J. Korean Soc. Math. Educ. Ser. B: Pure Appl. Math. ISSN(Print) 1226-0657
https://doi.org/10.7468/jksmeb.2020.27.4.231 ISSN(Online) 2287-6081
Volume 27, Number 4 (November 2020), Pages 231–249

AN EFFICIENT AND ROBUST NUMERICAL METHOD FOR OPTION PRICES IN A TWO-ASSET JUMP-DIFFUSION MODEL

Chaeyoung Lee [a], Jian Wang [a], Hanbyeol Jang [b], Hyunsoo Han [b], Seongjin Lee [b], Wonjin Lee [b], Kisung Yang [c] and Junseok Kim [a,*]

Abstract. We present an efficient and robust finite difference method for a two-asset jump diffusion model, which is a partial integro-differential equation (PIDE). To speed up a computational time, we compute a matrix so that we can calculate the non-local integral term fast by a simple matrix-vector operation. In addition, we use bilinear interpolation to solve integral term of PIDE. We can obtain more stable value by using the payoff-consistent extrapolation. We provide numerical experiments to demonstrate a performance of the proposed numerical method. The numerical results show the robustness and accuracy of the proposed method.

1. Introduction

It is generally assumed that the return on asset prices follows a normal distribution. However, it can be found in the real financial market that the assumption is not correct. The sharp decline and fear of sudden movements in financial assets occur more frequently than if the return on financial assets is assumed to be a normal distribution. This means that the distribution of the return on asset prices has a fat tail. This situation results in a discontinuity in asset prices. In other words, the prices of assets have jumped. This is important for the theory and practice of derivatives because it has a big impact when valuing financial products. This is why a jump-term is needed to valuation a financial instrument to more suit for the real financial market.

A number of models that are more generic and consistent with the empirical behavior of market prices have been proposed. To cover normally distributed jumps in

Received by the editors February 21, 2020. Accepted September 01, 2020.
2010 *Mathematics Subject Classification.* 65M06, 91G60, 91G80.
Key words and phrases. jump-diffusion, Simpson's rule, non-uniform grid, implicit finite difference method, derivative securities.

[*] Corresponding author.

asset returns, Merton proposed Merton's jump-diffusion model [17] which extends the Black–Scholes model (BSM) at first time. The jump process is used in many areas of modeling [21]. Also, it derives a pricing formula for a single-asset European option on a stock price with jumps. There is another well-known generalization of BSM. Kou's double exponential jump-diffusion model [15] imposed a higher probability for extreme jumps and allowed asymmetry in the jump distribution. Kou [15] also developed analytical solutions to several single-asset option pricing problems including European call and put options. These alternative models are reported to be superior to the classical BSM in describing empirical observations and their modeling flexibility [19]. The presence of jumps is also important in multi-asset option pricing. In general, the price of a multi-asset option is highly sensitive to the dependence structure among underlying assets [20]. Therefore, modeling jump is an integral part of multi-asset option pricing. Many authors are interested in the jump-diffusion model, solve the non-local integral term, and try to calculate the option that follows the jump-diffusion model using the numerical methodology. Tankov and Volchkova [20] introduced the jump process by using Fourier transform and Poisson process. The authors in [13] calculated option price of one asset jump diffusion model by the numerical method. The authors in [5] proposed that computation of implicit finite difference method included solution of dense linear systems. Authors used fixed point iteration method and fast Fourier transform to solve dense linear system. In [8], the authors proposed an explicit finite difference method with two-dimensional Gauss–Hermite quadrature that solved integral term of two-asset jump-diffusion partial integro-differential equation (PIDE). In [2], the authors assessed the basket options where the dynamics of basket assets is described as a constant-elasticity-of-variance (CEV) jump diffusion system. In [14], the authors applied the PIDE for considering the two-dimensional structural default model with jumps to compute credit default swaps and first-to-default swaps. The authors studied its stability and consistency by using the von Neumann stability analysis. In [9], the authors applied radial basis functions partition of unity (RBF-PU) method and three time level Crank–Nicolson Leapfrog discretization to compute European option. Radial basis functions is used to solve multi-dimensional parabolic partial differential equation (PDE) but it has some difficulties. The authors applied RBF-PU method to overcome adverse condition in global mesh-free methods. In the jump diffusion model, option pricing is complex because it has an integral term unlike a general Black–Scholes PDE.

In this paper, we are going to price the option of having two underlying assets that follow the jump-diffusion process, such as call on max and put on min options. Consider the payoff function as $u(x, y, 0)$. The call on max option whose payoff is given as $u(x, y, 0) = \max(\max(x, y) - K,\ 0)$ with the strike price K, the underlying asset prices x, y. Likewise, put on min option whose payoff is given as $u(x, y, 0) = \max(K - \min(x, y),\ 0)$. We are going to use the bilinear interpolation and the simple matrix-vector operation to solve non-local integral term. When we calculate the price of an option with two underlying assets, we divide it into two areas and use the bilinear interpolation to address the overestimation of the value of the option in the part where the option price's graph bends. To obtain more stable value, we use the payoff-consistent extrapolation.

This paper is organized as follows. In Section 2, we describe the two-asset jump-diffusion equation based on the BSM and the proposed numerical scheme in detail. In Section 3, we present numerical experiments to verify accuracy and efficiency of our proposed algorithm. In Section 4, conclusions are drawn.

2. Numerical Solution

2.1. Derivation of the Governing equation Black and Scholes found that the second order PDE for the price of an option $u(x, t)$ on the stock that is called the BSM [1]. A critical assumption in the BSM is that the dynamics a stock price is governed by a single Brownian motion [11], thereby yielding a continuous stochastic process for the stock price movement. However, BSM cannot describe discontinuous feature such as jump [16] which asset prices have. Rebonato described the use of the Black–Scholes formula by financial practitioners [18]. There are two typical examples of sudden drops in asset prices during financial crisis. First, it is noted for Black Monday, 19 October 1987. On that day, the Dow Jones Industrial average depreciated almost 25 percent from 2,245 to 1,738 [3]. Second, the shock of 2008 was unexampled in that capital devaluation and loss in fortune were felt widely in main industry. With this effect, U.S. stock prices fell sharply 37 percent in 2008 [10]. From now on, we shall derive the governing equation based on the original BSM with the price of the underlying assets x, y and time t. The value of an option $u(x, y, t)$ is governed by the following two-asset BSM for $(x, y, t) \in \mathbf{R}_+^2 \times [0, T)$:

$$\frac{\partial u}{\partial t} + \frac{\sigma_1^2 x^2}{2}\frac{\partial^2 u}{\partial x^2} + \frac{\sigma_2^2 y^2}{2}\frac{\partial^2 u}{\partial y^2} + rx\frac{\partial u}{\partial x} + ry\frac{\partial u}{\partial y} + \rho\sigma_1\sigma_2 xy\frac{\partial^2 u}{\partial x \partial y} - ru = 0, \tag{2.1}$$

where T is maturity, σ_1 and σ_2 are the volatilities of x and y, respectively, r is the risk free interest rate, and ρ is the correlation between x and y. However, as we saw in the two previous examples, the distribution of stock price returns is more likely to result in extreme events in the stock market than when normally distributed. For this reason, Merton proposed a model for option pricing when underlying stock returns are discontinuous, which is simply called the Merton's jump-diffusion model [17]. The derivative securities pricing using the Merton's jump-diffusion model usually provides efficient and robust valuation. The two-asset jump-diffusion model is as follows:

$$\frac{dx}{x} = (r - \lambda k_1)dt + \sigma_1 dz_1 + (e^{\xi} - 1)dq, \tag{2.2}$$

$$\frac{dy}{y} = (r - \lambda k_2)dt + \sigma_2 dz_2 + (e^{\eta} - 1)dq, \tag{2.3}$$

$$dq = \begin{cases} 0 & \text{with probability } 1 - \lambda\ dt \\ 1 & \text{with probability } \lambda\ dt \end{cases},$$

where z_1 and z_2 are standard Brownian motions with correlation $E[dz_1 dz_2] = \rho dt$, λ is Poisson process of intensity, $e^{\xi} - 1$ and $e^{\eta} - 1$ are impulse functions, k_1 and k_2 represent the expected relative jump sizes, i.e., $k_1 = E[e^{\xi} - 1] = e^{\hat{\mu}_1 + \frac{1}{2}\hat{\sigma}_1^2} - 1$ and $k_2 = E[e^{\eta} - 1] = e^{\hat{\mu}_2 + \frac{1}{2}\hat{\sigma}_2^2} - 1$. Since x and y have the impulse functions which produce the jump, the asset price jumps from x to xe^{ξ} and from y to ye^{η} if a jump occurs. Then, we are able to get the PIDE given by

$$\begin{aligned} \frac{\partial u}{\partial t} &+ \frac{\sigma_1^2 x^2}{2}\frac{\partial^2 u}{\partial x^2} + \frac{\sigma_2^2 y^2}{2}\frac{\partial^2 u}{\partial y^2} + \rho\sigma_1\sigma_2 xy\frac{\partial^2 u}{\partial x \partial y} + (r - \lambda k_1)x\frac{\partial u}{\partial x} \\ &+ (r - \lambda k_2)y\frac{\partial u}{\partial y} - ru + \lambda\left(\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} u(xe^{\xi}, ye^{\eta})G(\xi,\eta)\,d\xi\,d\eta - u\right) = 0. \end{aligned} \tag{2.4}$$

Here, G is taken to be a distributed jump amplitude with bivariate normal probability density function (PDF) as follows:

$$G(\xi,\eta) = \frac{1}{2\pi\hat{\sigma}_1\hat{\sigma}_2\sqrt{1-\rho_{\xi\eta}^2}}\exp\left(-\frac{z}{2(1-\rho_{\xi\eta}^2)}\right),$$

where $z = (\frac{\xi - \hat{\mu}_1}{\hat{\sigma}_1})^2 - \frac{2\rho_{\xi\eta}(\xi-\hat{\mu}_1)(\eta-\hat{\mu}_2)}{\hat{\sigma}_1\hat{\sigma}_2} + (\frac{\eta-\hat{\mu}_2}{\hat{\sigma}_2})^2$, $\hat{\sigma}_1$ and $\hat{\sigma}_2$ are standard deviations of the jumps ξ and η, respectively. $\hat{\mu}_1$ and $\hat{\mu}_2$ are means of the jumps ξ and η, respectively. $\rho_{\xi\eta}$ is correlation of ξ and η [5, 8].

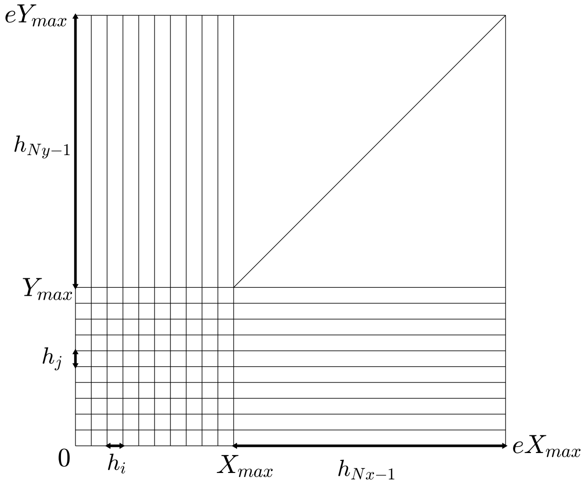


Figure 1. Schematic illustration of the computational domain.

2.2. Discretization with finite differences We solve Eq. (2.4) numerically using a finite difference method [7]. By the change of variable $\tau = T - t$ with maturity T, we can rewrite Eq. (2.4) as follows:

$$\begin{aligned}(2.5)\ \frac{\partial u}{\partial \tau} = \frac{\sigma_1^2 x^2}{2}\frac{\partial^2 u}{\partial x^2} + \frac{\sigma_2^2 y^2}{2}\frac{\partial^2 u}{\partial y^2} + \rho\sigma_1\sigma_2 xy\frac{\partial^2 u}{\partial x \partial y} + (r - \lambda k_1)x\frac{\partial u}{\partial x} + (r - \lambda k_2)y\frac{\partial u}{\partial y} \\ -ru + \lambda\left(\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} u(xe^{\xi}, ye^{\eta})G(\xi,\eta)\, d\xi\, d\eta - u\right).\end{aligned}$$

Let us discretize the jump-diffusion equation (2.5) on the computational domain. In the x direction, we use the grid defined by $x_0 = 0$ and $x_{i+1} = x_i + h_i$ for $i = 0, \cdots, N_x - 1$, where N_x is the number of grid intervals. Here, h_i is the grid spacing which is uniform from $i = 1$ to $i = N_x - 2$ and the last spatial step size is $h_{N_x-1} = (e-1)X_{max}$. We assume that $x_{N_x-1} = X_{max}$, $x_{N_x} = X_{max} + h_{N_x-1}$, and $x_{N_x} = eX_{max}$. In the y direction, the grid is defined in the same way. We denote the numerical approximation $u_{ij}^n \approx u(x_i, y_j, n\Delta\tau)$, where $\Delta\tau = T/N_\tau$ is the time step size, N_τ is the total number of time steps, and $n = 0,\ 1, ...,\ N_\tau$. Then we have the computational domain of price of the underlying assets x and y, see Figure 1.

Next, we consider the numerical quadrature of the following integral term:

$$(2.6)\qquad \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} u(xe^{\xi}, ye^{\eta})G(\xi,\eta)\mathrm{d}\xi\mathrm{d}\eta.$$

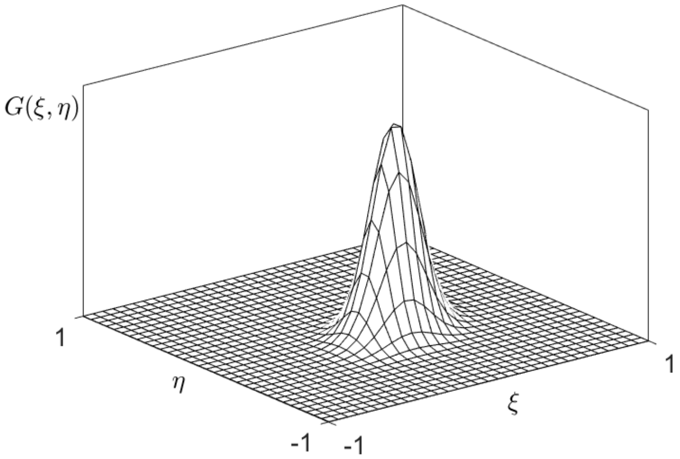


Figure 2. Bivariate normal PDF $G(\xi, \eta)$.

To evaluate the integral term (2.6), we reduce an infinite domain $(-\infty, \infty) \times (-\infty, \infty)$ to a finite domain $[-a, b] \times [-a, b]$. We calculate the option values at point $(S_1, S_2) = (100, 100)$ on the different domains used to calculate non-local integral term (2.6). Here, the value is the price of the European put on the min option and we use the same parameters as in Section 3. Table 1 lists the option values and the elapsed times. Even if the values of a and b are greater than 1, there is no difference from the option value when $a = b = 1$, but the elapsed time is much longer. To efficiently calculate the option value with the jump-diffusion model, we can use the finite domain $[-1, 1] \times [-1, 1]$.

Table 1. Effective domain $[-a, b] \times [-a, b]$.

a*=*b	**0.5**	**1**	**2**
value	8.8885	9.1146	9.1147
elapsed time	20.713918	66.014685	258.745284

Therefore, we use the non-local integral term (2.7) instead of the term (2.6).

$$\int_{-1}^{1}\int_{-1}^{1} u(xe^{\xi}, ye^{\eta})G(\xi,\eta)\mathrm{d}\xi\mathrm{d}\eta. \tag{2.7}$$

Figure 2 shows the G value on the domain $[-1, 1] \times [-1, 1]$.

For the option price u_{ij}^n, we define the discrete difference operators $\mathcal{L}_{OS}^x$ and $\mathcal{L}_{OS}^y$ by

$$\begin{aligned}
\mathcal{L}_{OS}^{x} u_{ij}^{n+\frac{1}{2}} =& \frac{(\sigma_1 x_i)^2}{2} \frac{u_{i-1,j}^{n+\frac{1}{2}} - 2u_{ij}^{n+\frac{1}{2}} + u_{i+1,j}^{n+\frac{1}{2}}}{h^2} + (r - \lambda k_1) x_i \frac{u_{i+1,j}^{n+\frac{1}{2}} - u_{i-1,j}^{n+\frac{1}{2}}}{2h} - \frac{r}{2} u_{ij}^{n+\frac{1}{2}} \\
&+ \frac{1}{2}\sigma_1\sigma_2\rho x_i y_j \frac{u_{i+1,j+1}^{n} - u_{i-1,j+1}^{n} - u_{i+1,j-1}^{n} + u_{i-1,j-1}^{n}}{4h^2}, \\
\mathcal{L}_{OS}^{y} u_{ij}^{n+1} =& \frac{(\sigma_2 y_j)^2}{2} \frac{u_{i,j-1}^{n+1} - 2u_{ij}^{n+1} + u_{i,j+1}^{n+1}}{h^2} + (r - \lambda k_2) y_j \frac{u_{i,j+1}^{n+1} - u_{i,j-1}^{n+1}}{2h} - \frac{r}{2} u_{ij}^{n+1} \\
&+ \frac{1}{2}\sigma_1\sigma_2\rho x_i y_j \frac{u_{i+1,j+1}^{n+\frac{1}{2}} - u_{i-1,j+1}^{n+\frac{1}{2}} - u_{i+1,j-1}^{n+\frac{1}{2}} + u_{i-1,j-1}^{n+\frac{1}{2}}}{4h^2} \\
&+ \lambda \left(I_{-1}^{1} I_{-1}^{1} [u\left(x_i e^{\xi}, y_j e^{\eta}\right) G\left(\xi, \eta\right)] \Delta\xi \Delta\eta - u_{ij}^{n+\frac{1}{2}} \right),
\end{aligned}$$

respectively. Then, using the operator splitting method [12] to solve Eq. (2.5), we can consist of the following two discrete equations:

$$\frac{u_{ij}^{n+\frac{1}{2}} - u_{ij}^{n}}{\Delta\tau} = \mathcal{L}_{OS}^{x} u_{ij}^{n+\frac{1}{2}}, \tag{2.8}$$

$$\frac{u_{ij}^{n+1} - u_{ij}^{n+\frac{1}{2}}}{\Delta\tau} = \mathcal{L}_{OS}^{y} u_{ij}^{n+1}. \tag{2.9}$$

According to Eqs. (2.8) and (2.9), the numerical solutions at time level $n + \frac{1}{2}$ and at time level $n + 1$ are sequentially updated.

2.3. How to deal with boundary condition and calculation The summation of Eqs. (2.8) and (2.9) follows that the implicit finite difference form in Eq. (2.5).

$$\frac{u_{ij}^{n+1} - u_{ij}^{n}}{\Delta\tau} = \mathcal{L}_{OS}^{x} u_{ij}^{n+\frac{1}{2}} + \mathcal{L}_{OS}^{y} u_{ij}^{n+1}. \tag{2.10}$$

The numerical solution algorithm using the operator splitting method is by proceed in two-step. First, we can rewrite Eq. (2.8) as

$$\alpha_i u_{i-1,j}^{n+\frac{1}{2}} + \beta_i u_{ij}^{n+\frac{1}{2}} + \gamma_i u_{i+1,j}^{n+\frac{1}{2}} = f_{ij}, \tag{2.11}$$

where

$$\begin{aligned}
\alpha_i &= -\frac{\sigma_1^2 x_i^2}{2h^2} + \frac{(r - \lambda k_1) x_i}{2h},\ \beta_i = \frac{1}{\Delta\tau} + \frac{\sigma_1^2 x_i^2}{h^2} + \frac{r}{2},\ \gamma_i = -\frac{\sigma_1^2 x_i^2}{2h^2} - \frac{(r - \lambda k_1) x_i}{2h}, \\
f_{ij} &= \frac{1}{2}\rho\sigma_1\sigma_2 x_i y_j \frac{u_{i+1,j+1}^{n} - u_{i+1,j-1}^{n} - u_{i-1,j+1}^{n} + u_{i-1,j-1}^{n}}{4h^2} + \frac{u_{ij}^{n}}{\Delta\tau}.
\end{aligned}$$

Here, the boundary values at $x = 0$ and $x = X_{max}$ are obtained by the zero Dirichlet boundary condition and the linear boundary condition, respectively. Then the system of discrete equation (2.11) can be rewritten the following tridiagonal form:

$$(2.12)\quad \begin{pmatrix} \beta_1 & \gamma_1 & 0 & \dots & 0 \\ \alpha_2 & \beta_2 & \gamma_2 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \alpha_{N_x-2} & \beta_{N_x-2} & \gamma_{N_x-2} \\ 0 & \dots & 0 & \alpha_{N_x-1}-\gamma_{N_x-1} & \beta_{N_x-1}+2\gamma_{N_x-1} \end{pmatrix} \begin{pmatrix} u_{1,j}^{n+\frac{1}{2}} \\ u_{2,j}^{n+\frac{1}{2}} \\ \vdots \\ u_{N_x-2,j}^{n+\frac{1}{2}} \\ u_{N_x-1,j}^{n+\frac{1}{2}} \end{pmatrix} = \begin{pmatrix} f_{1,j}^* \\ f_{2,j} \\ \vdots \\ f_{N_x-2,j} \\ f_{N_x-1,j}^* \end{pmatrix},$$

where $f_{1,j}^* = f_{1,j} - \alpha_1 u_{0,j}^n$ and $f_{N_x-1,j}^* = f_{N_x-1,j} - \gamma_{N_x-1}\ u_{N_x,j}^n$ for every fixed index j. Next, Eq. (2.9) is rewritten as follows:

$$(2.13)\quad \alpha_j u_{i,j-1}^{n+1} + \beta_j u_{ij}^{n+1} + \gamma_j u_{i,j+1}^{n+1} = g_{ij},$$

where

$$\alpha_j = -\frac{\sigma_2^2 y_j^2}{2h^2} + \frac{(r-\lambda k_2)y_j}{2h},\ \beta_j = \frac{1}{\Delta\tau} + \frac{\sigma_2^2 y_j^2}{h^2} + \frac{r}{2},\ \gamma_j = -\frac{\sigma_2^2 y_j^2}{2h^2} - \frac{(r-\lambda k_2)y_j}{2h},$$

$$g_{ij} = \frac{1}{2}\rho\sigma_1\sigma_2 x_i y_j \frac{u_{i+1,j+1}^{n+\frac{1}{2}} - u_{i+1,j-1}^{n+\frac{1}{2}} - u_{i-1,j+1}^{n+\frac{1}{2}} + u_{i-1,j-1}^{n+\frac{1}{2}}}{4h^2} + \frac{u_{ij}^{n+\frac{1}{2}}}{\Delta\tau}$$
$$+\lambda\left(I_{-1}^1 I_{-1}^1\left[u\left(x_i e^{\xi}, y_j e^{\eta}\right) G\left(\xi,\eta\right)\right]\Delta\xi\Delta\eta - u_{ij}^{n+\frac{1}{2}}\right).$$

Here, the boundary values at $y = 0$ and $y = Y_{max}$ are obtained by the zero Dirichlet boundary condition and the linear boundary condition, respectively. Similarly, the system of discrete equation (2.13) can be rewritten the following tridiagonal form:

$$(2.14)\quad \begin{pmatrix} \beta_1 & \gamma_1 & 0 & \dots & 0 \\ \alpha_2 & \beta_2 & \gamma_2 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \alpha_{N_y-2} & \beta_{N_y-2} & \gamma_{N_y-2} \\ 0 & \dots & 0 & \alpha_{N_y-1}-\gamma_{N_y-1} & \beta_{N_y-1}+2\gamma_{N_y-1} \end{pmatrix} \begin{pmatrix} u_{i,1}^{n+1} \\ u_{i,2}^{n+1} \\ \vdots \\ u_{i,N_y-2}^{n+1} \\ u_{i,N_y-1}^{n+1} \end{pmatrix} = \begin{pmatrix} g_{i,1}^* \\ g_{i,2} \\ \vdots \\ g_{i,N_y-2} \\ g_{i,N_y-1}^* \end{pmatrix},$$

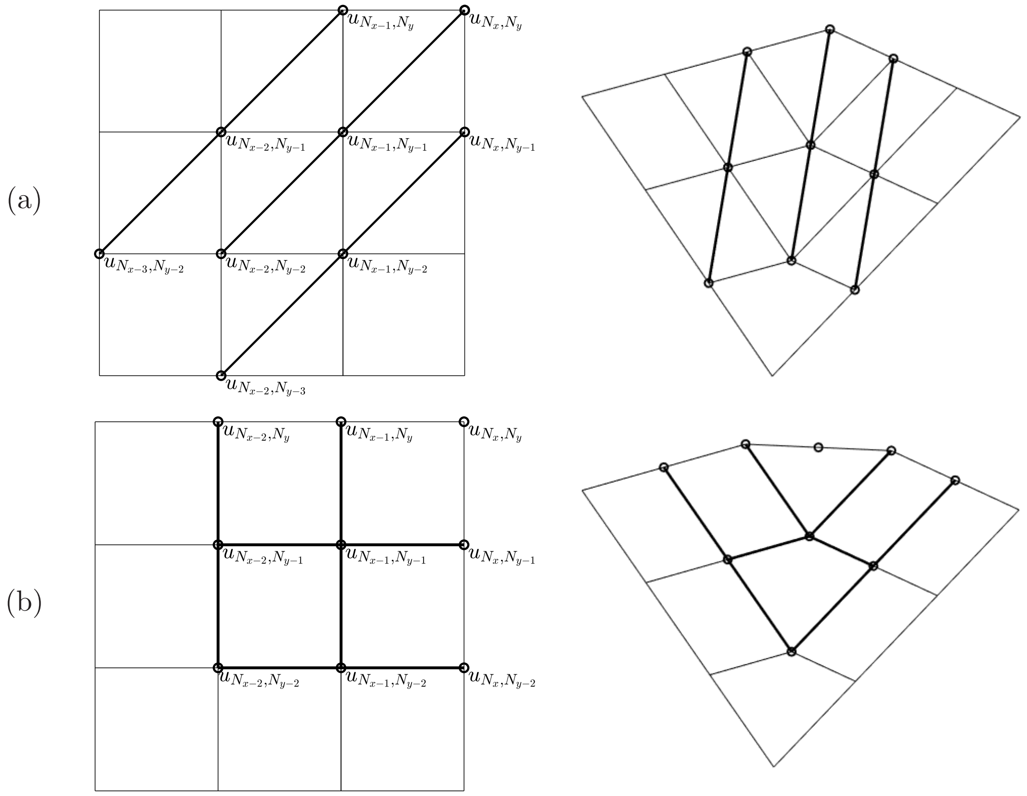


Figure 3. Schematic illustrations of (a) payoff-consistent and (b) conventional linear extrapolation to find boundary points.

where $g^*_{i,1} = g_{i,1} - \alpha_1 u_{i,0}^{n+\frac{1}{2}}$ and $g^*_{i,N_y-1} = g_{i,N_y-1} - \gamma_{N_y-1}\ u_{i,N_y}^{n+\frac{1}{2}}$ for every fixed index i. To solve the tridiagonal forms (2.13) and (2.15), we use the Thomas algorithm which can directly obtain the inverse of tridiagonal matrix [6]. In addition, we use the payoff-consistent extrapolation which is consistent with the value obtained from using the payoff function at the boundary points: $u_{N_x,N_y} = 2u_{N_x-1,N_y-1} - u_{N_x-2,N_y-2}$ [4], as shown in Figure 3(a). In the same way, the payoff-consistent extrapolation is applied at u_{N_x,N_y-1} and u_{N_x-1,N_y}. By using this method, the numerical solution with high correlation can be solved stably. In general, the linear boundary condition is mostly used in financial engineering: $u_{N_x,N_y} = 2u_{N_x-1,N_y} - u_{N_x-2,N_y}$. However, the linear extrapolation at the boundary points generates the discrepancy of the value obtained from using the payoff function at the boundary points. Figure 3(b) illustrates the schematic diagrams of conventional linear extrapolation to find boundary points.

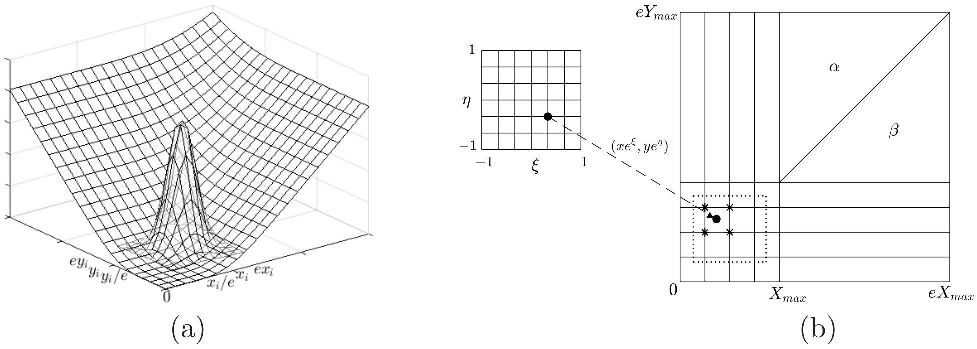


Figure 4. Schematic illustrations of (a) the option price $u(x, y)$ and $G(\xi, \eta)$ function, and (b) area of $u(x, y)$ at $x = x_i e^{\xi},\ y = y_j e^{\eta}$.

Therefore, we use the composite Simpson's rule for the quadrature. Given that the interval $[-1, 1]$ is divided by M grid points, the subintervals have equal width, $2/(M-1)$. Here, $M-1$ is a positive integer.

$$\begin{aligned}
& I_{-1}^{1} I_{-1}^{1} u(xe^{\xi}, ye^{\eta}) G(\xi, \eta)\, \Delta\xi \Delta\eta \;=\; I_{-1}^{1} I_{-1}^{1} g(\xi, \eta) \Delta\xi \Delta\eta \\
&= \frac{\Delta\xi\Delta\eta}{9} \Bigg[g(\xi_0, \eta_0) + g(\xi_0, \eta_M) + g(\xi_M, \eta_0) + g(\xi_M, \eta_M) \\
&\quad +4 \sum_{l=1}^{M/2} g(\xi_0, \eta_{2l-1}) + 2 \sum_{l=1}^{M/2-1} g(\xi_0, \eta_{2l}) + 4 \sum_{l=1}^{M/2} g(\xi_M, \eta_{2l-1}) + 2 \sum_{l=1}^{M/2-1} g(\xi_M, \eta_{2l}) \\
&\quad +4 \sum_{k=1}^{M/2} g(\xi_{2k-1}, \eta_0) + 2 \sum_{k=1}^{M/2-1} g(\xi_{2k}, \eta_0) + 4 \sum_{k=1}^{M/2} g(\xi_{2k-1}, \eta_M) + 2 \sum_{k=1}^{M/2-1} g(\xi_{2k}, \eta_M) \\
&\quad +16 \sum_{l=1}^{M/2} \left(\sum_{k=1}^{M/2} g(\xi_{2k-1}, \eta_{2l-1}) \right) + 8 \sum_{l=1}^{M/2-1} \left(\sum_{k=1}^{M/2} g(\xi_{2k-1}, \eta_{2l}) \right) \\
&\quad +8 \sum_{l=1}^{M/2} \left(\sum_{k=1}^{M/2-1} g(\xi_{2k}, \eta_{2l-1}) \right) + 4 \sum_{l=1}^{M/2-1} \left(\sum_{k=1}^{M/2-1} g(\xi_{2k}, \eta_{2l}) \right) \Bigg].
\end{aligned}$$

Figure 4(a) shows a schematic illustration of the option price $u(x, y)$ and $G(\xi, \eta)$ function.

In order to speed up the calculation, we store the factors of each point (x_i, y_j) used in bilinear interpolation as a matrices before the main time-step iterations. The factor matrix A and B are defined as $A = (a_{i,k})_{N_x \times (M-1)}$ and $B = (b_{j,l})_{N_y \times (M-1)}$, where $a_{i,k} = (x_i e^{\xi_k} - x_{i_k})/h_{i_k},\ x_{i_k - 1} \le x_i e^{\xi_k} < x_{i_k}$, for some $2 \le i_k \le N_x + 1$ and $b_{j,l} = (y_j e^{\eta_l} - y_{j_l})/h_{j_l},\ y_{j_l - 1} \le y_j e^{\eta_l} < y_{j_l}$, for some $2 \le j_l \le N_y + 1$. Then, we can use bilinear interpolation as

$$\begin{aligned} g(\xi_k, \eta_l) \;=\; & G(\xi_k, \eta_l)\Big(u(x_{i_k}, y_{j_l})(1-a_{i,k})(x_{i_k} - x_{i_k-1})(1-b_{j,l})(y_{j_l} - y_{j_l-1}) \\ & + u(x_{i_k-1}, y_{j_l-1})a_{i,k}(x_{i_k} - x_{i_k-1})b_{j,l}(y_{j_l} - y_{j_l-1}) \\ & + u(x_{i_k-1}, y_{j_l})a_{i,k}(x_{i_k} - x_{i_k-1})(1-b_{j,l})(y_{j_l} - y_{j_l-1}) \\ & + u(x_{i_k}, y_{j_l-1})(1-a_{i,k})(x_{i_k} - x_{i_k-1})b_{j,l}(y_{j_l} - y_{j_l-1})\Big) \\ & \Big/ (x_{i_k} - x_{i_k-1})(y_{j_l} - y_{j_l-1}). \end{aligned}$$

Figure 4(b) schematically illustrates the method of bilinear interpolation. If $x_i e^{\xi_k} > x_{N_x}$ and $y_j e^{\eta_l} > y_{N_y}$, we divide it into two regions α and β as shown in Figure 4(b) and execute interpolation using the plane equation with three points to resolve the overestimate of the option value.

3. Numerical Experiment

In this section, we perform numerical tests to show efficiency and robustness of the proposed numerical scheme under the jump-diffusion model. All computations were run in MATLAB R2018a. We consider a European put on min option whose payoff is given as $u(x, y, 0) = \max(K - \min(x, y),\ 0)$, where K is strike price. We select $K = 100$, risk-free interest rate $r = 0.05$, the diffusion parameters $\sigma_1 = 0.12$, $\sigma_2 = 0.15$, $\rho = 0.3$, and the jump parameters $\lambda = 0.6$, $\hat{\mu_1} = -0.1$, $\hat{\mu_2} = 0.1$, $\hat{\sigma}_1 = 0.17$, $\hat{\sigma}_2 = 0.13$, $\rho_{\xi\eta} = -0.2$. We discretize the operator in space by localization of the computational domain $\Omega = [0, L_x] \times [0, L_y]$, where $L_x = L_y = 300$. We use a reference value in [5] and compare our numerical value with the reference value using a percentage error.

3.1. Convergence test To find a suitable grid of interval $[-1, 1]$ for numerical integration, we test the effect of grid of interval $[-1, 1]$ with $M = 13, 15, 17, 19, 21, 23, 25$. We evaluate the numerical solution with $N_x = N_y = 121$ and $N_\tau = 360$. Table 2 shows the value of European put on min option price and percentage error between the numerical and reference values at point $(S_1, S_2) = (100, 100)$. Here, the reference value at the same point in [5] is 9.1178. We define the percentage error as

$$\text{percentage error} = \frac{|u_{ij}^{N_\tau} - \bar{u}|}{\bar{u}} \times 100. \tag{3.1}$$

The values of i and j will be taken to let $S_1 = 100$ and $S_2 = 100$, respectively, and $\bar{u}$ is the reference value. We set a tolerance of percentage error as 0.05. The result suggests that the value with $M = 17$ is accurate enough.

Table 2. $N_\tau = 360$, $N_x = N_y = 121$.

M	13	15	17	19	21	23	25
value	9.2109	9.0909	9.1146	9.1152	9.1121	9.1140	9.1144
percentage error	1.0211	0.2950	0.0351	0.02852	0.0625	0.0417	0.0373

Next, we find a suitable grid of time step. We evaluate the numerical solution with $N_x = N_y = 121$ and $M = 17$. Table 3 lists the results and shows that the value with $N_\tau = 360$ is accurate enough.

Table 3. $N_x = N_y = 121$, $M = 17$.

N_τ	30	90	180	360	720	1080	1440
value	9.0975	9.1100	9.1131	9.1146	9.1154	9.1157	9.1158
percentage error	0.2226	0.0855	0.0515	0.0351	0.0263	0.0230	0.0219

Table 4. $N_\tau = 360$, $M = 17$.

$N_x = N_y$	31	61	91	121	151	181	211
value	8.7415	9.0572	9.0991	9.1146	9.1219	9.1259	9.1284
percentage error	4.1271	0.6646	0.2051	0.0351	0.0450	0.0888	0.1163

Now, to find a suitable grid of space step, we take suitable parameters of grid of interval $[-1, 1]$ and time step size, $M = 17$ and $N_\tau = 360$. Table 4 shows that the error is decreasing and converges to the reference solution as we refine the grid. If $N_x = N_y$ are greater than 121, the error increases because of roundup error. As the result of this test, the value with $N_x = N_y = 121$ is accurate enough.

In Figure 5, we calculate the option value in a two-asset jump-diffusion model at point $(S_1, S_2) = (100, 100)$ for different grid of time step and space step, V is the option value.

Unless otherwise specifically stated, we use the parameters such as $M = 17$, $N_\tau = 360$, and $N_x = N_y = 121$ for the following test.

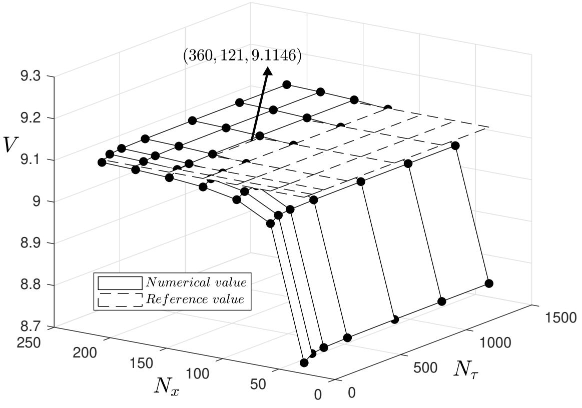


Figure 5. Result of the convergence test

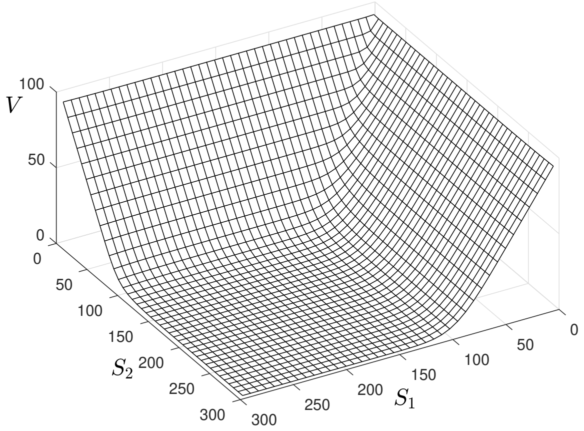
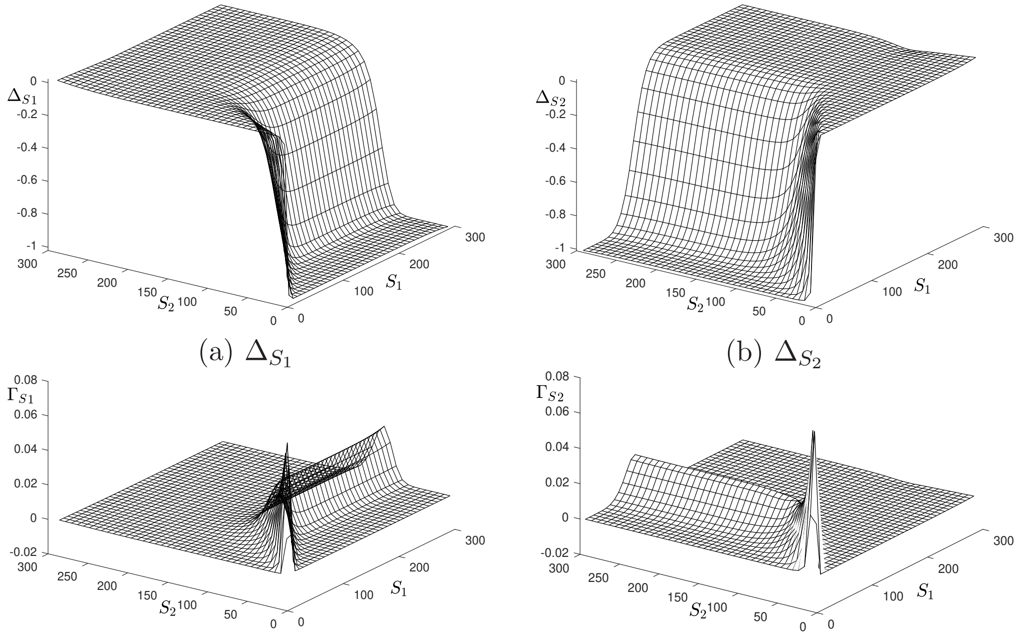


Figure 6. European put on min option value in a two-asset jump-diffusion model.

3.2. Numerical solution of European put on min option In this section, we demonstrate the performance of the proposed method for approximation of two-asset European put on min option pricing. The implementation of the proposed method and experiments has been divided into the two cases, option pricing and option Greeks. The two-asset initial values of S_1 and S_2 are defined by the domain $(0, L_x) \times (0, L_y)$. Figure 6 shows the option value in a two-asset jump-diffusion model.

We calculate the *Delta* ($\Delta = \partial V/\partial S$) and *Gamma* ($\Gamma = \partial^2 V/\partial S^2$) of the two-asset jump-diffusion model. Δ is the first derivative with respect to the underlying



(a) Δ_{S_1} (b) Δ_{S_2}

(c) Γ_{S_1} (d) Γ_{S_2}

Figure 7. Numerical solution of the derivatives of European put on min option in a two-asset jump-diffusion model. (a) and (b) are the first derivative with respect to S_1 and S_2, respectively; (c) and (d) are the second derivative with respect to S_1 and S_2, respectively.

asset S and Γ is the second derivative with respect to the underlying asset S. To compute Greeks, we apply the central finite difference method, i.e, $\Delta \approx [V(S+\Delta S) - V(S-\Delta S)]/(2\Delta S)$ and $\Gamma \approx [V(S-\Delta S) - 2V(S) + V(S+\Delta S)]/\Delta S^2$, where V is the option value, S is the underlying asset, and $\Delta S = L_x/(N_x - 1)$.

Figure 7 shows the numerical solution of the derivatives of European put on min option in a two-asset jump-diffusion model. Figure 7(a)–(d) represent Δ_{S_1}, Δ_{S_2}, Γ_{S_1}, and Γ_{S_2}, respectively. Here, Δ_{S_1} and Δ_{S_2} are the first derivatives with respect to the underlying assets of S_1 and S_2, respectively, and Γ_{S_1} and Γ_{S_2} are the second derivatives with respect to the underlying asset S_1 and S_2, respectively.

3.3. Comparison of interpolation methods In Section **2.3**, we explained the bilinear interpolation to solve the integral term of PIDE. In this test, we compare the option prices using different interpolation methods with the value obtained by the bilinear interpolation at point $(S_1, S_2) = (100, 100)$. Table 5 lists the option prices with three different interpolation methods and shows similar results. However, we can get a more stable option price by using the payoff-consistent extrapolation at boundary points and the bilinear interpolation method to solve the integral term of PIDE.

Table 5. Comparison of option prices with different interpolation methods.

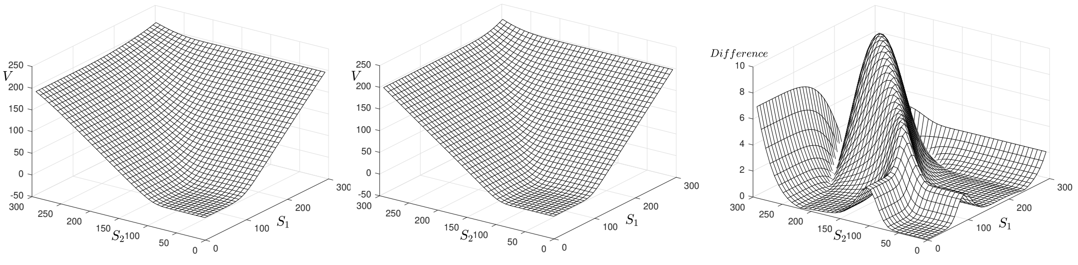
Method	**bilinear**	**nearest**	**spline**
value	9.1146	9.1425	9.1080

3.4. Comparison with benchmark paper In this section, we compare the computational times to price the European put on min option between our proposed method and benchmark paper's method. Fakharany et al. [8] proposed an explicit finite difference method with two-dimensional Gauss–Hermite quadrature for the integral term of two-asset jump-diffusion PIDE. We conduct the test with varying the values $N_x,\ N_y$, and N_τ. Other parameters are the same as in [8]. Table 6 shows that our method is faster than the reference method because we can use a relatively large time step size without the stability problem.

Table 6. Comparison of CPU time and the number of grid points (N_x, N_y, N_τ).

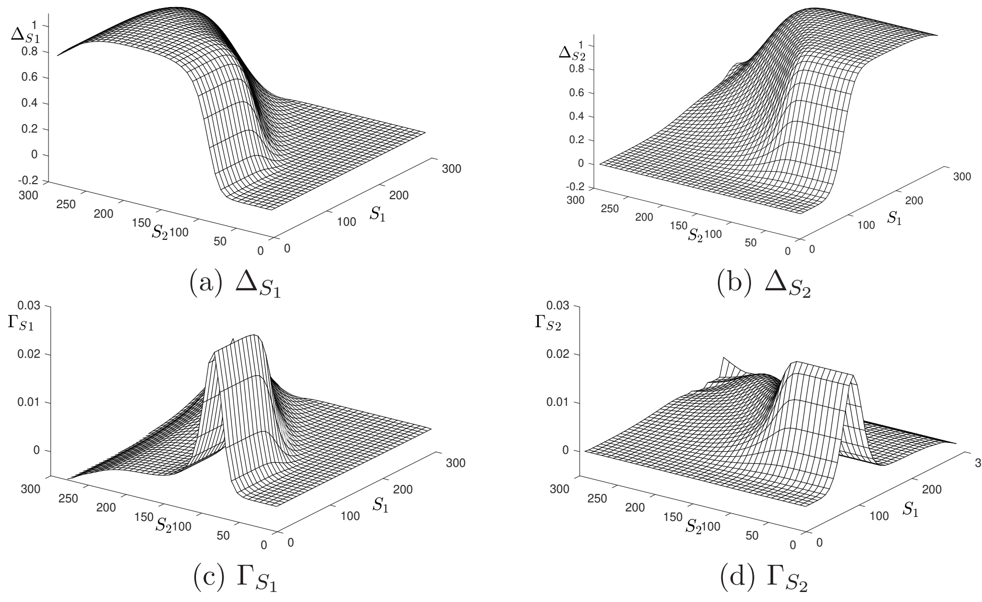
Our proposed method	Fakharany et al. [8]
0.27 (64,32,10)	0.17 (64,32,50)
1.15 (128,64,25)	2.63 (128,64,100)
4.99 (256,128,50)	10.72 (256,128,200)
35.15 (512,256,100)	59.17 (512,256,400)

3.5. Numerical solution of European call on max option In this section, we present numerical tests with a European call on max option using our proposed model. The option is given as $u(x, y, 0) = \max(\max(x, y) - K,\ 0)$ with the strike price K. We use the same parameters as in Section **3.2**. To show the efficiency and accuracy of our algorithm, we also calculate the European call on max option value with the traditional BSM. We take grid $M = 17$ of interval $[-1, 1]$ for numerical integration. The value of the European call on the max option of jump-diffusion model at point $(S_1, S_2) = (100, 100)$ is 16.7712 and the value of the same option of BSM at the same point is 12.0026.



(a) (b) (c)

Figure 8. Numerical solutions of the European call on max option in (a) a two-asset jump-diffusion model, (b) BSM, and (c) difference between two models.



(a) Δ_{S_1} (b) Δ_{S_2}

(c) Γ_{S_1} (d) Γ_{S_2}

Figure 9. Numerical solution of the derivatives of European call on max option in a two-asset jump-diffusion model. (a) and (b) are the first derivative with respect to S_1 and S_2, respectively; (c) and (d) are the second derivative with respect to S_1 and S_2, respectively.

As shown in Figure 8, schematic illustrations from the left to right illustrates option value surfaces of the proposed model and BSM, and the difference of the two models, respectively. Contrary to BSM, the proposed model considers the jump, and there is a particularly large difference in the part where there is a discontinuity.

Our proposed model considering the jump is more fitted to the real financial market than BSM. When dividends occur among shares or unexpected bad news causes a sharp decline in the market, the jumps happen. This is the reason why we say the jump-diffusion model is more fitted to the real financial market.

Figure 9 shows the numerical solution of the derivatives of European call on max option in a two-asset jump-diffusion model. From Figure 9(a) to (d), we present Δ_{S_1}, Δ_{S_2}, Γ_{S_1}, and Γ_{S_2} of the European call on max option in the two-asset jump-diffusion model. Option traders primarily take positions through underlying assets or other options to hedge risks, and the use of greeks can make the hedging more efficient and useful. As shown in Figures 7 and 9, there are many changes nearby in the events of the strike price. Both options are sensitive to the strike price nearby because they may have or may have not payoff depending on whether the underlying price is higher or lower than around the strike price.

4. Conclusions

In this paper, we proposed an efficient and robust finite difference method for option pricing using a two-asset jump-diffusion model, which is a two-dimensional PIDE. To speed up the computational time, we computed the factor matrix A and B so that we can calculate the non-local integral term fast by a simple matrix-vector operation. We used the bilinear interpolation and composite Simpson's rule to solve the integral term of PIDE. We divided it into two regions α and β as shown in Fig. 4(b) and executed interpolation using the plane equation with three points to resolve the overestimate of the option value. In addition, we applied the payoff-consistent extrapolation to get a stable numerical solution. To verify the superiority of the proposed method, we carried out the numerical tests for the valuation of European call on max and European put on min options. The computational results demonstrated that the proposed method is accurate and practical in computing two-asset jump-diffusion model. In this paper, we calculate the option price using jump-term to consider the real financial market. However, we used the constant volatility when we valued the option price. In general, the volatility of the underlying asset is not constant. Therefore, in future work, we will apply the proposed method to a variable volatility option pricing model for more practical application. Moreover, we plan to apply this model to more complicated derivative securities such as ELS (Equity-Linked Securities) that is widely issued in South Korea.

Acknowledgment

The author (Jian Wang) was supported by the China Scholarship Council (201808 260026). The corresponding author (J.S. Kim) was supported by the Brain Korea 21 Plus Project from the Ministry of Education of Korea. The authors thank the reviewers for their constructive and helpful comments on the revision of this article.

References

1. F. Black & M. Scholes: The pricing of options and corporate liabilities. J. Polit. Econ. **81** (1973), no. 3, 637–654.
2. L. Bo & Y. Wang: The pricing of basket options: A weak convergence approach. Oper. Res. Lett. **45** (2017), no. 2, 119–125.
3. J.C. Bogle: Black Monday and black swans. Financ. Anal. J. **64** (2008), no. 2, 30–40.
4. Y. Choi, D. Jeong, J. Kim, Y.R. Kim, S. Lee, S. Seo & M. Yoo: Robust and accurate method for the Black–Scholes equations with payoff-consistent extrapolation. Commun. Korean Math. Soc. **30** (2015), no. 3, 297–311.
5. S.S. Clift & P.A. Forsyth: Numerical solution of two asset jump diffusion models for option valuation. Appl. Numer. Math. **58** (2008), no. 6, 743–782.
6. S.D. Conte & C. de Boor: Elementary Numerical Analysis: An Algorithmic Approach. McGraw-Hill, Singapore, 1980.
7. D.J. Duffy: Finite Difference methods in financial engineering: a Partial Differential Equation approach. John Wiley & Sons, 2013.
8. M. Fakharany, V.N. Egorova & R. Company: Numerical valuation of two-asset options under jump diffusion models using Gauss–Hermite quadrature. J. Comput. Appl. Math. **330** (2018), 822–834.
9. A. Fereshtian, R. Mollapourasl & F. Avram: RBF approximation by partition of unity for valuation of options under exponential Lévy processes. J. Comput. Sci. **32** (2019), 44–55.
10. J. Foo & D. Witkowska: A Comparison of Global Financial Market Recovery after the 2008 Global Financial Crisis. Folia Oeconomica Stetinensia **17** (2017), no. 1, 109–128.
11. S. Gulen, C. Popescu & M. Sari: A New Approach for the Black–Scholes Model with Linear and Nonlinear Volatilities. Mathematics **7** (2019), no. 8, 760.
12. D. Jeong & J. Kim: A comparison study of ADI and operator splitting methods on option pricing models. J. Comput. Appl. Math. **247** (2013), 162–171.

13. D. Jeong, Y.R. Kim, S. Lee, Y. Choi, W.K. Lee, J.M. Shin, H.R. An, H. Hwang & J. Kim: A fast and robust numerical method for option prices and Greeks in a jump-diffusion model. J. Korean Soc. Math. Educ. Ser. B: Pure Appl. Math. **22** (2015), no.2, 159–168.

14. V. Kaushansky, A. Lipton & C. Reisinger: Numerical analysis of an extended structural default model with mutual liabilities and jump risk. J. Comput. Sci. **24** (2018), 218–231.

15. S.G. Kou: A jump-diffusion model for option pricing. Manage. Sci. **48** (2002), no. 8, 1086–1101.

16. S.S. Lee: Jumps and information flow in financial markets. Rev. Financ. Stud. **25** (2011), no. 2, 439–479.

17. R.C. Merton: Option pricing when underlying stock returns are discontinuous. J. Financ. Econ. **3** (1976), no. 1-2, 125–144.

18. R. Rebonato: Volatility and correlation: In the pricing of equity, FX and interest-rate options. John Wiley & Sons, Chichester, 1999.

19. P. Tankov & R. Cont: Financial modelling with jump processes. Chapman and Hall/CRC, London, UK, 2003.

20. P. Tankov & E. Voltchkova: Jump-diffusion models: a practitioners guide. Banque et Marchés **99** (2009), no. 1, 24.

21. Q. Wei: Zero-sum games for continuous-time Markov jump processes with risk-sensitive finite-horizon cost criterion. Oper. Res. Lett. **46** (2018), no. 1, 69–75.

[a]Department of Mathematics, Korea University, Seoul 02841, Republic of Korea
Email address: `cfdkim@korea.ac.kr`

[b]Department of Financial Engineering, Korea University, Seoul 02841, Republic of Korea

[c]School of Finance, College of Business Administration, Soongsil University, Seoul 06978, Korea