sobhani, A., Rezazadeh, H. and Beheshti, M. (2016) Numerical Method for Discrete Double Barrier Option Pricing with Time-Dependent Parameters. *Computational Economics*, **70**, 2006-2013.

[9] Golbabai, A., Ballestra, L.V. and Ahmadian, D. (2014) A Highly Accurate Finite Element Method to Price Discrete Double Barrier Options. *Computational Economics*, **44**, 153-173. https://doi.org/10.1007/s10614-013-9388-5

[10] Ahmadian, D. and Ballestra, L.V. (2015) A Numerical Method to Price Discrete Double Barrier Options under a Constant Elasticity of Variance Model with Jump Diffusion. *International Journal of Computer Mathematics*, **92**, 2310-2328. https://doi.org/10.1080/00207160.2014.986114

[11] Milev, M. and Tagliani, A. (2009) Numerical Valuation of Discrete Double Barrier Options. *Journal of Computational and Applied Mathematics*, **233**, 2468-2480.