J. Korean Soc. Math. Educ. Ser. B: Pure Appl. Math.
http://dx.doi.org/10.7468/jksmeb.2015.22.2.159
Volume 22, Number 2 (May 2015), Pages 159–168

ISSN(Print) 1226-0657
ISSN(Online) 2287-6081

A FAST AND ROBUST NUMERICAL METHOD FOR OPTION PRICES AND GREEKS IN A JUMP-DIFFUSION MODEL

Darae Jeong [a], Young Rock Kim [b], Seunggyu Lee [c], Yongho Choi [d], Woong-Ki Lee [e], Jae-Man Shin [f], Hyo-Rim An [g], Hyeongseok Hwang [h], and Junseok Kim [i,*]

Abstract. We propose a fast and robust finite difference method for Merton's jump diffusion model, which is a partial integro-differential equation. 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. Also, we use non-uniform grids to increase efficiency. We present numerical experiments such as evaluation of the option prices and Greeks to demonstrate a performance of the proposed numerical method. The computational results are in good agreements with the exact solutions of the jump-diffusion model.

1. Introduction

The derivative securities pricing abilities of the jump-diffusion option pricing model generally provide more efficient and robust valuation [7, 11]. The Black–Scholes (BS) model, proposed by Fischer Black and Myron Scholes in 1973 [1], is a mathematical method for option price and is given by

$$\frac{\partial u(x,t)}{\partial t} + \frac{1}{2}\sigma^2 x^2 \frac{\partial^2 u(x,t)}{\partial x^2} + rx\frac{\partial u(x,t)}{\partial x} - ru(x,t) = 0,$$

where x is the underlying asset, t is the time, $u(x,t)$ is the value of an option at t on x, σ is the volatility, and r is the risk neutral interest rate. According to the Merton's jump-diffusion model based on a Poisson process [10], we are able to describe as the following equation: $dx/x = (\mu - \lambda k)dt + \sigma dW + (\eta - 1)d\mathcal{U}$, where dW is the standard Brownian motion and $d\mathcal{U}$ is a Poisson process of intensity λ,

Received by the editors December 05, 2014. Revised February 03, 2015. Accepted Feb. 06, 2015.
2010 *Mathematics Subject Classification.* 65M06, 91G20, 91G60, 91G80.
Key words and phrases. jump-diffusion, Simpson's rule, non-uniform grid, implicit finite difference method, derivative securities.
*Corresponding author.

i.e., $d\mathcal{U}$ is 0 or 1 with probability $1-\lambda dt$ or λdt, respectively. Here η is taken to be a lognormally distributed jump amplitude with probability density

$$\widetilde{G}(\eta) = \frac{1}{\delta\eta\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\log(\eta)}{\delta}\right)^2},$$

where k is the expectation $\mathbb{E}(\eta-1)$ given by $e^{0.5\delta^2}-1$. Since the Brownian motion and Poisson process are supposed to be uncorrelated, the asset price jumps from x to xe^{ξ} if a jump occurs. Then, we are able to get a partial integro-differential equation given by

$$(1.1)\frac{\partial u}{\partial t} + \frac{\sigma^2 x^2}{2}\frac{\partial^2 u}{\partial x^2} + (r-\lambda k)x\frac{\partial u}{\partial x} - ru + \lambda\left(\int_{-\infty}^{\infty} u(xe^{\xi},t)G(\xi)\mathrm{d}\xi - u\right) = 0,$$

where $G(\xi) = \frac{1}{\sqrt{2\pi}\delta} e^{-\frac{1}{2}(\frac{\xi}{\delta})^2}$. The authors in [2] proposed implicit-explicit Runge–Kutta methods for the time integration to solve the integral term explicitly, giving higher-order accuracy schemes under weak stability time-step restrictions. Kwon and Lee developed a second-order convergent finite difference method to solve partial integro-differential equations which describe the behavior of option prices under jump-diffusion models [8]. This paper is organized as follows. In Section 2, we describe the proposed numerical scheme in detail. In Section 3, we present numerical experiments to validate accuracy and efficiency of our proposed algorithm. In Section 4, conclusions are drawn.

2. Numerical Solution

2.1. Discretization with finite differences We will use a finite difference method [5] to numerically solve Eq. (1.1). By changing variable $\tau = T-t$, where T is maturity, we can rewrite Eq. (1.1) as follows:

$$(2.1)\frac{\partial u}{\partial \tau} = \frac{\sigma^2 x^2}{2}\frac{\partial^2 u}{\partial x^2} + (r-\lambda k)x\frac{\partial u}{\partial x} - ru + \lambda\left(\int_{-\infty}^{\infty} u(xe^{\xi},\tau)G(\xi)\mathrm{d}\xi - u\right).$$

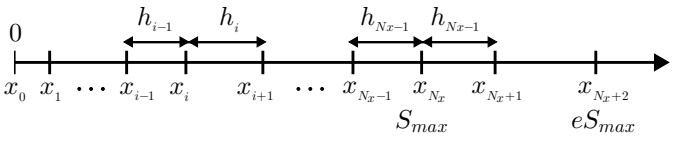


Figure 1. A non-uniform grid.

The jump-diffusion equation is discretized on a 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 and h_i is the grid spacing, see Fig. 1. And we assume that $x_{N_x} = S_{\max}$, $x_{N_x+1} = S_{\max} + h_{N_x-1}$, and $x_{N_x+2} = eS_{\max}$. Let us denote the numerical approximation of the solution by $u_i^n \approx u(x_i, n\Delta\tau)$, where $\Delta\tau = T/N_\tau$ is the time-step size and N_τ is the total number of time-steps. By applying an implicit scheme to Eq. (2.1), we have

$$\begin{aligned} \frac{u_i^{n+1} - u_i^n}{\Delta\tau} &= \frac{\sigma^2 x_i^2}{2}\left(\frac{\partial^2 u}{\partial x^2}\right)_i^{n+1} + (r - \lambda k)\, x_i \left(\frac{\partial u}{\partial x}\right)_i^{n+1} - r u_i^{n+1} \\ &\quad + \lambda\left(I_a^b[u\left(x_i e^{\xi}, n\Delta\tau\right) G(\xi)]\Delta\xi - u_i^n\right), \end{aligned} \tag{2.2}$$

where the first and second derivatives are defined as

$$\begin{aligned} \left(\frac{\partial u}{\partial x}\right)_i^{n+1} &= -\frac{h_i}{h_{i-1}(h_{i-1}+h_i)} u_{i-1}^{n+1} + \frac{h_i - h_{i-1}}{h_{i-1}h_i} u_i^{n+1} + \frac{h_{i-1}}{h_i(h_{i-1}+h_i)} u_{i+1}^{n+1}, \\ \left(\frac{\partial^2 u}{\partial x^2}\right)_i^{n+1} &= \frac{2}{h_{i-1}(h_{i-1}+h_i)} u_{i-1}^{n+1} - \frac{2}{h_{i-1}h_i} u_i^{n+1} + \frac{2}{h_i(h_{i-1}+h_i)} u_{i+1}^{n+1}, \end{aligned}$$

and $I_a^b g(\xi)\Delta\xi$ is an approximate numerical quadrature of $\int_{-\infty}^{\infty} g(\xi)\mathrm{d}\xi$. Using the composite Simpson's rule, we can rewrite Eq. (2.2) as

$$\alpha_i u_{i-1}^{n+1} + \beta_i u_i^{n+1} + \gamma_i u_{i+1}^{n+1} = f_i^n, \tag{2.3}$$

$$\text{where } \alpha_i = \frac{-\sigma^2 x_i^2 + (r - \lambda k)\, x_i h_i}{h_{i-1}(h_{i-1}+h_i)},\ \beta_i = \frac{\sigma^2 x_i^2 - (r - \lambda k)\, x_i (h_i - h_{i-1})}{h_{i-1}h_i} + r + \frac{1}{\Delta\tau},$$
$$\gamma_i = \frac{-\sigma^2 x_i^2 - (r - \lambda k)\, x_i h_{i-1}}{h_i(h_{i-1}+h_i)},\ f_i^n = \left(\frac{1}{\Delta\tau} - \lambda\right) u_i^n + \lambda I_a^b u\left(x_i e^{\xi}, n\Delta\tau\right) G(\xi)\,\Delta\xi.$$

We impose the zero Dirichlet boundary condition at $x = 0$ and the linear boundary condition at $x = S_{\max}$ [13]. The matrix form of the linear system (2.3) can be rewritten as

$$\begin{pmatrix} \beta_1 & \gamma_1 & 0 & \dots & 0 \\ \alpha_2 & \beta_2 & \gamma_2 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \alpha_{N_x-1} & \beta_{N_x-1} & \gamma_{N_x-1} \\ 0 & \dots & 0 & \alpha_{N_x} - \gamma_{N_x} & \beta_{N_x} + 2\gamma_{N_x} \end{pmatrix} \begin{pmatrix} u_1^{n+1} \\ u_2^{n+1} \\ \vdots \\ u_{N_x-1}^{n+1} \\ u_{N_x}^{n+1} \end{pmatrix} = \begin{pmatrix} f_1^n \\ f_2^n \\ \vdots \\ f_{N_x-1}^n \\ f_{N_x}^n \end{pmatrix}.$$

We solve the linear system by using the Thomas algorithm, which inverts a tri-

diagonal matrix directly. Next, we consider the numerical quadrature of the following integral term

$$\int_{-\infty}^{\infty} u(xe^{\xi},\tau)G(\xi)\mathrm{d}\xi. \tag{2.4}$$

To evaluate the integral term (2.4), we truncate the infinite domain $(-\infty,\infty)$ to $[a,b]$. Let us consider $[a,b]=[-1,1]$ case to be concrete. Therefore we use the composite Simpson's rule for the numerical integration.

$$\begin{aligned} &I_{-1}^{1}u\left(x_ie^{\xi},n\Delta\tau\right)G\left(\xi\right)\Delta\xi \\ &\quad = I_{-1}^{1}g(\xi)\Delta\xi = \frac{\Delta\xi}{3}\left[g(\xi_0)+2\sum_{j=1}^{M/2-1}g(\xi_{2j})+4\sum_{j=1}^{M/2}g(\xi_{2j-1})+g(\xi_M)\right], \end{aligned} \tag{2.5}$$

where M is a positive even integer. Figure 2(a) shows a schematic illustration of the option price $u(x,\tau)$ and $G(xe^{\xi})$ function.

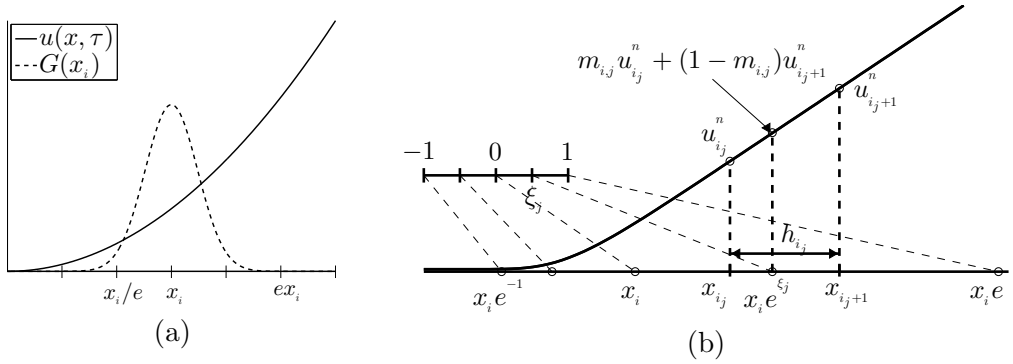


Figure 2. (a) Schematic illustration of the option price $u(x,\tau)$ and $G(x_ie^{\xi})$ function. (b) Linear interpolation of $u(x,\tau)$ at $x=x_ie^{\xi_j}$.

To accelerate the computation, we save the linear interpolation factors at each point x_i before the main time-step iterations. The factor matrix A is defined as

$$A=(m_{i,j})_{N_x\times(M-1)}, \tag{2.6}$$

where $m_{i,j}=(x_ie^{\xi_j}-x_{i_j})/h_{i_j}$, $x_{i_j}\le x_ie^{\xi_j}<x_{i_j+1}$, for some $0\le i_j\le N_x+1$. Then, we have $g(\xi_j)=(m_{i,j}u^n_{i_j}+(1-m_{i,j})u^n_{i_j+1})G(\xi_j)$, which is schematically illustrated in Fig. 2(b).

3. Numerical Experiment

In this section, we perform various numerical tests to show efficiency and robustness of the proposed numerical scheme under the jump-diffusion model. All computations were run in MATLAB version 7.7 [12].

3.1. European call option We consider a European call option whose payoff is given as $u(x,0) = \max(x-K,0)$, where K is strike price. We will use $K = 100, r = 0.03, \sigma = 0.3, \lambda = 0.1$, $\delta = 0.1$, and uniform grid sizes h unless otherwise specified. Closed-form solution $u_{\text{exact}}(x,t)$ of jump-diffusion model for European call option is formulated as an infinite series of Black, Scholes, and Merton's model [1, 10]:

$$u_{\text{exact}}(x,\tau) = \sum_{n=0}^{\infty} e^{-\hat{\lambda}\tau}\frac{(\hat{\lambda}\tau)^n}{n!}[x\Phi(d_{1,n}) - Ke^{-r_n\tau}\Phi(d_{2,n})], \tag{3.1}$$

$$\text{where } d_{1,n} = \frac{\ln\left(\frac{x}{K}\right) + \left(r_n + \frac{\sigma_n^2}{2}\right)\tau}{\sigma_n\sqrt{\tau}}, d_{2,n} = d_{1,n} - \sigma_n\sqrt{\tau}, r_n = r - \lambda k + \frac{n\ln(1+k)}{\tau},$$

$$\hat{\lambda} = \lambda(1+k),\ k = e^{\frac{\delta^2}{2}} - 1,\ \sigma_n^2 = \sigma^2 + \frac{n\delta^2}{\tau},\ \Phi(d) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{d} e^{-\frac{x^2}{2}}\,\mathrm{d}x.$$

We sum Eq. (3.1) up to $n = 200$. Figure 3 shows the payoff function and two numerical solutions at $\tau = 1$ with the BS and the BS with jump-diffusion models. Here, we take $h = 1$ and $\Delta\tau = 1/360$. The result shows that the option values with the jump-diffusion model are larger than those with the BS model without the jump-diffusion.

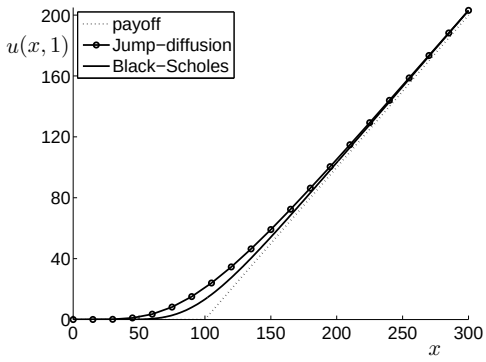


Figure 3. Payoff function and two numerical solutions at $\tau = 1$ with the BS and the BS with jump-diffusion models.

3.2. Effect of interval size on the numerical quadrature To find a suitable interval $[-b, b]$ for numerical integration, we test the effect of b on the results with $M = 50$. Figure 4 shows the numerical solutions at $\tau = 1$ with various values of b, such as 0.2, 0.3, 0.4, 0.8, 1, and 2. The result suggests that $b = 1$ is accurate enough. Therefore, we will use $[-1, 1]$ in the calculation of the integral term (2.5) from now on.

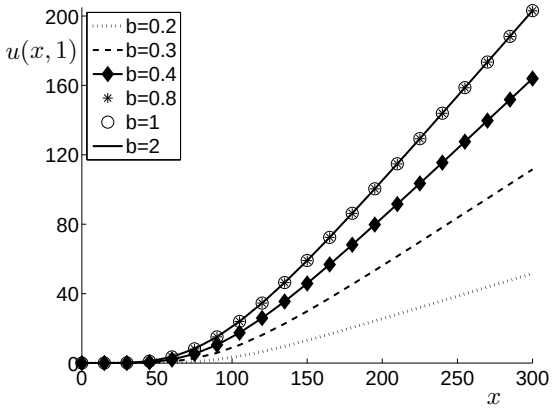


Figure 4. Numerical solutions at $\tau = 1$ with various values of b, such as, 0.2, 0.3, 0.4, 0.8, 1, and 2.

3.3. Convergence test We perform a convergence test to verify the accuracy of the proposed numerical method in this paper. In this test, we use the following parameters: $S_{\max} = 300$, and $T = 1$. The numerical solutions are computed with the uniform grids $h = 2^{2-n}$ and the uniform time-steps $\Delta\tau = 250h^2$ for $n = 1, 2$, and 3. We denote the error as $e = u - u_{\text{exact}}$, where u and u_{exact} are the numerical and analytic solutions, respectively. And we calculate its discrete l_2-norm error as $\|e\|_2 = \sqrt{\sum_{i=1}^{N_x} e_i^2 / N_x}$. Table 1 shows the l_2-norm errors and convergence rates for numerical solutions with respect to h. This result indicates that the scheme is first-order accurate in time and second-order accurate in space.

Table 1. l_2-norm errors and convergence rates for numerical solutions on uniform grid with respect to h.

h	2		1		0.5
error	0.0081574192		0.0020578644		0.0005628202
rate		1.986965		1.870402	

3.4. Computation of the Greeks We compute the Greeks by using the proposed scheme and compare them with the closed-form solutions. Delta (Δ) is the first derivative with respect to the underlying asset x.

$$\Delta = \frac{\partial u_{\text{exact}}}{\partial x} = \sum_{n=0}^{\infty} e^{-\hat{\lambda}\tau} \frac{(\hat{\lambda}\tau)^n}{n!} \Phi(d_{1,n}) \approx \frac{u_{i+1}^{N_t} - u_{i-1}^{N_t}}{2h}.$$

Gamma (Γ) is the second derivative with respect to the underlying asset x.

$$\Gamma = \frac{\partial^2 u_{\text{exact}}}{\partial x^2} = \sum_{n=0}^{\infty} e^{-\hat{\lambda}\tau} \frac{(\hat{\lambda}\tau)^n}{n!} \frac{e^{-d_{1,n}^2/2}}{\sigma_n x\sqrt{2\pi\tau}} \approx \frac{u_{i+1}^{N_t} - 2u_i^{N_t} + u_{i-1}^{N_t}}{h^2}.$$

Vega ($\mathcal{V}$) is the derivative with respect to volatility σ.

$$\mathcal{V} = \frac{\partial u_{\text{exact}}}{\partial \sigma} = \sum_{n=0}^{\infty} e^{-\hat{\lambda}\tau} \frac{(\hat{\lambda}\tau)^n}{n!} \frac{x\sqrt{\tau}}{\sqrt{2\pi}} e^{-0.5d_{1,n}^2} \approx \frac{u_i^{N_t}(\sigma + 0.5\Delta\sigma) - u_i^{N_t}(\sigma - 0.5\Delta\sigma)}{\Delta\sigma}.$$

Rho (ρ) is the derivative with respect to the risk-free interest rate r.

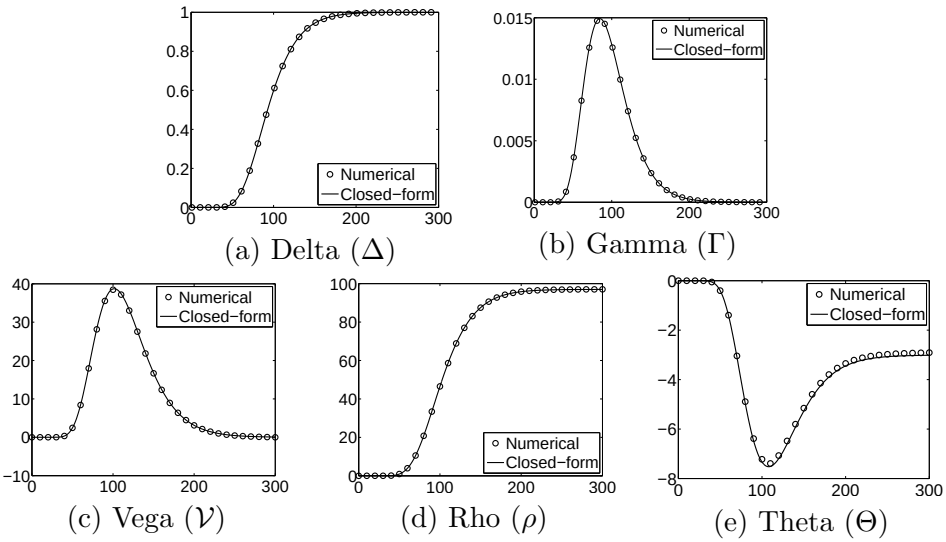
$$\rho = \frac{\partial u_{\text{exact}}}{\partial r} = \sum_{n=0}^{\infty} \tau e^{-\hat{\lambda}\tau} \frac{(\hat{\lambda}\tau)^n}{n!} K e^{-r_n\tau} \Phi(d_{2,n}) \approx \frac{u_i^{N_t}(r + 0.5\Delta r) - u_i^{N_t}(r - 0.5\Delta r)}{\Delta r}.$$

Theta (Θ) is the derivative with respect to the time t.

$$\begin{aligned}
\Theta = \frac{\partial u_{\text{exact}}}{\partial t} &= \sum_{n=0}^{\infty} e^{-\hat{\lambda}\tau} \frac{(\hat{\lambda}\tau)^n}{n!} \left[\hat{\lambda}\left\{x\Phi(d_{1,n}) - Ke^{-r_n\tau}\Phi(d_{2,n})\right\}\right. \\
&\quad \left. -Ke^{-r_n\tau}\left\{r_n\Phi(d_{2,n}) + \frac{\sigma_n}{2\sqrt{\tau}}\Phi'(d_{2,n})\right\}\right] \\
&\quad -\sum_{n=1}^{\infty} e^{-\hat{\lambda}\tau} \frac{(\hat{\lambda}\tau)^n}{n!} \frac{n}{\tau}\left[x\Phi(d_{1,n}) - Ke^{-r_n\tau}\Phi(d_{2,n})\right] \\
&\approx \frac{u_i^{N_t+1} - u_i^{N_t-1}}{2\Delta\tau}.
\end{aligned}$$

The parameters used in the calculations of numerical and closed-form solutions are as follows: $\Delta\sigma = 0.1$, $\Delta r = 0.01$, $\Delta t = 1/360$, $S_{\max} = 300$, and $h = 1$ on uniform grid. Figure 5(a), (b), (c), (d), and (e) are the results of Δ, Γ, $\mathcal{V}$, ρ, and Θ from the numerical and closed-form solutions, respectively.

3.5. Efficiency of non-uniform grid In this section, we show the efficiency of non-uniform grid on jump-diffusion model. We first evaluate the numerical solution with $S_{\max} = 600$, $N_x = 4800$ and $N_\tau = 1440$ on uniform grid. The following Table 2 represents the l_2-norm error and relative CPU time on uniform and non-uniform grids, respectively.



(a) Delta (Δ) (b) Gamma (Γ)

(c) Vega ($\mathcal{V}$) (d) Rho (ρ) (e) Theta (Θ)

Figure 5. Comparison between the numerical and closed-form solutions for the Greeks. (a), (b), (c), (d), and (e) are the results of Δ, Γ, $\mathcal{V}$, ρ, and Θ, respectively.

Table 2. Comparison with l_2-norm error on the interval [80, 120] and relative CPU time for numerical solutions on uniform and non-uniform grid.

	N_x	error	relative CPU time
Uniform grid	4800	9.20e-4	1
Adaptive grid	1086	1.03e-3	0.19391

Here, l_2-norm error is evaluated on the interval [80, 120], which is indicated a neighborhood of the exercise price 100. And we have numerical solution on non-uniform grid, which is defined as

$$x = [0\ 1.11 : 1.11 : 38.89\ 40 : 0.25 : 160\ 164.89 : 4.89 : 595.11\ 600].$$

In Table 2, 1 in CPU time of uniform grid stands for the relative calculation time for uniform grid. And the CPU time of non-uniform grid is scaled with that of the uniform grid. Through this test, we can confirm that non-uniform grid is more efficient than the uniform grid.

4. Conclusions

In this paper, we presented an efficient, fast, and robust finite difference method

for the valuation of European call options with jump-diffusion processes. The mathematical model for the process is a partial integro-differential equation (1.1), Merton's jump diffusion model. To speed up the computational time, we computed the factor matrix A in (2.6) so that we can calculate the non-local integral term fast by a simple matrix-vector operation. Also, we used non-uniform grids to increase efficiency. We presented numerical experiments such as evaluation of the option prices and the Greeks to demonstrate the performance of the proposed numerical method. The computational results were in good agreements with the exact solutions of the jump-diffusion model.

Acknowledgement

The corresponding author (J.S. Kim) was supported by The Small and Medium Business Administration.

References

1. F. Black & M. Scholes: The pricing of options and corporate liabilities. *J. Polit. Econ.* **81** (1973), 637–654.
2. M. Briani, R. Natalini & G. Russo: Implicit-explicit numerical schemes for jump-diffusion processes. *Calcolo* **44** (2007), no. 1, 33–57.
3. P. Carr & L. Cousot: A PDE approach to jump-diffusions. *Quant. Fin.* **11** (2011), no. 1, 33–52.
4. G. Cheang & C. Chiarella: A modern view on Merton's jump-diffusion model. *Quant. Fin. Res. Cent.* **287**, (2011).
5. D.J. Duffy: *Finite Difference methods in financial engineering: a Partial Differential Equation approach*, John Wiley & Sons, (2006).
6. L. Feng & V. Linetsky: Pricing options in jump-diffusion models: an extrapolation approach. *Oper. Res.* **56**, (2008), no. 2, 304–325.
7. J.W. Kremer & R.L. Roenfeldt: Warrant pricing: jump-diffusion vs. Black-Scholes. *J. Finan. Quant. Anal.* **28** (1993), no. 2, 255–272.
8. Y.H. Kwon & Y. Lee: A second-order finite difference method for option pricing under jump-diffusion models, *SIAM J. Numer. Anal.* **49** (2011), no. 6, 2598–2617.
9. S. Lee, Y. Li, Y. Choi, H. Hwang & J. Kim: Accurate and efficient computations for the Greeks of European multi-asset options. *J. KSIAM* **18** (2014), no. 1, 61–74.
10. R.C. Merton: Option pricing when underlying stock returns are discontinuous. *J. Polit. Econ.* **3** (1976), no. 1, 125–144.
11. P. Tankov: *Financial modelling with jump processes.* CRC press. (2003).

12. Using MATLAB, The MathWorks Inc., Natick, MA, 1998, http://www.mathworks.com
13. H. Windcliff, P.A. Forsyth & K.R. Vetzal: Analysis of the stability of the linear boundary condition for the Black-Scholes equation. *J. Comput. Fin.* **8** (2004), 65–92.

[a]DEPARTMENT OF MATHEMATICS, KOREA UNIVERSITY, SEOUL 136-713, REPUBLIC OF KOREA
Email address: tinayoyo@korea.ac.kr

[b]MAJOR IN MATHEMATICS EDUCATION, HANKUK UNIVERSITY OF FOREIGN STUDIES, SEOUL, 130-791, REPUBLIC OF KOREA
Email address: rocky777@hufs.ac.kr

[c]DEPARTMENT OF MATHEMATICS, KOREA UNIVERSITY, SEOUL 136-713, REPUBLIC OF KOREA
Email address: sky509@korea.ac.kr

[d]DEPARTMENT OF MATHEMATICS, KOREA UNIVERSITY, SEOUL 136-713, REPUBLIC OF KOREA
Email address: poohyongho@korea.ac.kr

[e]BUSINESS SCHOOL, KOREA UNIVERSITY, SEOUL 136-071, REPUBLIC OF KOREA
Email address: unggii@korea.ac.kr

[f]DEPARTMENT OF FINANCIAL ENGINEERING, KOREA UNIVERSITY, SEOUL 136-071, REPUBLIC OF KOREA
Email address: sim00@korea.ac.kr

[g]DEPARTMENT OF FINANCIAL ENGINEERING, KOREA UNIVERSITY, SEOUL 136-071, REPUBLIC OF KOREA
Email address: ahr78@korea.ac.kr

[h]DEPARTMENT OF FINANCIAL ENGINEERING, KOREA UNIVERSITY, SEOUL 136-071, REPUBLIC OF KOREA
Email address: hhs288@korea.ac.kr

[i]DEPARTMENT OF MATHEMATICS, KOREA UNIVERSITY, SEOUL 136-713, REPUBLIC OF KOREA
Email address: cfdkim@korea.ac.kr