Fourth Order Compact Scheme with Local Mesh Refinement for Option Pricing in Jump-Diffusion Model *

Spike T. Lee† Hai-Wei Sun‡

Abstract

The value of a contingent claim under a jump-diffusion process satisfies a partial integro-differential equation. A fourth order compact finite difference scheme is applied to discretize the spatial variable of this equation. It is discretized in time by an implicit-explicit method. Meanwhile, a local mesh refinement strategy is used for handling the non-smooth payoff condition. Moreover, the numerical quadrature method is exploited to evaluate the jump integral term. It guarantees a Toeplitz-like structure of the integral operator such that a fast algorithm is feasible. Numerical results show that this approach gives fourth order accuracy in space.

Key words: partial integro-differential equation, fourth order compact scheme, local mesh refinement, jump-diffusion, Toeplitz matrix
Mathematics Subject Classification: 65T50; 65M06; 65M12; 91B28

1 Introduction

In 1973, Black and Scholes [5] proposed a formula for pricing financial option contracts in pure-diffusion models. Nevertheless, empirical studies have revealed that these models are not capable of specifying the price dynamics when used for pricing options under real market circumstances. A large amount of models have been proposed to tackle these shortcomings since then. One of them is the jump-diffusion model presented by Merton [18] in 1976. In Merton's model, the asset return follows a standard Wiener process driven by a compound Poisson process with normally distributed jumps. Furthermore, volatility smiles and skews can be generated under this model by choosing the parameters of the jump process properly [3]. It reflects a more realistic condition of the market movements. Kou [17] also suggested a double exponential jump-diffusion model where jump sizes are double exponentially distributed. Aside from jump-diffusion standards, different models concerning stochastic volatility and jumps are popular among others [4, 10, 15].

*The research was partially supported by the research grant 033/2009/A from FDCT of Macao, UL020/08-Y2/MAT/JXQ01/FST and RG063/08-09S/SHW/FST from University of Macau.

†Department of Mathematics, University of Macau, Macao, China (ma76522@umac.mo).

‡Corresponding author. Department of Mathematics, University of Macau, Macao, China (HSun@umac.mo).

The valuation of a contingent claim under a jump-diffusion process requires solving a partial integro-differential equation (PIDE). This type of equation contains differential operators and a non-local integral term. Amin [2] proposed a first order accurate in time, explicit approach based on multinomial trees. However, the restrictive stability condition needs small time steps to achieve better accuracy. Zhang [28] suggested an implicit-explicit (IMEX) scheme, where the differential part is treated implicitly and the integral part explicitly. The IMEX scheme avoids inverting dense matrices with explicit treatment of the integral term. High order time stepping schemes are proposed as well because the IMEX scheme has slow first order convergence in time. Andersen and Andreasen [3] used a second order accurate in time, operator-splitting alternating directions implicit (ADI) method with fast Fourier transform (FFT) evaluation of the convolution integral. d'Halluin et al. [14] also developed a second order accurate in time, Crank-Nicolson time stepping scheme for the PIDE. However, it requires a fixed point iterative procedure to solve the dense system of equations owing to the implicit treatment of the integral term. Almendral and Oosterlee [1] used a matrix splitting technique and a second order backward difference formula (BDF2) to price European options under jump diffusion processes. Feng and Linetsky [11] improved the IMEX scheme by a new high order extrapolation approach. This method is independent of the choice of spatial discretization and suitable for any PIDE solver based on the IMEX scheme.

Most of the existing schemes exploit standard central difference discretization in the spatial direction, which only has second order convergence. Nevertheless, high order schemes are not customary tools for option pricing because financial payoff functions are always non-smooth. As a result, it will affect the convergence rate of high order schemes. In this paper we are particularly interested in the fourth order compact (FOC) finite difference scheme in [12]. The FOC scheme was widely studied for tackling convection-diffusion equations in the past years [21, 22, 25, 26]. Recently Tangman et al. [23] suggested the FOC scheme with a grid stretching approach for numerical pricing of options under Black-Scholes pure-diffusion model, but it remains uncertain how to generalize this method to a jump-diffusion case.

In this paper, we focus on pricing European call options and double barrier call options for a single underlying asset in Merton's jump-diffusion model. We exploit the FOC scheme to solve the PIDE. In time direction, we prefer the easily implemented IMEX scheme coupled with extrapolation approach following Feng and Linetsky's idea [11]. To emphasize the spatial convergence behavior, we simply employ the Richardson extrapolation, which is less accurate than the one in [11] but comparatively more straightforward, to achieve high order accuracy in time. Instead of coordinate transformations, a local mesh refinement strategy is used for handling the non-smooth payoff condition. For the integral term, we exploit the numerical quadrature method which guarantees the Toeplitz-like structure of the integral operator. Thus fast algorithms for matrix-vector multiplication can be applied. The rest of the paper is arranged as follows. In section 2 we review the jump-diffusion model for option pricing and illustrate the IMEX scheme with extrapolation approach. In section 3 we propose the FOC scheme for the PIDE, carry out a stability analysis for

the FOC scheme and introduce the local mesh refinement strategy. Issues regarding the evaluation of the non-local integral term are discussed in section 4. In section 5 we present the numerical results. Finally we give the concluding remarks in section 6.

2 Option pricing and IMEX scheme

2.1 Option pricing in jump-diffusion models

Since the pure-diffusion model presented by Black and Scholes [5] cannot portray the price dynamics under large market movements, more general models such as the exponential Lévy models have been proposed. In exponential Lévy models, the market price of the asset S_τ with time $\tau \in [0, T]$ and T being the maturity time is represented as the exponential of a Lévy process X_τ. The Lévy process X_τ on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ is a stochastically continuous process with stationary independent increments. For convenience, we model the asset value by [1]:

$$S_\tau = S_0 e^{X_\tau}$$

on the filtered probability space $(\Omega, \mathcal{F}, \mathcal{F}_\tau, \mathbb{P})$, where $\mathcal{F}_\tau$ is the filtration representing the history of the asset. Particularly in jump-diffusion models, the log-price of the underlying contains a diffusive component and jumps can occur at Poisson times. Then the risk-neutral dynamics of the asset price S_τ is described by the following diffusion process [3, 14]:

$$\frac{dS_\tau}{S_\tau} = (r - \delta - \lambda\kappa)d\tau + \sigma dz + (\eta - 1)dq,$$

where $r \geq 0$ is the risk-free interest rate, $\delta \geq 0$ is the continuous dividend yield, $\lambda > 0$ is the arrival intensity of the Poisson process dq, $\sigma > 0$ is the stock return volatility, dz is the increment of a Brownian motion, and $\eta - 1$ is an impulse function making S jump to $S\eta$. We denote the expectation of the impulse function by $\kappa = \mathbb{E}(\eta - 1)$. It is assumed that the Poisson and the Brownian motion processes are uncorrelated. Without loss of generality, we only illustrate the framework with zero dividend $\delta = 0$.

Suppose φ is a T-maturity payoff for a European option contract. Then the value of the option $w(S, \tau)$ is the discounted expectation under risk-neutral measure $\mathbb{Q}$ [9, 11]:

$$w(S, \tau) = e^{-r(T-\tau)}\mathbb{E}_\mathbb{Q}[\varphi(S_T)].$$

By Feynman-Kac formula [24], $w(S, \tau)$ can be computed by solving a backward PIDE on $[0, +\infty) \times [0, T]$:

$$-w_\tau = \frac{\sigma^2}{2}S^2 w_{SS} + (r - \lambda\kappa)Sw_S - (r + \lambda)w + \lambda \int_0^\infty w(S\eta, \tau)g(\eta)d\eta, \tag{2.1}$$

where $g(\eta)$ is the density function of the jump size distribution.

It is common to make the exponential change of variables

$$S = Ke^x \quad \text{and} \quad \eta = e^z,$$

where K is the strike price, and reverse the time direction

$$t = T - \tau.$$

Then equation (2.1) reduces to a forward PIDE of $v(x,t)$ on $(-\infty,+\infty)\times[0,T]$:

$$v_t = \frac{\sigma^2}{2}v_{xx} + (r - \lambda\kappa - \frac{\sigma^2}{2})v_x - (r+\lambda)v + \lambda\int_{-\infty}^{\infty} v(x+z,t)f(z)dz, \tag{2.2}$$

where $f(z) = g(e^z)e^z$ and $v(x+z,t) = w(Ke^{x+z}, T-t)$.

For a European call option, the initial condition is

$$v(x,0) = \max(Ke^x - K, 0).$$

Moreover, the corresponding boundary conditions are

$$v(x,t) \approx \begin{cases} 0, & \text{as } x \to -\infty, \\ Ke^x - Ke^{-rt}, & \text{as } x \to +\infty. \end{cases} \tag{2.3}$$

Note that European put options can be handled in a similar way.

We would also like to consider a double barrier call option of European style [11]. This option acts like a vanilla European call option and they share the same payoff condition. The difference is that the contract becomes void when the stock S goes below the lower barrier level D or increases above the upper barrier level U. With regard to the exponential change $S = Ke^x$, the lower and upper barriers are $x_d = \log(D/K)$ and $x_u = \log(U/K)$ respectively. Thus for a double barrier call option, we have the following homogeneous boundary conditions:

$$v(x,t) = \begin{cases} 0, & \text{as } x \le x_d, \\ 0, & \text{as } x \ge x_u. \end{cases}$$

Our aim is to find the option value $v(x,t)$ at $t = T$. In this paper, we consider the jump-diffusion model presented by Merton [18] where jumps are normally distributed with mean μ and variation γ. The jump size distribution $g(\eta)$ is given by

$$g(\eta) = \frac{e^{-[\log(\eta)-\mu]^2/2\gamma^2}}{\sqrt{2\pi}\gamma\eta}.$$

After changing the variable, we have

$$f(z) = g(e^z)e^z = \frac{e^{-(z-\mu)^2/2\gamma^2}}{\sqrt{2\pi}\gamma}. \tag{2.4}$$

Note that $f \ge 0$ is the probability density function of the Gaussian distribution and hence

$$\int_{-\infty}^{\infty} f(z)dz = 1. \tag{2.5}$$

2.2 IMEX scheme and extrapolation approach

It is numerically challenging to solve a PIDE because of the non-local integral term. In the following, we introduce the IMEX scheme.

In 1997, Zhang [28] suggested the IMEX scheme for option pricing and then this method becomes accepted by others [9, 11, 24]. Suppose a PIDE is given by:

$$u_t = \mathcal{D}u + \mathcal{I}u,$$

where $\mathcal{D}$ is a differential operator and $\mathcal{I}$ is an integral operator. As $\mathcal{I}$ usually makes the system dense, we utilize the IMEX scheme which is noted for avoiding dense matrices inversion. First the time interval $[0, T]$ is divided into J time steps, with $\Delta t = T/J$ and $t_j = j\Delta t,\ j = 0, 1, \ldots, J$. Let u^j be the approximation at time level t_j, we then apply the IMEX Euler scheme, where the differential part is treated implicitly and the integral part is treated explicitly for $j = 0, 1, \ldots, J-1$:

$$\frac{u^{j+1} - u^j}{\Delta t} = \mathcal{D}u^{j+1} + \mathcal{I}u^j,$$

or

$$u^{j+1} - \Delta t\mathcal{D}u^{j+1} = u^j + \Delta t\mathcal{I}u^j. \tag{2.6}$$

Our ultimate goal is to find u^J, the option values at time T. At each time step, the linear system (2.6) is solved to determine the vector u^{j+1} with u^0 being the given initial guess. However, the IMEX scheme is only first order accurate in time. Note that the time-dependent option function generated by a European style option contract is smooth. Suppose the IMEX scheme is unconditionally stable, we can employ the Richardson extrapolation method to raise accuracy. By combining the IMEX scheme with Richardson extrapolation method, we manage to avoid solving dense systems and at the same time attain high order convergence. Let $u_{p,1}$ be the solution obtained at time T with step size $\Delta t/2^{p-1}$, $p = 1, 2, \ldots, s$, where s is the stage number. Then the Richardson extrapolation formula is given by

$$u_{p,q} = \frac{2^{q-1}u_{p,q-1} - u_{p-1,q-1}}{2^{q-1} - 1}, \quad p = 2, 3, \ldots, s, \quad q = 2, 3, \ldots, p. \tag{2.7}$$

By this process, we have achieved a better approximation $u_{s,s}$ at the sth stage.

Recall that the PIDE (2.2) on $(-\infty, +\infty) \times [0, T]$ is given by

$$v_t = av_{xx} + bv_x - (r + \lambda)v + \lambda(v * f), \tag{2.8}$$

where

$$a = \frac{1}{2}\sigma^2 > 0, \qquad b = r - \lambda\kappa - \frac{\sigma^2}{2}, \tag{2.9}$$

and the convolution

$$(v * f)(x, t) = \int_{-\infty}^{\infty} v(x + z, t)f(z)dz = \int_{-\infty}^{\infty} v(y, t)f(y - x)dy.$$

Let v^j and $(v*f)^j$ denote the approximations to v and $v*f$ respectively at time level t_j, $j = 1, 2, \ldots, J$. Then the IMEX scheme is applied as discussed previously:

$$\frac{v^{j+1} - v^j}{\Delta t} = a v_{xx}^{j+1} + b v_x^{j+1} - (r+\lambda) v^{j+1} + \lambda (v*f)^j,$$

or

$$-a\Delta t v_{xx}^{j+1} - b\Delta t v_x^{j+1} + [1 + (r+\lambda)\Delta t] v^{j+1} = v^j + \lambda \Delta t (v*f)^j. \tag{2.10}$$

We then proceed to consider the spatial discretization of the semi-discretized equation (2.10).

3 FOC scheme and local mesh refinement

3.1 FOC scheme

We briefly introduce how to obtain an FOC scheme for (2.10); see [21] for details. For simplicity, we assume that the coefficients of (2.8) are constants. Note that the boundary conditions can be imposed if a finite computational domain is used [14]. Hence we truncate the infinite x-domain $(-\infty, \infty)$ to $[x_{\min}, x_{\max}]$ for European call options. For double barrier call options, it is natural to set $x_{\min} = x_d$ and $x_{\max} = x_u$ since the option values are canceled outside. The finite domain is then divided into L_0 steps with mesh size

$$\Delta x = \frac{x_{\max} - x_{\min}}{L_0}. \tag{3.11}$$

Then the uniform computational grid with $L_0 + 1$ points is denoted by

$$\{x_0, x_1, \ldots, x_{L_0}\} \equiv \Omega_x \cup \{x_{\min}, x_{\max}\},$$

where $x_0 = x_{\min}$ and $x_{L_0} = x_{\max}$. Here we assume that L_0 is an even number.

Define the central finite difference operators at the grid point $x_l \in \Omega_x$ as follows:

$$\delta_x v_l^j = \frac{v_{l+1}^j - v_{l-1}^j}{2\Delta x} \quad \text{and} \quad \delta_x^2 v_l^j = \frac{v_{l+1}^j - 2v_l^j + v_{l-1}^j}{\Delta x^2},$$

where Δx is the mesh size in the spatial direction and v_l^j is an approximation to $v(x_l, t_j)$. Moreover, $(v*f)_l^j$ is an approximation to $(v*f)(x_l, t_j)$. By Taylor's theorem, we have the following relations:

$$\delta_x v_l^{j+1} = (v_x)_l^{j+1} + \frac{\Delta x^2}{6}(v_{xxx})_l^{j+1} + \mathcal{O}(\Delta x^4),$$

$$\delta_x^2 v_l^{j+1} = (v_{xx})_l^{j+1} + \frac{\Delta x^2}{12}(v_{xxxx})_l^{j+1} + \mathcal{O}(\Delta x^4),$$

where $(v_x)_l^j$ is an approximation to $v_x(x_l, t_j)$ and so on. Hence $(v_x)_l^{j+1}$ and $(v_{xx})_l^{j+1}$ can be approximated by

$$(v_x)_l^{j+1} = \delta_x v_l^{j+1} - \frac{\Delta x^2}{6}(v_{xxx})_l^{j+1} + \mathcal{O}(\Delta x^4),$$

$$(v_{xx})_l^{j+1} = \delta_x^2 v_l^{j+1} - \frac{\Delta x^2}{12}(v_{xxxx})_l^{j+1} + \mathcal{O}(\Delta x^4).$$

Putting these approximations into (2.10) gives

$$\begin{aligned} & -a\Delta t\left[\delta_x^2 v_l^{j+1} - \frac{\Delta x^2}{12}(v_{xxxx})_l^{j+1}\right] \\ & -b\Delta t\left[\delta_x v_l^{j+1} - \frac{\Delta x^2}{6}(v_{xxx})_l^{j+1}\right] + [1+(r+\lambda)\Delta t]v_l^{j+1} \\ = \;& v_l^j + \lambda\Delta t(v*f)_l^j + \mathcal{O}(\Delta x^4). \end{aligned} \tag{3.12}$$

We remark here that $(v_{xxx})_l^{j+1}$ and $(v_{xxxx})_l^{j+1}$ are obtained by differentiating both sides of (2.10) with respect to x. Eventually, $(v_{xxx})_l^{j+1}$ and $(v_{xxxx})_l^{j+1}$ are expressed in terms of some first and second derivatives. To derive an FOC scheme, we only need to apply the central difference scheme to approximate all these derivatives. For the integral term, the approximations are written as

$$\delta_x(v*f)_l^j = \frac{(v*f)_{l+1}^j - (v*f)_{l-1}^j}{2\Delta x},$$

$$\delta_x^2(v*f)_l^j = \frac{(v*f)_{l+1}^j - 2(v*f)_l^j + (v*f)_{l-1}^j}{\Delta x^2},$$

and their computations will be discussed later in Section 4. After putting the approximations of $(v_{xxx})_l^{j+1}$ and $(v_{xxxx})_l^{j+1}$ into (3.12), we get a three point stencil of FOC difference scheme for $l = 1, 2, \ldots, L_0 - 1$,

$$\begin{aligned} & [1+(r+\lambda)\Delta t][(1-\alpha-\overline{\alpha})v_l^{j+1} + \alpha v_{l-1}^{j+1} + \overline{\alpha}v_{l+1}^{j+1}] \\ & +\Delta t[(\beta+\overline{\beta})v_l^{j+1} - \beta v_{l-1}^{j+1} - \overline{\beta}v_{l+1}^{j+1}] \\ = \;& (1-\alpha-\overline{\alpha})[v_l^j + \lambda\Delta t(v*f)_l^j] + \alpha[v_{l-1}^j + \lambda\Delta t(v*f)_{l-1}^j] \\ & +\overline{\alpha}[v_{l+1}^j + \lambda\Delta t(v*f)_{l+1}^j], \end{aligned} \tag{3.13}$$

where

$$\begin{cases} \alpha = \dfrac{1}{12} - \dfrac{b\Delta x}{24a}, & \overline{\alpha} = \dfrac{1}{12} + \dfrac{b\Delta x}{24a}, \\ \beta = \dfrac{b^2}{12a} + \dfrac{a}{\Delta x^2} - \dfrac{b}{2\Delta x}, & \overline{\beta} = \dfrac{b^2}{12a} + \dfrac{a}{\Delta x^2} + \dfrac{b}{2\Delta x}. \end{cases} \tag{3.14}$$

Consequently, according to (3.13), we can write the discretized equation in matrix form for $j = 0, 1, \ldots, J-1$:

$$M_1\bar{v}^{j+1} = M_2[\bar{v}^j + \lambda\Delta t(\bar{v}*f)^j] + \bar{v}_b^j, \tag{3.15}$$

where M_1 and M_2 are (L_0-1)-by-(L_0-1) tridiagonal matrices

$$\begin{aligned} M_1 = \quad & \text{tridiag}\,[[1+(\mathrm{r}+\lambda)\Delta\mathrm{t}]\alpha - \Delta\mathrm{t}\beta, \quad [1+(r+\lambda)\Delta t](1-\alpha-\overline{\alpha}) + \Delta t(\beta+\overline{\beta}), \\ & [1+(r+\lambda)\Delta t]\overline{\alpha} - \Delta t\overline{\beta}] \end{aligned}$$

and

$$M_2 = \text{tridiag}\left[\alpha,\ 1-\alpha-\overline{\alpha},\ \overline{\alpha}\right].$$

The vectors are given as

$$\bar{v}^j = [v_1^j, v_2^j, \ldots, v_{L_0-1}^j]^\intercal \quad \text{and} \quad (\bar{v} * f)^j = [(v*f)_1^j, (v*f)_2^j, \ldots, (v*f)_{L_0-1}^j]^\intercal.$$

Note that the time-dependent vector $\bar{v}_b^j$ contains the related values of the boundary conditions for European call options, or simply equals to a zero vector for double barrier call options. At each time step, a tridiagonal system (3.15) can be solved easily. After solving J linear systems, the resulting $\bar{v}^J$ is regarded as the approximation to the true solution.

We now proceed to demonstrate the discretization of the convolution integral. The algorithms for computing the integral will be given in Section 4. To handle the non-local y-domain of the convolution integral, we apply the method used by Almendral and Oosterlee [1], which does not require truncating the infinite integral domain. Instead we split the y-domain into $[x_1, x_{L_0-1}]$ and $[x_1, x_{L_0-1}]^c = [-\infty, x_1] \cup [x_{L_0-1}, \infty]$. Hence the integral becomes

$$\int_{x_1}^{x_{L_0-1}} v(y,t_j) f(y-x_l)dy + \int_{[x_1,x_{L_0-1}]^c} v(y,t_j) f(y-x_l)dy. \tag{3.16}$$

For the first integral in (3.16), we discretize the integral domain $[x_1, x_{L_0-1}]$ by Ω_x and write the integral in discrete form by the fourth order accurate composite Simpson's rule. Let ϕ_l^j denote the approximation of the first integral, then ϕ_l^j is given by

$$\phi_l^j = \sum_{k=1}^{L_0-1} w_k v_k^j f(x_k - x_l)\Delta x + \mathcal{O}(\Delta x^4), \tag{3.17}$$

where

$$[w_1, \ldots, w_{L_0-1}] = \overbrace{\left[\frac{1}{3}, \frac{4}{3}, \frac{2}{3}, \frac{4}{3}, \frac{2}{3}, \ldots, \frac{1}{3}\right]}^{L_0-1}.$$

For the second integral in (3.16), it needs to be computed on the non-local domain $[x_1, x_{L_0-1}]^c$. For European call options, we already know that $v(y,t_j)$ can be approximated by asymptotic boundary conditions:

$$v(y,t_j) \approx \begin{cases} 0, & \text{if } y \to -\infty, \\ Ke^y - Ke^{-rt_j}, & \text{if } y \to \infty. \end{cases}$$

We let ψ_l^j denote the approximation of the second integral. Then ψ_l^j is given by

$$\begin{aligned} \psi_l^j &= \int_{x_{L_0-1}}^{\infty} (Ke^y - Ke^{-rt_j}) f(y-x_l)dy \\ &= Ke^{x_l+\mu+\gamma^2/2}\mathbb{N}\left(\frac{x_l - x_{L_0-1} + \mu + \gamma^2}{\gamma}\right) - Ke^{-rt_j}\mathbb{N}\left(\frac{x_l - x_{L_0-1} + \mu}{\gamma}\right), \end{aligned}$$

where $\mathbb{N}$ is the standard normal cumulative distribution function. After all, the convolution integral is approximated by

$$(v * f)_l^j = \phi_l^j + \psi_l^j.$$

We remark that for double barrier call options with homogeneous boundary conditions, the splitting is better selected as $[x_{\min}, x_{\max}]$ and $[x_{\min}, x_{\max}]^c = [-\infty, x_{\min}] \cup [x_{\max}, \infty]$. Thus the second integral, which is located on $[x_{\min}, x_{\max}]^c$, should be equal to zero.

3.2 Stability analysis

In order to achieve high order accuracy for the time direction as discussed in Section 2.2, we study the unconditional stability of the FOC scheme (3.13) with the IMEX method in the following. We first introduce the following lemma.

Lemma 1 (see [16]) *For a two-level finite difference method which has the form*

$$\hat{U}^{j+1}(\xi) = \omega(\xi)\hat{U}^j(\xi),$$

where $\hat{U}^j$ is the Fourier transform of the grid function U^j at time level t_j, and $\omega(\xi)$ is called the amplification factor of the method at wave number ξ. If $\omega(\xi)$ satisfies

$$|\omega(\xi)| \leq 1 + c\Delta t,$$

where c is independent of ξ, then it follows that the method is stable.

Note that the PIDE (2.2) has constant coefficients and Dirichlet boundary conditions. Also the FOC scheme with IMEX method (3.13) is a two-level finite difference method. In the following, we will adopt Lemma 1 and the von Neumann stability in the proof. We refer the readers to see the details in [14], which applies the same method for stability analysis.

Theorem 1 *Suppose the constants $\sigma > 0$, $r \geq 0$, and $\lambda > 0$. The convolution integral in* (3.13) *is split up into two parts, where the first part is discretized by the fourth order accurate composite Simpson's rule* (3.17) *and the second part is explicitly given. Then the FOC scheme* (3.13) *is unconditionally stable.*

Proof: Let $E^j = [E_0^j, E_1^j, \ldots, E_{L_0}^j]^\intercal$ be the perturbation error vector. Note that $E_0^j = 0$ and $E_{L_0}^j = 0$ since boundary conditions are imposed. To apply the Fourier transform, we assume that the boundary conditions can be replaced by periodicity conditions. For the convolution integral, the splitting method is used and the second integral can be exactly computed. Then we define the inverse discrete Fourier transform with a special scaling factor as follows [14]:

$$E_l^j = \frac{1}{X} \sum_{k=1}^{L_0-1} (G_k)^j \exp(i\Lambda_k l), \tag{3.18}$$

and

$$f_l = \frac{1}{X}\sum_{m=1}^{L_0-1} F_m \exp(i\Lambda_m l), \tag{3.19}$$

where $f_l = f(x_l)$, $i \equiv \sqrt{-1}$, $\Lambda_k = 2\pi k/(L_0-1)$ is the wave number, $(G_k)^j$ and F_m are the discrete Fourier coefficients of E^j and f respectively, and X is the width of $[x_1, x_{L_0-1}]$ along the x-axis. Here we clarify that the notation $(G_k)^j$ means the jth power of G_k. From Lemma 1, we aim to verify that G_k, the Fourier coefficient of E^j, or the amplification factor, satisfies the following property:

$$|G_k| < 1 + c\Delta t, \quad k \in [1, 2, \ldots, L_0 - 1],$$

where the constant c should be independent of k. From the composite Simpson's rule, the discrete convolution satisfies

$$(E^j * f)_l = \frac{X}{L_0 - 2}\sum_{n=1}^{L_0-1} w_n E_n^j f_{n-l}, \tag{3.20}$$

which is fourth order accurate. Substituting (3.18) and (3.19) into (3.20), we have

$$\begin{aligned}(E^j * f)_l &= \frac{X}{L_0-2}\sum_{n=1}^{L_0-1}\frac{w_n}{X}\sum_{k=1}^{L_0-1}(G_k)^j\exp(i\Lambda_k n)\frac{1}{X}\sum_{m=1}^{L_0-1}F_m\exp(i\Lambda_m(n-l)) \\ &= \frac{1}{X}\sum_{k,m=1}^{L_0-1}(G_k)^j F_m\exp(-i\Lambda_m l)\sum_{n=1}^{L_0-1}\frac{w_n\exp(i\Lambda_k n)\exp(i\Lambda_m n)}{L_0-2}.\end{aligned}$$

Orthogonality leads to

$$\sum_{n=1}^{L_0-1}\frac{w_n\exp(i\Lambda_k n)\exp(i\Lambda_m n)}{L_0-2} = \begin{cases} 1, & \text{if } m=-k, \\ 0, & \text{otherwise.}\end{cases}$$

Ultimately

$$(E^j * f)_l = \frac{1}{X}\sum_{k=1}^{L_0-1}(G_k)^j F_{-k}\exp(i\Lambda_k l). \tag{3.21}$$

We can drop the summation to treat each Fourier component separately because of linearity. Substituting (3.18) and (3.21) into the FOC scheme (3.13) and eliminating the common factor $(G_k)^j\exp(i\Lambda_k l)$, we obtain

$$\begin{aligned}& G_k\left\{[1+(r+\lambda)\Delta t]\left[(1-\alpha-\overline{\alpha})+\alpha\exp(-i\Lambda_k)+\overline{\alpha}\exp(i\Lambda_k)\right]\right. \\ & \left.+\Delta t\left[(\beta+\overline{\beta})-\beta\exp(-i\Lambda_k)-\overline{\beta}\exp(i\Lambda_k)\right]\right\} \\ = \;& (1+\lambda\Delta t F_{-k})\left[(1-\alpha-\overline{\alpha})+\alpha\exp(-i\Lambda_k)+\overline{\alpha}\exp(i\Lambda_k)\right],\end{aligned}$$

or

$$G_k = \frac{1+\lambda\Delta t F_{-k}}{1+(r+\lambda)\Delta t+\Delta t\omega_0}, \tag{3.22}$$

where

$$\omega_0 = \frac{(\beta+\overline{\beta}) - \beta\exp(-i\Lambda_k) - \overline{\beta}\exp(i\Lambda_k)}{(1-\alpha-\overline{\alpha}) + \alpha\exp(-i\Lambda_k) + \overline{\alpha}\exp(i\Lambda_k)} \equiv \frac{\omega_1 + i\omega_3}{\omega_2 + i\omega_4}$$

and

$$\begin{cases} \omega_1 = (\beta+\overline{\beta})(1-\cos\Lambda_k), & \omega_2 = 1-(\alpha+\overline{\alpha})(1-\cos\Lambda_k), \\ \omega_3 = (\beta-\overline{\beta})\sin\Lambda_k, & \omega_4 = (\overline{\alpha}-\alpha)\sin\Lambda_k. \end{cases}$$

From the forward transform of f, we have

$$F_{-k} = \frac{X}{L_0-1}\sum_{m=1}^{L_0-1} f_m \exp(i\Lambda_m k).$$

Recall that $f \geq 0$ is the Gaussian density satisfying (2.5); i.e., the integral of f over the real line equals to 1. Note that f_m is the discrete values of f at grid points x_m. If we consider the numerical quadrature of f by using f_m, then the summation approximately tends to 1 but is not necessarily less than 1. Nevertheless, we can assume the following property as long as L_0 is reasonably large:

$$\frac{X}{L_0-1}\sum_{m=1}^{L_0-1} f_m \leq 2.$$

Thus

$$|F_{-k}| = \left|\frac{X}{L_0-1}\sum_{m=1}^{L_0-1} f_m \exp(i\Lambda_m k)\right| \leq \frac{X}{L_0-1}\sum_{m=1}^{L_0-1} f_m \leq 2.$$

We take the modulus on both sides of (3.22) to derive

$$|G_k| = \frac{|1+\lambda\Delta t F_{-k}|}{|1+(r+\lambda)\Delta t + \Delta t\omega_0|} \leq \frac{1+2\lambda\Delta t}{|1+(r+\lambda)\Delta t + \Delta t\omega_0|}.$$

Recall that $\lambda > 0$ and $r \geq 0$. Now we proceed to prove that ω_0 has a non-negative real part; i.e.,

$$\omega_1\omega_2 + \omega_3\omega_4 \geq 0.$$

By (3.14), the coefficients α, $\overline{\alpha}$, β and $\overline{\beta}$ satisfy

$$\begin{cases} \alpha+\overline{\alpha} = \dfrac{1}{6}, & \overline{\alpha}-\alpha = \dfrac{b\Delta x}{12a}, \\ \beta+\overline{\beta} = \dfrac{b^2}{6a} + \dfrac{2a}{\Delta x^2}, & \beta-\overline{\beta} = -\dfrac{b}{\Delta x}. \end{cases}$$

Hence direct manipulations give

$$\begin{aligned} & \omega_1\omega_2 + \omega_3\omega_4 \\ = \;& (1-\cos\Lambda_k)\left(\frac{b^2}{6a}+\frac{2a}{\Delta x^2}\right)\left(\frac{5}{6}+\frac{1}{6}\cos\Lambda_k\right) - \frac{b^2}{12a}\sin^2\Lambda_k \\ = \;& (1-\cos\Lambda_k)\left[\left(\frac{b^2}{6a}+\frac{2a}{\Delta x^2}\right)\left(\frac{5}{6}+\frac{1}{6}\cos\Lambda_k\right) - \frac{b^2}{12a}(1+\cos\Lambda_k)\right] \\ = \;& (1-\cos\Lambda_k)\left[\frac{b^2}{18a}(1-\cos\Lambda_k) + \frac{2a}{\Delta x^2}\left(\frac{5}{6}+\frac{1}{6}\cos\Lambda_k\right)\right]. \end{aligned}$$

According to (2.9) we have $a = \frac{1}{2}\sigma^2 > 0$, hence $\omega_1\omega_2 + \omega_3\omega_4 \geq 0$. Consequently

$$|G_k| \leq \frac{1 + 2\lambda\Delta t}{|1 + (r + \lambda)\Delta t + \Delta t\omega_0|} < 1 + 2\lambda\Delta t,$$

for any k. By Lemma 1, the FOC scheme (3.13) is unconditionally stable. □

3.3 Local mesh refinement

In the case of pricing a European call option or a double barrier call option, the financial payoff function is given by

$$v(x, 0) = \max(Ke^x - K, 0);$$

i.e., the non-smooth region is around the strike price $x^* = 0$. Thus it is necessary to concentrate grid points near x^*. Note that for European call options, the localization range $x_{\min}$ and $x_{\max}$ can always be selected appropriately such that x^* belongs to the uniform mesh. In this case we can concentrate grid points exactly around x^*. However, for double barrier options, the truncation is decided by the given lower and upper barriers and hence x^* might not be a grid node in an equally spaced computational domain. Therefore, we gather grid nodes around the grid point closest to x^*.

Grid points concentration is often accomplished by coordinate transformations [24]. In this paper, we apply the local mesh refinement strategy instead. One advantage of local mesh refinement over the coordinate transformation strategy is that it can be implemented adaptively [27]. Assume that the spatial direction is first discretized by a uniform mesh with initial mesh size

$$\Delta x_0 \equiv \Delta x = \frac{x_{\max} - x_{\min}}{L_0},$$

where $L_0 + 1$ is the beginning number of grid points in the x-direction. We then divide the initial mesh size Δx_0 by two at a time and add grid nodes with regard to the mesh size $\Delta x_h = \Delta x_0/2^h$, $h = 1, 2, \ldots$. Take the European call option case for example. Consider the four closest grid nodes to x^*:

$$x^* - 2\Delta x_0,\ x^* - \Delta x_0,\ x^*,\ x^* + \Delta x_0,\ x^* + 2\Delta x_0,$$

then we add four new nodes to obtain:

$$\begin{aligned} &x^* - 2\Delta x_0,\ \underline{x^* - 3\Delta x_1},\ x^* - \Delta x_0,\ \underline{x^* - \Delta x_1},\ x^*,\ \underline{x^* + \Delta x_1}, \\ &x^* + \Delta x_0,\ \underline{x^* + 3\Delta x_1},\ x^* + 2\Delta x_0. \end{aligned}$$

We repeat the steps until some h reaches the stopping criterion $\Delta x_h \leq \Delta x_0^2$. Eventually, there are $P_{\Delta x_0} = 4\lceil -\log_2 \Delta x_0 \rceil$ new nodes all together, where the operator $\lceil x \rceil$ rounds x to the nearest integer greater than or equal to x. For double barrier call options, we not only have to perform local mesh refinement near x^*, but also on boundary ends owing to discontinuity. Since the one-sided refinement is used for boundaries, the total number of

new nodes is $P_{\Delta x_0} = 8\lceil -\log_2 \Delta x_0 \rceil$. Note that the number $P_{\Delta x_0}$ is of order $\mathcal{O}(\log L_0)$ in both cases. Let $L = L_0 + P_{\Delta x_0}$, then $L+1$ is the total number of grid points after local mesh refinement. Denote the set of newly added grid points by Ω_{new}, then the refined computational domain is denoted by

$$\{\tilde{x}_0, \tilde{x}_1, \ldots, \tilde{x}_L\} = \Omega_x \cup \Omega_{new} \cup \{x_{\min}, x_{\max}\} \equiv \widetilde{\Omega}_x \cup \{x_{\min}, x_{\max}\},$$

where $\tilde{x}_0 = x_{\min}$ and $\tilde{x}_L = x_{\max}$. Note that every interior point $\tilde{x}_l \in \widetilde{\Omega}_x$ corresponds to a mesh size Δx_h, $h = 1, 2, \ldots$.

In Figure 1, we first sketch a uniformly discretized computational domain. The non-smooth region is assumed to be around the center point. Then we demonstrate what the discretized domain becomes after two steps of local mesh refinement strategy.

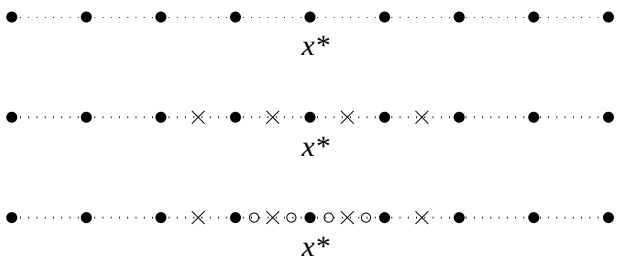

Figure 1: A discretized computational domain with local mesh refinement.

Since the computational grid has been refined, it is no longer uniform. Thus the standard FOC scheme can not be implemented directly. However, the grid points are added proportionally in our case. Hence the refined mesh grid is locally uniform and we can still apply the standard FOC scheme with certain treatment for the new grid points. Consequently, according to (3.13), we write the discretized equation in matrix form for $j = 0, 1, \ldots, J-1$:

$$\widetilde{M}_1 \tilde{v}^{j+1} = \widetilde{M}_2[\tilde{v}^j + \lambda \Delta t (\tilde{v} * f)^j] + \tilde{v}_b^j, \tag{3.23}$$

where $\tilde{v}^j$ is an approximation to

$$[v(\tilde{x}_1, t_j), v(\tilde{x}_2, t_j), \ldots, v(\tilde{x}_{L-1}, t_j)]^\intercal$$

and $(\tilde{v} * f)^j$ is an approximation to

$$[(v * f)(\tilde{x}_1, t_j), (v * f)(\tilde{x}_2, t_j), \ldots, (v * f)(\tilde{x}_{L-1}, t_j)]^\intercal.$$

As before, the time-dependent vector $\tilde{v}_b^j$ contains the relevant values of boundary conditions for European call options, or just equals to a zero vector for double barrier call options. Note that $\widetilde{M}_1$ and $\widetilde{M}_2$ are $(L-1)$-by-$(L-1)$ tridiagonal-like matrices after local mesh refinement because the three-point stencil of FOC scheme only involves three non-zero entries on each row.

Remark: The local mesh refinement strategy is regarded as a numerical remedy for the non-smoothness. The stability of using local mesh refinement is beyond the scope of this paper, yet numerical results will show its effect of recovering desired accuracy.

4 Evaluation of the integral term

Evaluation of the integral term is required to solve the linear system (3.23) at each time step. The numerical quadrature method is used for evaluating the integral term in our case.

4.1 Evaluation of the integral term with uniform mesh

We first consider the original system (3.15) where the mesh grid is uniform and evaluate the following convolution integral

$$\int_{-\infty}^{\infty} v(y,t_j)f(y-x_l)dy$$

to obtain $(\bar{v}*f)^j = [(v*f)_1^j, (v*f)_2^j, \ldots, (v*f)_{L_0-1}^j]^\intercal$. As mentioned before, the splitting method is used and the first integral on $[x_1, x_{L_0-1}]$ can be computed by the composite Simpson's rule (3.17). Furthermore, the right-hand side of (3.17) can be written in matrix-vector multiplication form $A\hat{v}^j$ for $l,k = 1,2,\ldots,L_0-1$, where

$$[A]_{l,k} = f(x_k - x_l)$$

and

$$\hat{v}^j = \frac{\Delta x}{3}[v_1^j, 4v_2^j, 2v_3^j, 4v_4^j, 2v_5^j, \ldots, v_{L_0-1}^j]^\intercal.$$

We now reduce the evaluation of integral to a matrix-vector product $A\hat{v}^j$. Note that A is an (L_0-1)-by-(L_0-1) Toeplitz matrix. It is well known that the matrix-vector multiplication $A\hat{v}^j$ can be computed by FFTs. We first embed A into a $(2L_0-2)$-by-$(2L_0-2)$ circulant matrix

$$C = \begin{bmatrix} A & \times \\ \times & A \end{bmatrix}.$$

After diagonalizing C by the Fourier matrix and forming a matrix-vector product as follows:

$$\begin{bmatrix} A & \times \\ \times & A \end{bmatrix} \begin{bmatrix} \hat{v}^j \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} A\hat{v}^j \\ \dagger \end{bmatrix}, \tag{4.24}$$

we obtain $A\hat{v}^j$ in three FFTs with $\mathcal{O}(L_0 \log L_0)$ complexity [6, 7].

Combining the second integral, which has an explicit evaluation formula, we obtain

$$(\bar{v}*f)^j = \Phi^j + \Psi^j = A\hat{v}^j + \Psi^j,$$

where Φ^j and Ψ^j correspond respectively to the vector forms of ϕ_l^j and ψ_l^j for $l = 1,2,\ldots,L_0-1$. For double barrier call options, the algorithm for computing the integral is fundamentally the same as the one for European options.

4.2 Evaluation of the integral term with local mesh refinement

After applying the local mesh refinement strategy, the evaluation of the integral term also needs certain modifications. We still apply the previous splitting method, and the second outer integral is computed in the same way. The first local integral is what we need to pay attention to. For the computational y-grid, we continue using the uniform grid Ω_x with mesh size Δx. As before, the integral for $l = 1, 2, \ldots, L-1$:

$$\int_{x_1}^{x_{L_0-1}} v(y, t_j) f(y - \tilde{x}_l) dy$$

can be approximated by the fourth order composite Simpson's rule. Furthermore, it can also be written in a matrix-vector product $\widetilde{A}\hat{v}^j$ for $l = 1, 2, \ldots, L-1$, $k = 1, 2, \ldots, L_0 - 1$, where

$$[\widetilde{A}]_{l,k} = f(x_k - \tilde{x}_l), \quad x_k \in \Omega_x, \quad \tilde{x}_l \in \widetilde{\Omega}_x,$$

and the vector $\hat{v}^j$ is the same as before. We remark that the $(L-1)$-by-(L_0-1) matrix $\widetilde{A}$ no longer yields the Toeplitz structure. Hence $\widetilde{A}\hat{v}^j$ can not be computed directly by Toeplitz matrix-vector multiplication (4.24). However, we notice that $\widetilde{A}$ is a Toeplitz-like matrix which only contains a few irregular rows regarding the new grid nodes $\tilde{x}_l \in \Omega_{new}$. To exploit the major Toeplitz structure, we evaluate the integral values at $\tilde{x}_l \in \Omega_x$ and $\tilde{x}_l \in \Omega_{new}$ separately.

First we neglect the refined grid nodes and eliminate their corresponding rows, then $\widetilde{A}$ is reduced to the (L_0-1)-by-(L_0-1) Toeplitz matrix A, which is the one in the uniform mesh case. Therefore, the integral values at $\tilde{x}_l \in \Omega_x$ can be obtained by (4.24), which is of order $\mathcal{O}(L_0 \log L_0)$. To derive the remaining integral values at the refined grid nodes $\tilde{x}_l \in \Omega_{new}$, we simply carry out a direct matrix-vector multiplication. Since the number of refined grid nodes is of order $\mathcal{O}(\log L_0)$, and the vector $\hat{v}^j$ being multiplied is of length $\mathcal{O}(L_0)$, the cost of such multiplication is of order $\mathcal{O}(L_0 \log L_0)$, which is at the same cost level as (4.24). That means totally speaking, the integral term for all $\tilde{x}_l \in \widetilde{\Omega}_x$ can be evaluated with $\mathcal{O}(L_0 \log L_0)$ complexity.

5 Numerical results

5.1 European call options

In this section, we give the numerical results of pricing European call options under Merton's jump-diffusion model. The analytical formula of pricing the European call options for zero dividend has been found [18] for Merton's model:

$$w(S, \tau) = \sum_{k=0}^{\infty} \frac{e^{-\lambda(1+\kappa)\tau} [\lambda(1+\kappa)\tau]^k}{k!} w_{BS}(S, K, \tau, \sigma_k, r_k),$$

where $\tau = T - t$ is the time, $\kappa = \mathbb{E}(\eta - 1) = e^{(\mu + \gamma^2/2)} - 1$ denotes the expectation of the impulse function and w_{BS} is the Black-Scholes formula given by

$$w_{BS}(S, K, \tau, \sigma_k, r_k) = S\mathbb{N}(d_1) - Ke^{-r\tau}\mathbb{N}(d_2),$$

$$\sigma_k^2 = \sigma^2 + \frac{k\gamma^2}{\tau}, \quad r_k = r - \lambda\kappa + \frac{k\log(1+\kappa)}{\tau},$$
$$d_1 = \frac{\log(S/K) + (r + \sigma^2/2)\tau}{\sigma\sqrt{\tau}} \quad \text{and} \quad d_2 = d_1 - \sigma\sqrt{\tau}.$$

Here S is the underlying stock price, $\mathbb{N}$ is the standard normal cumulative distribution function. Note that this analytical solution relies greatly on the Black-Scholes formula. Since the Black-Scholes formula is well known for pricing European call options, we can regard this analytical expression as the exact solution.

We also compare a second order central difference discretization scheme with the FOC scheme (3.13). The input parameters are $T = 0.25$, $K = 100$, $\sigma = 0.25$, $r = 0.05$, $\delta = 0$, $\lambda = 0.1$, $\mu = -0.9$ and $\gamma = 0.45$. For truncating the infinite spatial domain to $[x_{\min}, x_{\max}]$, we select $x_{\min} = -1.4$ and $x_{\max} = 5.0$. By Theorem 1, the unconditional stability allows us to achieve high order accuracy for the time direction by extrapolation approach. According to Richardson extrapolation approach, we can derive an approximation at the sth extrapolation stage. As long as s is taken sufficiently large, the convergence properties of the spatial algorithm will not be influenced. Therefore, we first divide the time interval into $J = 25$ time steps and assume $s = 6$. The value at the sixth extrapolation stage in (2.7), denoted by $v_{\Delta x}$, is accepted as the approximation at maturity time T using spatial mesh size Δx.

Recall that $L_0 + 1$ is the number of grid points in the x-direction. Let "Error at x^*" be the difference between the true solution and the approximation at $x^* = 0$. The "l^∞ error" is the infinity norm error between the true solution and the approximation. Moreover, the column "Order" shows the convergence order defined by

$$\text{Order} = \log_2 \frac{\|v_{true} - v_{\Delta x}\|_\infty}{\|v_{true} - v_{\Delta x/2}\|_\infty},$$

where v_{true} is the true solution at corresponding grid points at maturity time T. Theoretically speaking, the convergence order equals to two for quadratic convergence and four for quartic convergence. Table 1 shows that a standard central difference scheme clearly attains second order accuracy.

However, we notice that the FOC scheme using uniform grid only achieves second order convergence in Table 2. Specifically, we notice from Figure 2 that the grid nodes with larger errors are the ones around x^*. It is due to the non-smoothness of the financial payoff function. Therefore, the local mesh refinement should be applied to restore the fourth order convergence.

In Table 3, $L_0 + 1$ is the beginning number of grid points in the x-direction, Δx_0 is the initial mesh size, $P_{\Delta x_0} = 4\lceil -\log_2 \Delta x_0 \rceil$ is the number of new nodes we have to add, and $L + 1$ is the number of grid points in the x-direction after local mesh refinement. We see that the FOC scheme with local mesh refinement restores fourth order convergence. Figure 2 shows the effect of local mesh refinement by drawing the error distribution at time T. Before using the local mesh refinement, a mesh grid with $L_0 = 256$ obtains errors around 10^{-2}, while the refined mesh grid significantly lowers the errors to 10^{-5}. The option value at $t = T$ and the payoff function are displayed in Figure 3.

Table 1: Errors and convergence orders of central difference scheme for pricing a European call option under Merton's jump-diffusion model.

L_0	Δx	Error at x^*	l^∞ error	Order
32	1/5	1.62e+0	1.62e+0	-
64	1/10	4.51e-1	4.51e-1	1.847
128	1/20	9.76e-2	9.76e-2	2.207
256	1/40	2.38e-2	2.38e-2	2.038
512	1/80	5.91e-3	5.91e-3	2.001
1024	1/160	1.47e-3	1.47e-3	2.002

Table 2: Errors and convergence orders of FOC scheme with uniform grid for pricing a European call option under Merton's jump-diffusion model.

L_0	Δx	Error at x^*	l^∞ error	Order
32	1/5	1.21e+0	1.21e+0	-
64	1/10	2.84e-1	2.84e-1	2.094
128	1/20	6.48e-2	6.48e-2	2.131
256	1/40	1.59e-2	1.61e-2	2.010
512	1/80	3.97e-3	4.01e-3	2.004
1024	1/160	9.91e-4	1.00e-3	2.000

Table 3: Errors and convergence orders of FOC scheme with local mesh refinement for pricing a European call option under Merton's jump-diffusion model.

L_0	Δx_0	$P_{\Delta x_0}$	L	Error at x^*	l^∞ error	Order
32	1/5	12	44	1.69e-2	2.20e-2	-
64	1/10	16	80	1.82e-3	4.67e-3	2.234
128	1/20	20	148	1.16e-4	3.87e-4	3.593
256	1/40	24	280	5.49e-6	2.77e-5	3.802
512	1/80	28	540	6.61e-7	1.88e-6	3.883
1024	1/160	32	1056	4.66e-8	1.23e-7	3.935

5.2 Double barrier call options

In this section, we give the numerical results of pricing double barrier call options under Merton's jump-diffusion model. The input parameters are given by $x_{\min} = x_d = \log(0.8)$, $x_{\max} = x_u = \log(1.2)$, $T = 1$, $K = 100$, $\sigma = 0.1$, $r = 0.05$, $\delta = 0.02$, $\lambda = 3$, $\mu = -0.05$ and $\gamma = 0.086$, which are also used by Feng and Linetsky [11]. We still divide the time interval

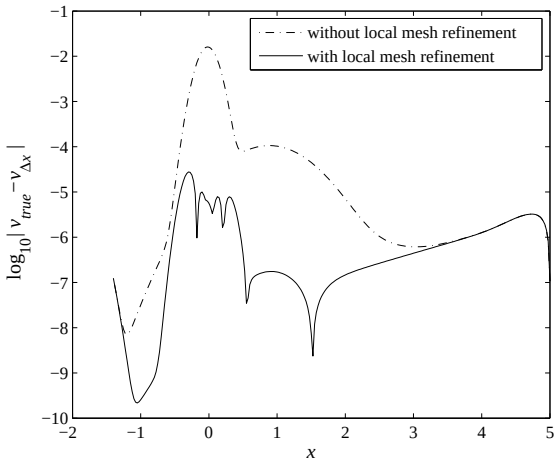

Figure 2: Error distribution of FOC scheme for pricing European call options under Merton's jump diffusion model with $L_0 = 256$ at time T.

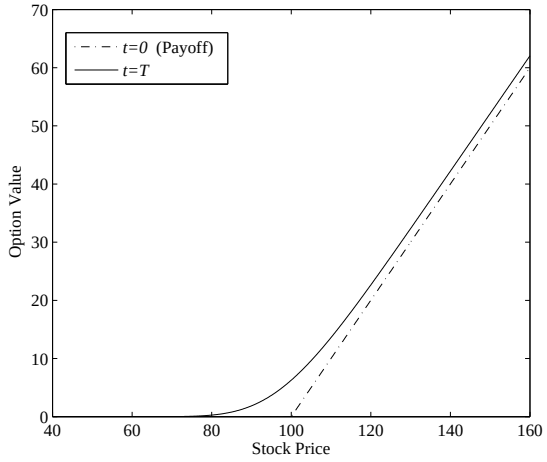

Figure 3: Option value at $t = 0$ and $t = T$ with $L_0 = 1024$ for pricing European call options under Merton's jump diffusion model.

into $J = 25$ time steps and set the extrapolation stage number as $s = 6$ such that the spatial convergence is not influenced. Since there is no explicit formula for pricing double barrier call options, we measure the l^∞ error between any level and its succeeding level. Also the convergence order is redefined by the $\log 2$-ratio of errors between neighboring

levels:

$$\text{Order} = \log_2 \frac{\|v_{\Delta x} - v_{\Delta x/2}\|_\infty}{\|v_{\Delta x/2} - v_{\Delta x/4}\|_\infty}.$$

We also show the reference option price $v(x^*, T)$, in comparison with the option value obtained by Feng and Linetsky. As before $L_0 + 1$ is the initial number of grid points in the spatial direction, $P_{\Delta x_0}$ is the number of newly added nodes, and $L + 1$ is the number of spatial grid points after local mesh refinement. From Table 4 we can still observe the fourth order convergence when the FOC scheme with local mesh refinement is applied to price double barrier call options.

Table 4: Errors and convergence orders of FOC scheme with local mesh refinement for pricing a double barrier call option under Merton's jump-diffusion model.

L_0	$P_{\Delta x_0}$	L	$v(x^*, T)$	l^∞ error	Order
20	48	68	1.96465512	1.29e-4	-
40	56	96	1.96473605	7.60e-6	4.087
80	64	144	1.96472899	5.15e-7	3.882
160	72	232	1.96472852	3.32e-8	3.955
320	80	400	1.96472849	1.86e-9	4.162
640	88	728	1.96472849	1.26e-10	3.883
1280	96	1376	1.96472849	-	-

6 Concluding remarks

In this paper we have applied an FOC scheme for solving a PIDE, which arises from the option pricing problem in Merton's jump-diffusion model. We remark here that the exponential variable change in the PIDE is only for simplicity. The FOC scheme also applies to a convection-diffusion equation with variable coefficients and the resulting system can still be easily solved [13, 21].

It is well known that the option values of American options have discontinuities in their second derivatives at the early exercise boundary, and also a non-smoothness occurs in the time direction. To achieve higher than second order accuracy, the current spatial refinement strategy and extrapolated IMEX scheme need to be revised. For an example, the use of a moving space-time mesh, or the adaptive extrapolation schemes mentioned in [11] will be considered for American option pricing in jump-diffusion models. In addition, other advanced jump-diffusion models including the variance gamma model and the CGMY model will be taken into account. We will also investigate financial models which involve stochastic volatility. With the advent of stochastic volatility, we have to deal with an additional variable in a high-dimensional case.

Acknowledgements. The authors would like to thank Professor Deng Ding for introducing the financial model to us and Jun Liu for suggestions on stability proof and numerical experiments. The authors are also grateful to the anonymous referees for constructive comments which benefit this paper a lot.

References

[1] A. Almendral and C. Oosterlee, Numerical valuation of options with jumps in the underlying, Appl Numer Math 53 (2005), 1–18.

[2] K. Amin, Jump diffusion option valuation in discrete time, J Finance 48 (1993), 1833–1863.

[3] L. Andersen and J. Andreasen, Jump-diffusion processes: Volatility smile fitting and numerical methods for option pricing, Rev Deriv Res 4 (2000), 231–262.

[4] D. Bates, Jumps and stochastic volatility: Exchange rate processes implicit in Deutsche Mark options, Rev Financ Stud 9 (1996), 69–107.

[5] F. Black and M. Scholes, The pricing of options and corporate liabilities, J Polit Economy 81 (1973), 637–654.

[6] R. Chan and X. Jin, An Introduction to Iterative Toeplitz Solvers, SIAM, Philadelphia, 2007.

[7] R. Chan and M. Ng, Conjugate gradient methods for Toeplitz systems, SIAM Rev 38 (1996), 427–482.

[8] R. Cont and P. Tankov, Financial Modeling with Jump Processes, Chapman & Hall/CRC Press, 2004.

[9] R. Cont and E. Voltchkova, A finite difference scheme for option pricing in jump diffusion and exponential Lévy models, SIAM J Numer Anal 43 (2005), 1596–1626.

[10] D. Duffie, J. Pan and K. Singleton, Transform analysis and asset pricing for affine jump-diffusions, Econometrica 68 (2000), 1343–1376.

[11] L. Feng and V. Linetsky, Pricing options in jump-diffusion models: An extrapolation approach, Oper Res 56 (2008), 304–325.

[12] M. Gupta, A fourth-order Poisson solver, J Comput Phys 55 (1984), 166–172.

[13] M. Gupta, R. Manohar and J. Stephenson, A single cell high order scheme for the convection-diffusion equation with variable coefficients, Internat J Numer Methods Fluids 4 (1985), 71–80.

[14] Y. d'Halluin, P. Forsyth and K. Vetzal, Robust numerical methods for contingent claims under jump diffusion processes, IMA J Numer Anal 25 (2005), 87–112.

[15] S. Heston, A closed-form solution for options with stochastic volatility with applications to bond and currency options, Rev Financ Stud 6 (1993), 327–343.

[16] R. LeVeque, Finite Difference Methods for Ordinary and Partial Differential Equations: Steadt-State and Time-Dependent Problems, SIAM, Philadelphia, 2007.

[17] S. Kou, A jump-diffusion model for option pricing, Manag Sci 48 (2002), 1086–1101.

[18] R. Merton, Option pricing when underlying stock returns are discontinuous, J Financ Eco 3 (1976), 125–144.

[19] N. Rambeerich, D. Tangman, A. Gopaul and M. Bhuruth, Exponential time integration for fast finite element solutions of some financial engineering problems, J Comput Appl Math 224 (2009), 668–678.

[20] R. Seydel, Tools for Computational Finance, Springer, Berlin, 2002.

[21] W. Spotz, High-Order Compact Finite Difference Schemes for Computational Mechanics, PhD Thesis, University of Texas at Austin, 1995.

[22] W. Spotz and G. Carey, High-order compact scheme for the steady stream-function vorticity equations, Internat J Numer Methods Engrg 38 (1995), 3497–3512.

[23] D. Tangman, A. Gopaul and M. Bhuruth, Numerical pricing of options using high-order compact finite difference schemes, J Comput Appl Math 218 (2008), 270–280.

[24] D. Tavella and C. Randall, Pricing Financial Instruments: The Finite Difference Method, John Wiley and Sons, New York, 2000.

[25] J. Zhang, An explicit fourth-order compact finite difference scheme for three dimensional convection-diffusion equation, Commun Numer Methods Engrg 14 (1998), 263–280.

[26] J. Zhang, L. Ge and M. Gupta, Fourth order compact scheme for 3D convection diffusion equation with boundary layers on nonuniform grids, Neural Parallel Sci Comput 8 (2000), 373–392.

[27] J. Zhang, H. Sun and J. Zhao, High order compact scheme with multigrid local mesh refinement procedure for convection diffusion problems, Comput Methods Appl Mech Engrg 191 (2002), 4661–4674.

[28] X. Zhang, Numerical analysis of American option pricing in a jump-diffusion model, Math Oper Res 22 (1997), 668–690.