I
Vol. 00, No. 00, Month 200x, 1–26

Options Pricing under the One-Dimensional Jump-diffusion Model using the Radial Basis Function Interpolation Scheme

Tat Lung (Ron) Chan[a*] and Simon Hubbert[b′]

[a] *UEL Royal Docks Business School, University of East London, Docklands Campus, 4-6 University Way, London E16 2RD*;
[b] *Department of Economics, Mathematics and Statistics, Birkbeck, University of London, Malet Street, London WC1E 7HX.*

(*Received 00 Month 200x; in final form 00 Month 200x*)

This paper will demonstrate how European and American option prices can be computed under the jump-diffusion model using the radial basis function (RBF)interpolation scheme. The RBF interpolation scheme is demonstrated by solving an option pricing formula, a one-dimensional partial integro-differential equation (PIDE). We select the cubic spline radial basis function and propose a simple numerical algorithm to establish a finite computational range for the improper integral of the PIDE. This algorithm can improve the approximation accuracy of the integral with the application of any quadrature. Moreover, we offer a numerical technique termed cubic spline factorisation to solve the inversion of an ill-conditioned RBF interpolant, which is a well-known research problem in the RBF field. Finally, we numerically prove that in the European case, our RBF-interpolation solution is second-order accurate for spatial variables, while in the American case, it is second-order accurate for spatial variables and first-order accurate for time variables.

Keywords: European options, American options, jump-diffusion models, radial basis functions, cubic spline

1. Introduction

In this paper, we demonstrate the computation of the European and American option prices using the jump-diffusion model and radial basis function (RBF) interpolation techniques. The RBF methods were recently proposed to numerically solve initial value and free-boundary problems for the classical Black and Scholes equation in either one- or multiple-asset cases [23, 24, 29, 36]. In this paper, for the jump-diffusion model, as with other Lévy-type models, the Black and Scholes PDE is replaced by a partial integro-differential operator (PIDE), which involves a global term in the form of an integral operator. The PIDE has the following form [cf. 16, 48, 49]:

$$\partial_\tau u(x,\tau) = \frac{1}{2}\sigma^2\partial_x^2 u + \left(r - q - \frac{1}{2}\sigma^2 - \eta\right)\partial_x u - (r+\lambda)u + \lambda\int_{\mathbb{R}} u(x+y,\tau)f(y)dy \tag{1}$$

*Corresponding author. Email: t.l.chan@uel.ac.uk

′Corresponding author. Email: s.hubbert@ems.bbk.ac.uk

ISSN: 0020-7160 print/ISSN 1029-0265 online

DOI: 10.1080/0020716YYxxxxxxxx
http://www.informaworld.com

Our principal contribution is to demonstrate an efficient numerical solution (1) using RBFs, both for initial value and free-boundary problems (as with the American options). We have chosen the jump-diffusion model as a typical case on which to test this RBF methodology. However, our method can be easily extended to other contexts in which the basic pricing equation is a PIDE, such as the Carr-Geman-Madan-Yor (CGMY) [15] or variance Gamma (VG) [13, 41] Lévy-type models. These models will be addressed in another paper [11].

PIDEs, such as the Merton [44] and Kou Models [34, 35], are typically treated using a traditional finite difference method (FDM) or finite element method (FEM). In an FDM, the PIDE is fully discretised on an equidistant grid after having (artificially) localised the equations to some bounded interval/domain in $\mathbb{R}$. The global integral term can be computed using numerical quadrature or fast Fourier transform (FFT) [see, 1–6, 9, 17–19, 28, 52]. In contrast, the FEM is defined by piecewise polynomial functions or wavelet functions on regular triangularisations. This technique is used to approximate solutions of the partial differential terms and integral term [cf. 2, 42, 43].

Our approach not only provides a new research direction for applying RBFs in option pricing but also resolves problems that arise from using FDMs and RBFs. We summarise the problems as follows:

(1) In FDM, the Crank-Nicolson scheme is one of the most popular methods for time discretisation; however, with a short time to maturity, it produces wiggles in both the option price and its sensitivities near the strike price (at which the first-order differentiation is discontinuous). Giles and Carter shed light on this problem [26] by suggesting Rannacher's time stepping method (a mixture of four half-timesteps of the backward Euler and Crank-Nicolson methods). Although Giles and Carter provide proof of their methods and resolve the problem, they limit their methods to pricing European options under the Black-Scholes model and do not extend their ideas to solve (1) or American-type options.
(2) Recently, the RBF-interpolation scheme using a multiquadric (MQ) basis function was proposed to numerically solve the classical Black and Scholes PDE [cf. 23, 24, 29, 36] because of its comparatively higher accuracy. The MQ contains a shape parameter, which plays a critical role in the accuracy of the interpolation [cf. 53]. Unfortunately, no theoretical proof for selecting an optimal shape parameter [cf. 53] in the MQ basis function has emerged to date.
(3) The standard approach to solving the radial basis function interpolation problem has been recognised as ill conditioned for many years [cf. 21, chapter 16], particularly when infinitely smooth basic functions, such as the MQ and Gaussian functions, are used with small values for their associated shape parameters. Recent research papers [e.g. 20, 22, 25, 37] have suggested different numerical techniques for solving the RBF ill-conditioning problem, but these techniques are restricted to solving the simple interpolation problem and cannot solve PDEs. Although Ling and his co-workers [e.g. 10, 39, 40] address the ill-conditioning problem using preconditioning methods and extend their work to solve PDEs, the methods are not applicable to PIDEs.

Our RBFan approximation method using a cubic spline as a basis function will circumvent these disadvantages. This paper is divided into five sections, including this introduction. Section 2 provides a brief review of both the Merton and Kou jump-diffusion models. In Section 3, we first explain and then define our RBF algorithm for solving PIDEs, which we then implement in the jump-diffusion model. Section 4 contains our numerical results for both the European and American call and put options, including an analysis of the maximum error, root-mean-square error, rate of convergence and approximation of delta and gamma hedging formulas and a comparison of the accuracy of our solution with that of the FDM and FEM. Section 5 concludes the paper.

2. PIDE Option Pricing Formula in the Jump-diffusion Market

A jump-diffusion process has two main building blocks, a Brownian process and a compound Poisson process. We use a Brownian process $(W_t)_{t\geq 0}$ to describe the evolution of a risky asset $(S_t)_{t\geq 0}$ and a compound Poisson process $(N_t)_{t\geq 0}$ to describe the jumps occurring in $(S_t)_{t\geq 0}$. In the model, jumps represent rare events, such as crashes and/or drawdowns, at random intervals in $(S_t)_{t\geq 0}$. To ensure positivity and independent and stationary log-returns of the asset [cf. 16], S_t is typically modelled as an exponential jump-diffusion process:

$$S_t = S_0 e^{L_t} \tag{2}$$

where S_0 is the asset price at time zero and L_t is defined as follows:

$$L_t := \gamma_c t + \sigma W_t + \sum_{i=1}^{N_t} Y_i \tag{3}$$

where γ_c is a risk-neutral drift term, σ is the volatility, W_t represents the Brownian motion, N_t is the Poisson process with an intensity λ, and Y_i is an i.i.d. sequence of random variables. Moreover, the characteristic function of this process can be considered a special case of the Lévy-Khintchine formula [16]:

$$\mathbb{E}\left[e^{iuL_t}\right] = \exp\left(t\left(iu\gamma_c - \frac{\sigma^2 u^2}{2} + \lambda\int_{\mathbb{R}}(e^{iux}-1)f(x)dx\right)\right) \tag{4}$$

where

$$\gamma_c = r - q - \frac{1}{2}\sigma^2 - \lambda\eta. \tag{5}$$

In this equation, r is the risk-neutral interest rate, q represents the compounded dividends and η is a constant equal to $\int_{\mathbb{R}}(e^x - 1)f(x)dx$. The value of η is determined by $f(x)$, the probability density function of Y_i given in (3). In the classical Merton model [44], for any $i \in \{1, 2, \ldots\}$, Y_i represents log-normally distributed variables with $Y_i \sim \mathbb{N}(\mu_J, \sigma_J^2)$,

$$f(x) := \frac{1}{\sqrt{2\pi}\sigma_j} e^{(x-\mu_J)^2/2\sigma_J^2}, \tag{6}$$

and

$$\eta = e^{\mu_J + \sigma_J^2/2} - 1 \tag{7}$$

(for the details on (7), we refer the reader to [8, 16]). If we replace $f(x)$ with exponential density functions defined by

$$f(x) := p\alpha_1 e^{-\alpha_1 x}\mathbb{1}_{x\geq 0} + (1-p)\alpha_2 e^{\alpha_2 x}\mathbb{1}_{x\leq 0}, \tag{8}$$

we obtain a new value for η:

$$\eta = \frac{p\alpha_1}{\alpha_1 - 1} + \frac{(1-p)\alpha_2}{\alpha_2 + 1} - 1. \tag{9}$$

With this new $f(x)$, the model is called the Kou model [34, 35]. The calculation of η is found by simply integrating e^x over the real line with $\alpha_1 > 1$ and $\alpha_2 > 0$ [cf. 16].

2.1 *European options*

Because $\sigma > 0$ in (3), a risk-neutral probability measure $\mathbb{Q}$ [cf. 47, Theorems 33.1 and 33.2] is required, and as a result, γ_c in (3) guarantees that the discounted process $e^{-(r-q)t}S_t$ is a martingale process. Based on the risk-neutral arguments and the fact that $e^{-(r-q)t}S_t$ is a martingale process (see, for example, [16]), we can derive the following PIDE pricing formula that describes the price of a European contingent claim $u(x,\tau)$ in $\log S_t = x$ over the time to maturity, $\tau = T - t$ [cf. 16]:

$$\begin{aligned} \partial_\tau u(x,\tau) &= \frac{1}{2}\sigma^2\partial_x^2 u + \left(r - q - \frac{1}{2}\sigma^2 - \eta^*\right)\partial_x u - (r+\lambda)u + \\ &\quad \lambda\int_{\mathbb{R}} u(x+y,\tau)f(y)dy, \\ &=: \mathcal{L}[u](x,\tau). \end{aligned} \tag{10}$$

with an initial value of

$$u(x,0) = g(x) := G(e^x) = \begin{cases} \max\{e^x - K, 0\}\,, \text{ call option} \\ \max\{K - e^x, 0\}\,, \text{ put option} \end{cases} \tag{11}$$

where $\eta^* = \eta + \int_{\mathbb{R}} xf(x)\,dx$ and K is the strike price.

2.2 *American options*

For an American put option, we must consider the possibility of early exercise [e.g., 16, 48, 49]. As a result, the highest value of an American option can be achieved by maximising over all allowed exercise strategies:

$$u(x,\tau) = \operatorname{ess\,sup}_{\tau^*\in\Gamma(\mathrm{t,T})}\mathrm{E_t^Q}\left[\mathrm{e}^{-(\mathrm{r-q})(\tau^*-\mathrm{t})}\mathrm{G}\big(\mathrm{e}^{\mathrm{x}_{\tau^*}}\big)\right] \tag{12}$$

where $\Gamma(t,T)$ denotes the set of non-anticipating exercise times τ^* that satisfy $t \leq \tau^* \leq T$. To compute the $u(x,\tau)$ for the American put option, one can solve the following linear complementarity problem [16, 48, 49]:

$$\partial_\tau u(\tau, x) - \mathcal{L}u(x, \tau) \geq 0, \text{ in } (0, T) \times \mathbb{R} \tag{13}$$

$$u(x, \tau) - G(e^x) \geq 0, \text{ a.e. in } (0, \mathrm{T}) \times \mathbb{R} \tag{14}$$

$$\big(u(x, \tau) - G(e^x)\big)\left(\partial_\tau u(\tau, x) - \mathcal{L}u(x, \tau)\right) = 0, \text{ in } (0, T) \times \mathbb{R} \tag{15}$$

$$u(x, 0) = G(e^x), \tag{16}$$

Because this paper only considers a jump-diffusion model with $\sigma > 0$ and a finite jump intensity, the smooth pasting condition

$$\frac{\partial u(x_{\tau^*}, \tau^*)}{\partial x} = -1$$

is valid at the time of exercise $\tau*$, as noted by Pham [46]. The value of an American put option is therefore continuously differentiable with respect to the underlying $(0, T) \times \mathbb{R}$; in particular, the derivative is continuous across the exercise boundary.

Remark 1 One should note that if we set $\lambda = 0$, (10) will become the original Black-Scholes PDE.

3. Mesh-free Numerical Approximation Method

The RBF interpolation scheme is a well-known meshless technique for reconstructing an unknown function from scattered data that has numerous applications in various fields, such as geological terrain modelling, surface reconstruction in imaging, and numerically solving partial differential equations in applied mathematics. In particular, RBFs have recently been used to solve PDEs in quantitative finance. Several authors, including Fausshauer *et al.* [23, 24], Larsson *et al.* [36], Pettersson *et al.* [45] and Hon and Mao [29], have suggested RBFs as a tool for solving Black-Scholes equations for European and American options. This numerical scheme for estimating partial derivatives using RBFs was originally proposed by Kansa [32] and resulted in a new method for solving partial differential equations [33].

To solve the PIDE, one must first obtain an RBF approximation of the initial value or pay-off of the option. Once we have a disposition of such an RBF interpolant, we can implement an RBF scheme to solve the PIDE using this RBF interpolant as the initial value. The general idea of the proposed numerical scheme is to approximate the unknown function $u(x, \tau)$ using an RBF interpolant with the RBF scheme to determine the interpolation points for the initial value and deriving a system for the linear constant coefficient ODE by requiring that the PIDE (10) be satisfied for the chosen RBF interpolation points.

After selecting the interpolation points $x_j \in \mathbb{R}$, we Approximate the solution $u(x, \tau)$ in (10) for any fixed time to maturity τ using its RBF interpolant as follows:

$$u(x, \tau) \simeq \sum_{j=1}^{N} \rho_j(\tau)\phi(||x - x_j||_2) =: \widetilde{u}(x, \tau). \tag{17}$$

Because the radial basis function is not time dependent, the time derivative of

$\widetilde{u}(x,\tau)$ in equation (10) is simply

$$\frac{\partial \widetilde{u}(x,\tau)}{\partial \tau} = \sum_{j=1}^{N} \frac{d\rho_j(\tau)}{d\tau}\phi(|x-x_j|). \tag{18}$$

Moreover, the first and second partial derivatives of $\widetilde{u}(x,\tau)$ with respect to x are as follows:

$$\frac{\partial \widetilde{u}(x,\tau)}{\partial x} = \sum_{j=1}^{N} \rho_j(\tau)\frac{\partial \phi(|x-x_j|)}{\partial x} and \tag{19}$$

$$\frac{\partial^2 \widetilde{u}(x,\tau)}{\partial x^2} = \sum_{j=1}^{N} \rho_j(\tau)\frac{\partial^2 \phi(|x-x_j|)}{\partial x^2}. \tag{20}$$

For the particular case when ϕ is the cubic spline,

$$\frac{\partial \phi(|x-x_j|)}{\partial x} = \begin{cases} 3(|x-x_j|)^2 & \text{if } x-x_j>0, \\ -3(|x-x_j|)^2 & \text{if } x-x_j<0, \end{cases} \tag{21}$$

$$\frac{\partial^2 \phi(|x-x_j|)}{\partial x^2} = 6(|x-x_j|). \tag{22}$$

In this research, we chose the cubic spline rather than the more popular MQ and IMQ as the basis function because of its simplicity, accuracy and lack of shape parameters.

3.1 *Transforming the PIDE to a System of ODEs using RBFs*

Given a set of interpolation points $x_1,\ldots,x_j,\ldots,x_N$ in $\mathbb{R}$ and an RBF ϕ, we can construct $N\times N$ matrices $\boldsymbol{A}$, $\boldsymbol{A}_x$ and $\boldsymbol{A}_{xx}$ that are defined by $\big(\phi(|x_i-x_j|)\big)_{1\le i,j\le N}$, $\big(\phi^{'}(|x_i-x_j|)\big)_{1\le i,j\le N}$ and $\big(\phi^{''}(|x_i-x_j|)\big)_{1\le i,j\le N}$, respectively. In this case, the x_j values are chosen according to the equally spacing method (ESM) described in the literature [23, 24, 29]. The equally spacing method provides a mechanism for choosing equally spaced points in a finite interval. Using the ESM, we determine an interval $[x_{\min},x_{\max}]$ outside of which we can neglect the contribution of $u(x,\tau)$ to the global integral term of the PIDE (10), and for a given $N=0,1,2,\ldots,$

$$x_j := x_j^{\Delta x} = x_{\min} + j\Delta x,\ j=0,1,2,\ldots,N-1 \tag{23}$$

where $\Delta x = (x_{\max}-x_{\min})/N$. We also define a matrix-valued function $y \to \boldsymbol{A}(y)$ by $\big(\phi(|x_i+y-x_j|)\big)_{1\le i,j\le N}$. If we substitute $\widetilde{u}(x,\tau)$ for $u(x,\tau)$ in (10) and require that the PIDE be satisfied in the interpolation points x_j, we arrive at the following system of ODEs for the vector $\boldsymbol{\rho}(\tau) := \big(\rho_1(\tau),\ldots,\rho_N(\tau)\big)$:

$$\begin{aligned} \boldsymbol{A}\boldsymbol{\rho}_\tau = \frac{\sigma^2}{2}\boldsymbol{A}_{xx}\boldsymbol{\rho} + \left(r-q-\frac{\sigma^2}{2}-\lambda\eta\right)\boldsymbol{A}_x\boldsymbol{\rho} + (r+\lambda)\boldsymbol{A}\boldsymbol{\rho} + \\ \lambda\left(\int_{-\infty}^{\infty}\boldsymbol{A}(y)f(y)\,\mathrm{d}y\right)\boldsymbol{\rho}, \end{aligned} \tag{24}$$

where $\rho_\tau := \frac{\partial \rho}{\partial \tau}$, and we recall that $f(y)$ is

$$\begin{cases} \frac{1}{\sigma_J\sqrt{2\pi}} e^{-\frac{(y-\mu_J)^2}{2\sigma_J^2}} & \text{for the Merton model or} \\ p\alpha_1 e^{-\alpha_1 x}\mathbb{1}_{x\geq 0} + (1-p)\alpha_2 e^{\alpha_2 x}\mathbb{1}_{x\leq 0} & \text{for the Kou model.} \end{cases}$$

Before applying a suitable numerical integration algorithm to the integral terms in (24), we truncate the integrals from an infinite to finite computational range. Briani *et al.* [9], Cont and Voltchkova [17], Tankov and Voltchkova [51] and d'Halluin *et al.* [18, 19] have provided different numerical techniques for determining a finite computational range to reduce errors in the numerical approximation when performing this truncation. In this paper, we adopt the Briani *et al.* numerical technique for truncating the integral domain of our PIDE (cf. [9]) in both the Merton and Kou models. A proof is provided in Appendix A. If $\epsilon > 0$, the formula for selecting a bounded interval $[y_{-\epsilon}, y_\epsilon]$ for the set of points y in the Merton case is as follows:

$$y_\epsilon = \sqrt{-2\sigma_J^2 \log(\epsilon\sigma_J\sqrt{2\pi}/2)} + \mu_J, \ \forall\, y \geq 0 \tag{25}$$

$$y_{-\epsilon} = -y_\epsilon, \ \forall\, y < 0, \tag{26}$$

and in the Kou model we have

$$y_\epsilon = \log\big(\epsilon/p\big)/(1-\alpha_1), \ \forall\, y \geq 0 \tag{27}$$

$$y_{-\epsilon} = -\log\big(\epsilon/(1-p)\big)/(1-\alpha_2), \ \forall\, y < 0. \tag{28}$$

We therefore transform equation (24) into

$$\begin{aligned} \boldsymbol{A}\boldsymbol{\rho}_\tau = \frac{\sigma^2}{2}\boldsymbol{A}_{xx}\boldsymbol{\rho} + \left(r - q - \frac{\sigma^2}{2} - \lambda\eta\right)\boldsymbol{A}_x\boldsymbol{\rho} + (r+\lambda)\boldsymbol{A}\boldsymbol{\rho} + \\ \lambda\left(\int_{y_{-\epsilon}}^{y_\epsilon} \boldsymbol{A}(y)f(y)\,\mathrm{d}y\right)\boldsymbol{\rho}. \end{aligned} \tag{29}$$

We use the adaptive Gauss-Kronrod quadrature in MATLAB to evaluate the matrix of the integrals in (29), which leads to the following approximation:

$$\int_{y_{-\epsilon}}^{y_\epsilon} \phi(|x_i + y - x_j|)f(y)\,\mathrm{d}y \approx \sum_{k=1}^{m} w_k\phi(|x_i + y_k - x_j|)f(y_k), \tag{30}$$

where w_k and y_k are suitable quadrature weights and quadrature points, respectively ; see [50] for details. To simplify the notation, we set

$$F(x_i - x_j) = \sum_{k=1}^{m} w_k\phi(|x_i + y_k - x_j|)f(y_k).$$

Then, the integrals in equation (29) can be approximated by

$$\int_{y_{-\epsilon}}^{y_\epsilon} \boldsymbol{A}(y) f(y)\,\mathrm{d}y \approx \begin{bmatrix} F(x_1-x_1) & F(x_1-x_2) & \dots & F(x_1-x_N) \\ F(x_2-x_1) & F(x_2-x_2) & \dots & F(x_2-x_N) \\ \dots & \dots & \dots & \dots \\ F(x_N-x_1) & F(x_N-x_2) & \dots & F(x_N-x_N) \end{bmatrix} = \boldsymbol{C}(y). \tag{31}$$

Substituting (31) into equation (29), we arrive at the new approximate equation:

$$\boldsymbol{A}\boldsymbol{\rho}_{\boldsymbol{\tau}} = \frac{\sigma^2}{2}\boldsymbol{A}_{xx}\boldsymbol{\rho} + \left(r - q - \frac{\sigma^2}{2} - \lambda\eta\right)\boldsymbol{A}_x\boldsymbol{\rho} + (r+\lambda)\boldsymbol{A}\boldsymbol{\rho} + \lambda\boldsymbol{C}(y)\boldsymbol{\rho}. \tag{32}$$

Because the cubic spline is a strictly conditionally positive definite function of order 2, the invertibility of $\boldsymbol{A}$ is not assumed without adding a real-valued polynomial of degree 1 in (17) [cf. 53]. Nevertheless, Bos and Salkauskas proved that $\boldsymbol{A}$ is non-singular in the univariate case [cf. 7, Theorem 5.1]. As a result, the invertibility of $\boldsymbol{A}$ is still guaranteed.

Although the invertibility of $\boldsymbol{A}$ can be proven for all ϕ values of interest, the inverse of $\boldsymbol{A}$, $\boldsymbol{A}^{-1}$, may often be ill conditioned to solve when its size increases [cf. 21, chapter 16], and an accurate solution using standard floating point arithmetic may be impossible. To address this problem, we factorise $\boldsymbol{A}$ into the following form [cf. 7, Theorem 3.7]:

$$\boldsymbol{A} = \boldsymbol{F}\boldsymbol{C}\boldsymbol{F}, \tag{33}$$

where $\boldsymbol{F}$ is an $N \times N$ matrix,

$$\begin{bmatrix} |x_1-x_1| & |x_1-x_2| & |x_1-x_3| & \dots & |x_1-x_N| \\ |x_2-x_1| & |x_2-x_2| & |x_2-x_3| & \dots & |x_2-x_N| \\ \vdots & & \ddots & & \vdots \\ |x_N-x_1| & |x_N-x_2| & |x_N-x_3| & \dots & |x_N-x_N| \end{bmatrix} \tag{34}$$

and $\boldsymbol{C}$ is a near-tridiagonal $N \times N$ matrix,

$$\begin{bmatrix} h-S & \frac{h}{2} & 0 & \cdots & 0 & \frac{S}{2} \\ \frac{h}{2} & 2h & \frac{h}{2} & 0 & \cdots & 0 \\ 0 & \frac{h}{2} & 2h & \frac{h}{2} & \cdots & 0 \\ \vdots & & & \ddots & & \vdots \\ 0 & 0 & \cdots & \frac{h}{2} & 2h & \frac{h}{2} \\ \frac{S}{2} & 0 & \dots & 0 & \frac{h}{2} & h-S \end{bmatrix}, \tag{35}$$

where h is the distance between x_{i+1} and x_i for $1 \le i \le N-1$ and $S = Nh$. We also have an explicit form of $\boldsymbol{F}^{-1}$ [cf. 7, Lemma 3.6] that is equal to

$$\begin{bmatrix} \frac{h-S}{2hS} & \frac{1}{2h} & 0 & \cdots & 0 & \frac{1}{2S} \\ \frac{1}{2h} & -\frac{1}{h} & \frac{1}{2h} & 0 & \cdots & 0 \\ 0 & \frac{1}{2h} & -\frac{1}{h} & \frac{1}{2h} & \cdots & 0 \\ \vdots & & & \ddots & & \vdots \\ 0 & 0 & \cdots & \frac{1}{2h} & -\frac{1}{h} & \frac{1}{2h} \\ \frac{1}{2S} & 0 & \ldots & 0 & \frac{1}{2h} & \frac{h-S}{2hS} \end{bmatrix}. \tag{36}$$

We perform Gaussian elimination with partial pivoting to calculate $\boldsymbol{C}^{-1}$. Then, we multiply both sides of (32) by $\boldsymbol{C}^{-1}$ and $\boldsymbol{F}^{-1}$ to obtain the following homogeneous system of ODEs with constant coefficients:

$$\begin{aligned} \boldsymbol{\rho}_\tau &= \boldsymbol{F}^{-1}\boldsymbol{C}^{-1}\boldsymbol{F}^{-1}\Big(\frac{\sigma^2}{2}\boldsymbol{A}_{xx} + \big(r - q - \frac{\sigma^2}{2} - \lambda\eta\big)\boldsymbol{A}_x + (r+\lambda)\boldsymbol{A} + \lambda\boldsymbol{C}(y)\Big)\boldsymbol{\rho} \\ &=: \boldsymbol{\Theta}\boldsymbol{\rho}, \end{aligned} \tag{37}$$

where $\boldsymbol{\Theta}$ is defined by the left-hand side. After some numerical experimentation, we found that the matrix $\boldsymbol{\Theta}$ is stiff. To illustrate why $\boldsymbol{\Theta}$ is stiff, we use the following example. Suppose we select -10 and 10 as the maximum and minimum logarithmic prices $x_{\min}$ $\big(\log(S_{\min})\big)$ and $x_{\max}$ $\big(\log(S_{\max})\big)$ in equation (23), respectively. Then, we use (23) to generate a list of 100 interpolation points. Based on the previously mentioned procedures and ideas, we can obtain the 100×100 matrix $\boldsymbol{\Theta}$ in (37). Then, we measure the stiffness ratio of $\boldsymbol{\Theta}$, which is the quotient of the largest and smallest eigenvalues of the Jacobian matrix $\boldsymbol{\Theta}$. Thus, we obtain a ratio of 1.2864×10^5, which implies that equation (37) is a stiff ODE, and therefore, we must use an implicit method to solve the ODEs, e.g., backward differentiation formulas (BDFs), a modified Rosenbrock formula of order 2, the trapezoidal rule or TR-BDF2, or an implicit Runge-Kutta formula with a first stage that is a trapezoidal rule step and a second stage that is a backward differentiation formula of order two . In this paper, we use the first option.

4. Numerical Results

4.1 *European Vanilla Options*

We first present a simple method for distributing a set of interpolation points. We then present our interpolation solutions and their convergence rates under the Black-Scholes and jump-diffusion models. Based on the cubic spline interpolation scheme, we further derive explicit delta and gamma hedging formulas and graphically illustrate the results. Finally, we compare the numerical results of the cubic spline interpolation scheme with those from the FDM and FEM.

A good method for placing interpolation points can determine the accuracy of our scheme. To achieve this goal, we set a range of $[x_{\min}, x_{\max}]$ and create N interpolation points via EMS (23). We then distribute the first $N/2$ points uniformly in $[x_{\min}, \log(K)]$, where K is the strike price, and the remaining points in $[\log(K), x_{\max}]$. Our scheme for distributing the interpolation points is illustrated in Figure 1.

In option trading, the region of most interest occurs when the mean of the stock prices approaches the strike price. Typically, the probability is low for a stock to default or to diverge greatly from the strike price. Therefore, we define the region

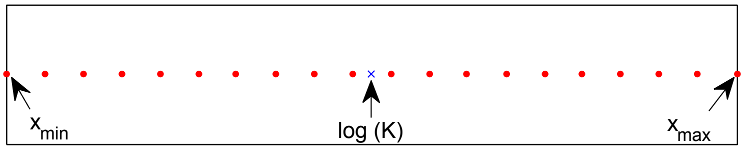


Figure 1. Uniform distributions of the interpolation points around the strike price obtained using EMS. The red dots represent the interpolation points. The blue cross is the location of the logarithmic strike price.

of interest as follows:

$$\hat{x}_i \in [\hat{x}_{\min}, \hat{x}_{\max}] := [\,\log(K/20), \log(2K)\,]. \tag{38}$$

Based on this region, we can test the accuracy of our cubic spline interpolation scheme using a set of evaluation points $\hat{x}_i^{\Delta x}$. We determine the grid points

$$\hat{x}_i := \hat{x}_i^{\Delta x} = \hat{x}_{\min} + j\Delta\hat{x},\ j = 0, 1, 2, \ldots, N_{\text{eval}} - 1 \tag{39}$$

where $\Delta\hat{x} = (\hat{x}_{\max} - \hat{x}_{\max})/N_{\text{eval}}$ with $x_{\min} \leq \hat{x}_{\min} \leq \hat{x}_{\max} \leq x_{\max}$ and N_{eval} is the number of evaluation points chosen. We will use the following three different measures for the errors: the maximum error

$$E_\infty = \max_{0 \leq i \leq N_{\text{eval}}} |f(\hat{x}_i) - \widetilde{u}(\hat{x}_i)|, \tag{40}$$

the root-mean-square (rms) error

$$E_2 = \sqrt{\frac{1}{N_{\text{eval}}} \sum_{0 \leq i \leq N_{\text{eval}}} |V(e^{\hat{x}_i}, t) - \widetilde{u}(\hat{x}_i)|^2}, \tag{41}$$

and the relative error

$$E_{\text{rel.}}(\hat{x}, t) = \frac{|V(e^{\hat{x}}, t) - \widetilde{u}(\hat{x}, t)|}{V(e^{\hat{x}}, t)}, \tag{42}$$

where $V(e^{\hat{x}}, t)$ and $\widetilde{u}(\hat{x}, t)$ are the exact and approximate values at point $(\hat{x}, t)$, respectively.

We also calculate the rate of convergence of the maximum error and rms error using $E_\infty(\hat{x}_i, T)$ and $E_2(\hat{x}_i, T)$. We define the following formulae:

$$E_\infty(\hat{x}_i, T) = C(1/N)^{R_\infty} \tag{43}$$

for the maximum error and

$$E_2(\hat{x}_i, T) = C(1/N)^{R_2} \tag{44}$$

for the rms error, where N is the number of interpolation points, C is a constant and R_2 is the rate of convergence, which is linear when it equals one and quadratic when it equals two.

Because the option price is approximated by the cubic spline interpolation scheme, we can develop approximate formulas to compute option Greeks (sensitivities in the option value to changes in the price of the underlying asset price model parameters). We focus only on expressing the formulae for both delta $\triangle$, the sensitivity or the rate of change in the option price $\widetilde{u}$ with respect to the change in the underlying logarithmic price and gamma Γ, the rate of change in delta with respect to the change in the underlying price. For $\triangle$ at any time τ, we have

$$\triangle = \left.\frac{\partial \widetilde{u}(x,\tau)}{\partial x}\right|_{\hat{x}} = \sum_{j=1}^{N} \rho_j(\tau) \left.\frac{\partial \phi(|x-x_j|)}{\partial x}\right|_{\hat{x}} \tag{45}$$

and for Γ, we have

$$\Gamma = \left.\frac{\partial \widetilde{u}(x,\tau)}{\partial x}\right|_{\hat{x}} = \sum_{j=1}^{N} \rho_j(\tau) \left.\frac{\partial^2 \phi(|x-x_j|)}{\partial x^2}\right|_{\hat{x}}. \tag{46}$$

As discussed in Section 3, the explicit forms of $\partial\phi(|x-x_j|)/\partial x$ and $\partial^2\phi(|x-x_j|)/\partial x^2$ are equal to (21) and (22), respectively.

The analytical price of a European call/put option in the Merton jump-diffusion model [44] is given by

$$V_{\mathrm{MJ}}(S_t,\tau,K,r,q,\sigma)$$

$$= \sum_{k=0}^{\infty} \frac{e^{-\lambda(1+\eta)\tau}((\lambda(1+\eta)\tau)^k}{k!} V_{BS}(S_t,\tau,K,r_k,\sigma_k,q), \tag{47}$$

where $\tau = T-t$ is the time to maturity, $\eta = e^{\mu_J + \frac{\sigma_J^2}{2}} - 1$ represents the expected percentage change in the stock price originating from a jump, $\sigma_k^2 = \sigma^2 + \frac{k\sigma_J^2}{T-t}$ is the observed volatility, $r_k = r - \lambda\eta + k\log(1+\eta)/(T-t)$, q is the dividend and V_{BS} the Black-Scholes price of a call and put computed as

$$V_{BS}(S_t,\tau,K,r_k,\sigma_k,q)$$
$$= \begin{Bmatrix} S_t e^{-q\tau}\Phi(d_{+,k}) - Ke^{-r_k\tau}\Phi(d_{-,k}) & \text{call option,} \\ Ke^{-r_k\tau}\Phi(-d_{-,k}) - S_t e^{-q\tau}\Phi(-d_{+,k}) & \text{put option,} \end{Bmatrix},$$

where $\Phi(\cdot)$ is the cumulative normal distribution and

$$d_{+,k} = \frac{\log(S_t/K) + (r_k - q + \sigma_k^2/2)\tau}{\sigma_k\sqrt{\tau}}, \quad d_{-,k} = d_{+,k} - \sigma_k\sqrt{\tau}.$$

For the derivation of $V_{\mathrm{MJ}}(S_t,\tau,K,r,q,\sigma)$, we refer the reader to [16, 44].

In general, for models in which the characteristic function of the Lévy process is known, an analytical solution for the PIDE (10) may be found using Fourier analysis [12, 38]. For the sake of simplicity and accuracy, we propose the Fourier space time-stepping method of Jackson *et al.* rather than that of Carr-Madan [12] or the FFT method of Lewis [38]. The idea of this method is based on a Fourier transform of the PIDE. Using an FFT and inverse fast Fourier transform (FFT^{-1}), the European option price can be obtained. The pricing formula for evaluating the European option can be expressed as follows:

$$V_{\mathrm{Kou}}(S,\tau,K,r,\sigma,q) = \mathrm{FFT}^{-1}[\,\mathrm{FFT}\,[V_{\mathrm{Kou}}(S,T)\,]e^{\psi\tau}\,] \tag{48}$$

where $\psi(z)$ is the characteristic function of the Kou model, which can be defined as

$$-\frac{\sigma^2 z^2}{2} + iz\gamma_c + \lambda\Big(\frac{p\alpha_1}{\alpha_1 - iz} + \frac{(1-p)\alpha_2}{\alpha_2 + iz} - 1\Big),$$

and $V_{\text{Kou}}(S,T)$ is the payoff function (11). For more details on this method, we refer the reader to [31]. This method reportedly has a second-order convergence in space in the European cases.

Our RBF algorithm for numerically solving (10) with initial condition (11) is as follows:

(1) Find the RBF approximation to the initial value $u(x,0)$ using the ESM (see (23)) to obtain a set of interpolation points $x_1,\ldots,x_n$ and an initial vector $\boldsymbol{\rho}(0) = \big(\rho_1(0),\ldots,\rho_N(0)\big)$.
(2) Then, use $\boldsymbol{\rho}(0)$ as the initial value for the system (37). By using any stiff ODE solver, we can determine $\boldsymbol{\rho}(T)$ at time T.
(3) Finally, substitute $\boldsymbol{\rho}(T)$ back into $\sum_{j=1}^{N}\rho_j(T)\phi(|x-x_j|)$ to obtain an approximate value for $u(x,T)$.

In our numerical experiment, we implement the algorithm in MATLAB R2007b and, as we did above, set our maximum and minimum logarithmic prices $x_{\min}$ $\big(\log(S_{\min})\big)$ and $x_{\max}$ $\big(\log(S_{\max})\big)$ to -10 and 10, respectively. To obtain a more accurate approximation of the integral in (29), we set ϵ in both (25) and (27) to 3.72×10^{-40} to determine a finite computational interval $[y_{-\epsilon}, y_\epsilon]$. Moreover, we use the function *quadgk*, which implements an adaptive Gauss-Kronrod quadrature for computing equation (30) and the function $ode15s$, which implements second-order backward differentiation formulas (BDFs) to calculate equation (37). The principal reason for choosing it is that according to [30], first- and second-order BDFs are A-stable (the stability region includes the entire left half of the complex plane). Because (37) is stiff, according to Theorem 4.11 (The Dahlquist second barrier) [30], the highest order of an A-stable multistep method,[1] such as BDFs, is two.

All tabulated parameters except those in Tables 3, 6 and 9 are chosen from different reports in the literature. The parameter $\sigma = 1$ in Tables 3, 6 and 9 is selected to stress our numerical algorithm. From Table 1 to 9, E_∞ and E_2 decrease when the number of interpolation points N increases. Our cubic spline interpolation scheme can obtain second-order convergence in space due to the limited smoothness of the cubic spline, which has second-order convergence (cf. [53]). In Figures 2, 3 and 4, oscillations do not occur around the strike K for small values of T when we approximate Δ and Γ. In Table 10, we compare the results of the FD used by Briani *et al.* [9] with those using our cubic spline interpolation scheme. Our numerical approximation scheme can achieve lower $E_{\text{rel.}}(\log S,T)$ values than either the ARS-233 or Explicit scheme. Tables 10 and 12 provide additional comparisons between the accuracy of our cubic spline approximation scheme and that of Almendral and Oosterlee's FDM and FEM with BDF2. To illustrate a fair comparison, we set our maximum and minimum logarithmic prices $x_{\min}$ and $x_{\max}$ to match those proposed by Almendral and Oosterlee in their numerical experiments. Thus, we set $[x_{\min}\ x_{\max}]$ to [-4 4] and [-6 6] in the Merton model (Table 11) and Kou model (Table 12), respectively. Our cubic spline interpolation scheme can attain a lower $E_{\text{rel.}}(\log S,T)$ value than the FDM or FEM with BDF2 in both the Merton and Kou cases.

[1]Multistep methods are used to the numerically solve ordinary differential equations. Conceptually, a numerical method begins at an initial point and takes a short step forward in time to the next solution point. The process continues with subsequent steps to map out the solution.

Table 1. E_∞ and E_2 of the cubic spline interpolation for pricing a European put under the Black-Scholes model. The parameters are as follows: $r = 0.04$, $q = 0$, $\sigma = 0.29$, $K = 1$ and $T = 1$. The parameters are taken from the literature [26]. The order of convergence is 2 in space.

N[a]	$E_\infty(\hat{x}_i, T)$[b]	R_∞	$E_2(\hat{x}_i, T)$[b]	R_2
100	4.207101E-03	N/A	1.864736E-03	N/A
600	1.195088E-04	1.988	5.143665E-05	2.004
1100	3.554622E-05	2.000	1.528321E-05	2.002
1600	1.679290E-05	2.001	7.219811E-06	2.001
2100	9.745141E-06	2.001	4.189909E-06	2.001
2600	6.354765E-06	2.002	2.732818E-06	2.001
3100	4.468110E-06	2.003	1.921950E-06	2.001
3600	3.311319E-06	2.004	1.424931E-06	2.001

[a] N is the number of interpolation points. $\hat{x}_i = \log S_i$ is any evaluation point ranging from $S = 0.05$ to 2, of which there are 1950. [b] T is the maturity time.

Table 2. E_∞ and E_2 of the cubic spline interpolation for pricing a European put under the Black-Scholes model. The parameters are as follows: $r = 0.05$, $q = 0$, $\sigma = 0.2$, $K = 1$ and $T = 2$. The parameters are taken from the literature [**?**]. The order of convergence is 2 in space.

N[a]	$E_\infty(\hat{x}_i, T)$[b]	R_∞	$E_2(\hat{x}_i, T)$[b]	R_2
100	1.924131E-02	N/A	4.690135E-03	N/A
600	7.143939E-04	1.838	1.296858E-04	2.003
1100	2.171519E-04	1.965	3.870772E-05	1.995
1600	1.031950E-04	1.986	1.830673E-05	1.998
2100	6.002721E-05	1.992	1.063352E-05	1.998
2600	3.919766E-05	1.995	6.934013E-06	2.002
3100	2.758717E-05	1.997	4.877540E-06	2.000
3600	2.046213E-05	1.998	3.616699E-06	2.000

[a] N is the number of interpolation points. $\hat{x}_i = \log S_i$ is any evaluation point ranging from $S = 0.05$ to 2, of which there are 1950. [b] T is the maturity time.

Table 3. E_∞ and E_2 of the cubic spline interpolation for pricing a European put under the Black-Scholes model. The parameters are as follows: $r = 0.3$, $q = 0.1$, $\sigma = 1$, $K = 1$ and $T = 0.25$, whereas the parameter $\sigma = 1$ is selected to stress our numerical algorithm. The order of convergence is 2 in space.

N[a]	$E_\infty(\hat{x}_i, T)$[b]	R_∞	$E_2(\hat{x}_i, T)$[b]	R_2
100	2.325676E-03	0.000	1.404611E-03	N/A
600	6.473617E-05	1.999	3.856043E-05	2.007
1100	1.923322E-05	2.002	1.145625E-05	2.002
1600	9.079037E-06	2.003	5.411921E-06	2.001
2100	5.265272E-06	2.004	3.140776E-06	2.001
2600	3.430306E-06	2.006	2.048580E-06	2.001
3100	2.406208E-06	2.016	1.441039E-06	2.000
3600	1.782442E-06	2.007	1.068202E-06	2.002

[a] N is the number of interpolation points. $\hat{x}_i = \log S_i$ is any evaluation point ranging from $S = 0.05$ to 2, of which there are 1950. [b] T is the maturity time.

Table 4. E_∞ and E_2 of the cubic spline interpolation for pricing a European call under the Merton model. The parameters are as follows: $r = 0.05$, $q = 0$, $\sigma = 0.15$, $\sigma_J = 0.45$, $\mu_J = -0.9$, $\lambda = 0.1$, $K = 1$ and $T = 0.25$. The parameters are taken from [6]. The order of convergence is 2 in space.

N[a]	$E_\infty(\hat{x}_i, T)$[b]	R_∞	$E_2(\hat{x}_i, T)$[b]	R_2
100	1.428497E-02	N/A	3.749983E-03	N/A
600	4.642130E-04	1.912	1.011341E-04	2.016
1100	1.402519E-04	1.975	3.011378E-05	1.999
1600	6.640377E-05	1.995	1.423346E-05	2.000
2100	3.860331E-05	1.995	8.262241E-06	2.000
2600	2.518672E-05	1.999	5.389115E-06	2.001
3100	1.772559E-05	1.997	3.790660E-06	2.000
3600	1.314288E-05	2.000	2.810697E-06	2.000

[a] N is the number of interpolation points. $\hat{x}_i = \log S_i$ is any evaluation point ranging from $S = 0.05$ to 2, of which there are 1950. [b] T is the maturity time.

Table 5. E_∞ and E_2 of the cubic spline interpolation for pricing a European put under the Merton jump-diffusion model. The parameters are as follows: $r = 0.05$, $q = 0.02$, $\sigma = 0.15$, $\sigma_J = 0.4$, $\mu_J = -1.08$, $\lambda = 0.1$, $K = 1$ and $T = 0.1$. The parameters are taken from [6]. The order of convergence is 2 in space.

N[a]	$E_\infty(\hat{x}_i, T)$[b]	R_∞	$E_2(\hat{x}_i, T)$[b]	R_2
100	1.956920E-02	N/A	4.723349E-03	N/A
600	7.326011E-04	1.833	1.305576E-04	2.003
1100	2.240092E-04	1.955	3.898655E-05	1.994
1600	1.069094E-04	1.974	1.844062E-05	1.998
2100	6.223777E-05	1.990	1.071235E-05	1.997
2600	4.062560E-05	1.997	6.985440E-06	2.002
3100	2.859186E-05	1.997	4.913762E-06	2.000
3600	2.121748E-05	1.995	3.643595E-06	2.000

[a] N is the number of interpolation points. $\hat{x}_i = \log S_i$ is any evaluation point ranging from $S = 0.05$ to 2, of which there are 1950. [b] T is the maturity time.

Table 6. E_∞ and E_2 of the cubic spline interpolation for pricing a European call under the Merton model. The parameters are as follows: $r = 0.05$, $q = 0.01$, $\sigma = 1$, $\sigma_J = 0.6$, $\mu_J = -1.08$, $\lambda = 0.1$, $K = 1$ and $T = 1$, whereas the parameter $\sigma = 1$ is selected to stress our numerical algorithm. The order of convergence is 2 in space.

N[a]	$E_\infty(\hat{x}_i, T)$[b]	R_∞	$E_2(\hat{x}_i, T)$[b]	R_2
100	1.026524E-03	N/A	7.090253E-04	N/A
600	2.819557E-05	2.006	1.945356E-05	2.007
1100	8.415823E-06	1.995	5.762520E-06	2.007
1600	3.999351E-06	1.986	2.712396E-06	2.011
2100	2.373272E-06	1.919	1.559774E-06	2.035
2600	1.601472E-06	1.842	1.004746E-06	2.059
3100	1.136188E-06	1.951	7.021072E-07	2.038
3600	8.358248E-07	2.053	5.221973E-07	1.980

[a] N is the number of interpolation points. $\hat{x}_i = \log S_i$ is any evaluation point ranging from $S = 0.05$ to 2, of which there are 1950. [b] T is the maturity time.

Table 7. E_∞ and E_2 of the cubic spline interpolation for pricing a European put under the Kou model. The parameters are as follows: $r = 0$, $q = 0$, $\sigma = 0.2$, $\alpha_1 = 3$, $\alpha_2 = 2$, $\lambda = 0.2$, $p = 0.5$, $K = 1$ and $T = 0.2$. The parameters are taken from the literature [2]. The order of convergence is 2 in space.

N[a]	$E_\infty(\hat{x}_i, T)$[b]	R_∞	$E_2(\hat{x}_i, T)$[b]	R_2
100	1.239165E-02	N/A	3.422908E-03	N/A
600	3.932126E-04	1.926	9.440247E-05	2.004
1100	1.179555E-04	1.986	2.808850E-05	2.000
1600	5.589111E-05	1.993	1.327392E-05	2.000
2100	3.246588E-05	1.998	7.705266E-06	2.000
2600	2.118103E-05	2.000	5.025765E-06	2.001
3100	1.490021E-05	2.000	3.535171E-06	2.000
3600	1.105067E-05	1.999	2.621377E-06	2.000

[a] N is the number of interpolation points. $\hat{x}_i = \log S_i$ is any evaluation point ranging from $S = 0.05$ to 2, of which there are 1950. [b] T is the maturity time.

Table 8. E_∞ and E_2 of the cubic spline interpolation for pricing a European call under the Kou model. The parameters are as follows: $r = 0.05$, $q = 0$, $\sigma = 0.15$, $\alpha_1 = 3.0465$, $\alpha_2 = 3.0465$, $\lambda = 0.1$, $p = 0.3445$, $K = 1$ and $T = 0.25$. The parameters are taken from the literature [14]. The order of convergence is 2 in space.

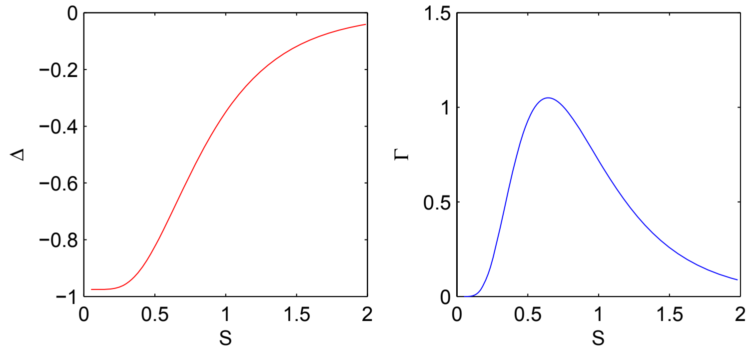
N[a]	$E_\infty(\hat{x}_i, T)$[b]	R_∞	$E_2(\hat{x}_i, T)$[b]	R_2
100	1.433875E-02	N/A	3.766745E-03	N/A
600	4.665677E-04	1.912	1.022079E-04	2.013
1100	1.404381E-04	1.981	3.043034E-05	1.999
1600	6.660275E-05	1.991	1.438190E-05	2.000
2100	3.868283E-05	1.998	8.348098E-06	2.000
2600	2.522395E-05	2.002	5.444331E-06	2.001
3100	1.773247E-05	2.003	3.828943E-06	2.001
3600	1.314079E-05	2.004	2.838628E-06	2.001

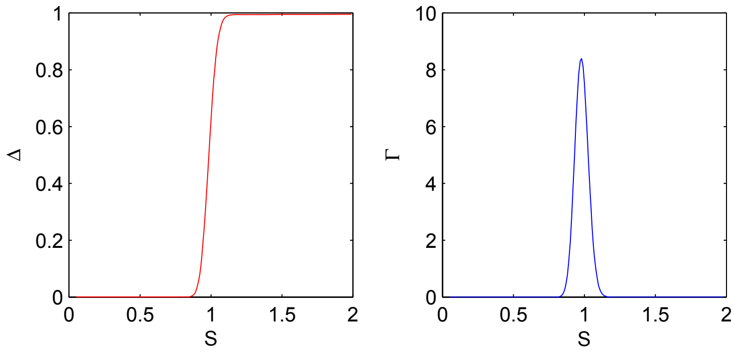
[a] N is the number of interpolation points. $\hat{x}_i = \log S_i$ is any evaluation point ranging from $S = 0.05$ to 2, of which there are 1950. [b] T is the maturity time.

Table 9. E_∞ and E_2 of the cubic spline interpolation for pricing a European put under the Kou model. The parameters are as follows: $r = 0.04$, $q = 0.03$, $\sigma = 1$, $\alpha_1 = 4$, $\alpha_2 = 4$, $\lambda = 0.3$, $p = 0.6$ $K = 1$ and $T = 1$, whereas the parameter $\sigma = 1$ is selected to stress our numerical algorithm. The order of convergence is 2 in space.

N[a]	$E_\infty(\hat{x}_i, T)$[b]	R_∞	$E_2(\hat{x}_i, T)$[b]	R_2
100	1.080306E-03	N/A	7.074108E-04	N/A
600	2.973137E-05	2.005	1.940773E-05	2.007
1100	8.870629E-06	1.995	5.757611E-06	2.005
1600	4.229400E-06	1.977	2.712641E-06	2.009
2100	2.490583E-06	1.947	1.567674E-06	2.016
2600	1.674611E-06	1.859	1.014582E-06	2.037
3100	1.191565E-06	1.935	7.096338E-07	2.032
3600	9.018770E-07	1.863	5.232205E-07	2.038

[a] N is the number of interpolation points. $\hat{x}_i = \log S_i$ is any evaluation point ranging from $S = 0.05$ to 2, of which there are 1950. [b] T is the maturity time.





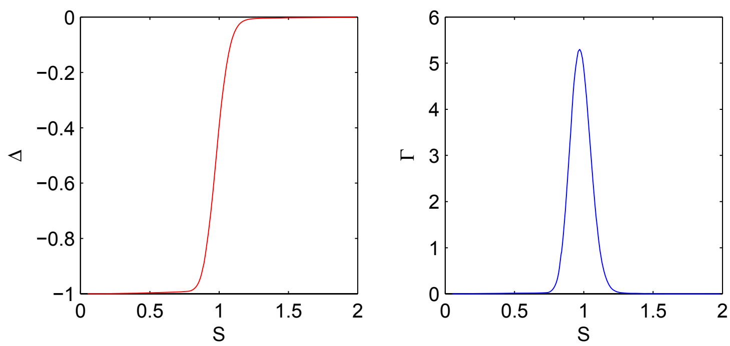


Figure 4. Put options delta Δ (left) and gamma Γ (right) in the Kou model. The number of interpolation points is 3600. The number of evaluation points ranging from $S = 0.05$ to 2 is 1950. The input parameters are provided in the caption for Table 8.

Table 10. Comparison between the explicit scheme ([9]), ARS-233 scheme ([9]) and cubic spline interpolation scheme for evaluating European call/put options under the Merton jump-diffusion model. The input parameters are as follows: $r = 0.05$, $q = 0$, $\sigma = 0.2$, $\sigma_J = 0.8$, $\mu_J = 0$, $\lambda = 0.1$, $K = 100$, $T = 1$, and $x = \log 100$. The reference prices of 13.218501 (call) and 8.341444 (put) and the parameters are from [9].

		Explicit scheme		ARS-233 scheme	
	N	$Value$	$E_{\text{rel.}}(\log S, T)$	$Value$	$E_{\text{rel.}}(\log S, T)$
Call	1024	13.286915	5.175624E-03	13.287427	5.214358E-03
Put	1024	8.319940	2.57797E-03	8.326102	1.839249E-03
		Cubic spline		N/A	
	N	$Value$	$E_{\text{rel.}}(\log S, T)$	$Value$	$E_{\text{rel.}}(\log S, T)$
Call	1024	13.219358	6.489263E-05	N/A	N/A
Put	1024	8.342301	1.027679E-04	N/A	N/A

Table 11. Comparison of the FDM with BDF2 ([2]), the FEM with BDF2 ([2]) and the cubic spline interpolation scheme for evaluating a European call (put) under the Merton model. The input parameters are as follows: $r = 0$, $q = 0$, $\sigma = 0.2$, $\sigma_J = 0.5$, $\mu_J = 0$, $\lambda = 0.1$, $K = 1$, $T = 1$, and $S = 1$. The reference prices of 0.094135525 for both the call and put, and the parameters are from [2].

	FD with BDF2		FE with BDF2	
N	$Value$	$E_{\text{rel.}}(\log S, T)$	$Value$	$E_{\text{rel.}}(\log S, T)$
1025	9.411968E-02	1.682457e-04	9.412972E-02	6.165536E-05
	Cubic spline		N/A	
N	$Value$	$E_{\text{rel.}}(\log S, T)$	$Value$	$E_{\text{rel.}}(\log S, T)$
1025	9.413023E-02	5.621522E-005	N/A	N/A

Table 12. Comparison of the FDM with BDF2 ([2]), the FEM with BDF2 ([2]) and the cubic spline interpolation scheme for evaluating a European call (put) under the Kou model. The input parameters are as follows: $r = 0$, $q = 0$, $\sigma = 0.2$, $\alpha_1 = 3$, $\alpha_2 = 2$, $\lambda = 0.2$, $p = 0.5$, $K = 1$, $T = 0.2$, and $S = 1$. The reference prices of 0.0426761 for both the call and put, and the parameters are from [2].

	FD with BDF2		FE with BDF2	
N	$Value$	$E_{\text{rel.}}(\log S, T)$	$Value$	$E_{\text{rel.}}(\log S, T)$
513	4.240E-02	6.346096E-03	4.24579E-02	5.1285862E-03
	Cubic Spline		N/A	
N	$Value$	$E_{\text{rel.}}(\log S, T)$	$Value$	$E_{\text{rel.}}(\log S, T)$
513	4.254583E-02	3.061686E-03	N/A	N/A

4.2 *American Vanilla Put Options*

In this section, we adapt an RBF algorithm to compute American put option prices. We then compare the option prices obtained from our RBF algorithm with those obtained from the Jackson et al. FST methods [31]. As mentioned in Section 2, an American put option problem is a free-boundary problem because of the possibility of early exercise at any point during its lifetime, leading to the free-boundary condition

$$u(x,\tau) = \max\big(K - e^x, u(x,\tau)\big).$$

Together with the smooth pasting condition mentioned in Section 2, this uniquely determines the exercise boundary.

The Jackson et al. FST methods suggest that their solutions can achieve second order in space when they implement their methods to price American put options. The methods are implemented in the context of the LCP. As described in Section 2, the value of an American option $u(\tau,x)$ is always greater than or equal to the payoff function $G(e^x)$. To numerically maintain the condition $u(\tau,x) - G(e^x) \geq 0$ continuously (see Section 2), the boundary conditions must be applied. The numerical algorithm for this idea can be defined as follows:

$$\begin{aligned} &V(S,(m+1)\Delta t, K, r, \sigma, q) \\ &= \max\{\mathrm{FFT}^{-1}[\,\mathrm{FFT}\,[V(S, m\Delta t, K, r, \sigma, q)\,]e^{\psi\Delta t}\,], G(e^x), \} \end{aligned} \tag{49}$$

where the time interval Δt is obtained by dividing the time to maturity T by the total number M; $m\Delta$ is the time step, where $m \in \{0,1,2,\ldots,M-1\}$, $\psi(z)$ is the characteristic function of the Merton/Kou models, $V(S,(m+1)\Delta t, K, r, \sigma, q)$ is the American put price at time $(m+1)\Delta t$ and the payoff condition $G(e^x)$ is equal to $\max(K - e^x, 0)$. These methods are also required to switch between the real and Fourier spaces at each time step when the American option prices are calculated for each time interval because no convenient representation exists for the max(.,.) operator in Fourier space. For the full schematic and numerical description of this method, we refer the readers to [31].

As before, we use the ESM to approximate $u(x,0) = \max(K - e^x, 0)$ and continue to work with the interpolation points found at $\tau = 0$. The algorithm now reads as follows:

(1) Divide time to maturity T by the total number of time-steps M to obtain time interval Δt and create a list of equally spaced time-points $m\Delta t$, $m \in \{0,1,2,\ldots,M-1\}$.
(2) Find the RBF approximation for the initial value $u(x,0)$ using the ESM. This will yield a set of interpolation points $x_1,\ldots,x_n$, together with an initial vector $\boldsymbol{\rho}(0) = \big(\rho_1(0),\ldots,\rho_N(0)\big)$.
(3) Assume that we have already determined $\boldsymbol{\rho}(m\Delta t)$ (if $m = 0$, we know $\boldsymbol{\rho}(0)$) in equation (37). Solve the system of (stiff) ODEs to find $\boldsymbol{\rho}\big((m+1)\Delta t\big)$ at the next successive time step, $(m+1)\Delta t$.
(4) Then, at time $(m+1)\Delta t$, for each interpolation point x_i, define

$$u\big(x_i,(m+1)\Delta t\big) = \max\Big((K - e^{x_i}), \sum_{j=1}^{N} \rho_j\big((m+1)\Delta t\big)\phi(|x_i - x_j|)\Big).$$

(5) Find a new vector $\boldsymbol{\rho}\big((m+1)\Delta t\big)$ such that $u\big(x_i,(m+1)\Delta t\big) = \sum_{j=1}^{N} \rho_j\big((m+$

$1)\Delta t\big)\phi(|x_i - x_j|)$ for all i.

(6) Repeat Steps 3 to 5 until $m = M - 1$.

(7) Finally, substitute $\boldsymbol{\rho}(T)$ back into $\sum_{j=1}^{N} \rho_j(T)\phi(|x - x_j|)$ to obtain an approximate value for $u(x, T)$.

The settings of our numerical experiment are identical to those in Section 4.1. The results from Tables 13 to 18 suggest that our cubic spline interpolation scheme for pricing American put options is second order in spatial variables and first order in time variables when the number of interpolation numbers N and the number of time-steps M_0 are twofold and fourfold, respectively. Moreover, Figures 5 to 7 indicate that oscillations do not occur around the strike K for small or large values of T when we approximate Δ and Γ.

5. Conclusion

We implemented an RBF interpolation scheme to price American put and European call/put options using the jump-diffusion model. By utilising the numerical scheme of Briani *et al.*, we determined a finite computational range for the global integral of the PIDE. Our results suggest that the interpolation scheme can achieve second-order convergence in both spatial variables for computing European prices. Our other numerical results demonstrate that our scheme is also able to obtain second-order convergence in spatial variables and first-order convergence in time variables when pricing American put options. In addition, we compared our interpolation scheme against the FDM and FEM. Our results suggest that one can achieve a high level of accuracy by implementing our method. For the RBF interpolation, we used a cubic spline basis function rather than an MQ basis function. This basis function not only avoids the open question of choosing an optimal shape parameter for MQ but also avoids directly inverting an ill-conditioned cubic spline interpolant. Finally, throughout the analysis of both Δ and Γ, our RBF interpolation method can resolve the oscillation problem around the strike in both the American and European cases.

At this stage of development, the RBF interpolation scheme is first order in time for American put options, although a second-order time-stepping scheme, BDFs of order 2, was also implemented. We are investigating various approaches to improve the cubic spline interpolation for time variables and will discuss these efforts in a future paper. In principle, our method extends to pure jump Lévy-type models for the underlying stocks, such as the variance gamma (VG) model or CGMY model [cf. 11].

Table 13. E_∞ and E_2 of the cubic spline interpolation for pricing an American put under the Merton model. The parameters are as follows: $r = 0.05$, $q = 0$, $\sigma = 0.15$, $\sigma_J = 0.45$, $\mu_J = -0.9$, $\lambda = 0.1$, $K = 1$ and $T = 0.25$. The parameters are taken from [6]. The order of convergence is 2 in space and 1 in time.

N[a]	M_0[b]	$E_\infty(\hat{x}_i, T)$[c]	R_∞	$E_2(\hat{x}_i, T)$[c]	R_2
225	10	2.368536E-03	N/A	1.007946E-03	N/A
450	40	7.746936E-04	1.612	2.740154E-04	1.879
900	160	2.260415E-04	1.777	6.969946E-05	1.975
1800	640	6.362341E-05	1.829	1.888980E-05	1.884
3600	2560	1.613907E-05	1.979	4.715908E-06	2.002

[a] N is the number of interpolation points. $\hat{x}_i = \log S_i$ is any evaluation point ranging from $S = 0.05$ to 2, of which there are 1950. [b] M_0 is the number of time steps . [c] T is the maturity time.

Table 14. E_∞ and E_2 of the cubic spline interpolation for pricing an American put under the Merton model. The parameters are as follows: $r = 0.05$, $q = 0.02$, $\sigma = 0.15$, $\sigma_J = 0.4$, $\mu_J = -1.08$, $\lambda = 0.1$, $K = 1$ and $T = 0.1$. The parameters are taken from the literature [6]. The order of convergence is 2 in space and 1 in time.

N[a]	M_0[b]	$E_\infty(\hat{x}_i, T)$[c]	R_∞	$E_2(\hat{x}_i, T)$[c]	R_2
225	10	3.401417E-03	N/A	7.995993E-04	N/A
450	40	1.318325E-03	1.367	2.451148E-04	1.706
900	160	3.744579E-04	1.816	6.873071E-05	1.834
1800	640	1.055849E-04	1.826	1.927219E-05	1.834
3600	2560	2.823205E-05	1.903	5.121082E-06	1.912

[a] N is the number of interpolation points. $\hat{x}_i = \log S_i$ is any evaluation point ranging from $S = 0.05$ to 2, of which there are 1950. [b] M_0 is the number of time steps. [c] T is the maturity time.

Table 15. E_∞ and E_2 of the cubic spline interpolation for pricing an American put under the Merton model. The parameters are as follows: $r = 0.05$, $q = 0.01$, $\sigma = 1$, $\sigma_J = 0.6$, $\mu_J = -1.08$, $\lambda = 0.1$, $K = 1$ and $T = 1$, whereas the parameter $\sigma = 1$ is selected to stress our numerical algorithm. The order of convergence is 2 in space and 1 in time.

N[a]	M_0[b]	$E_\infty(\hat{x}_i, T)$[c]	R_∞	$E_2(\hat{x}_i, T)$[c]	R_2
225	10	4.935878E-03	N/A	1.613323E-03	N/A
450	40	1.236617E-03	1.997	3.725615E-04	2.114
900	160	3.093198E-04	1.999	9.101657E-05	2.033
1800	640	7.734030E-05	2.000	2.133679E-05	2.093
3600	2560	1.932168E-005	2.001	5.074520E-06	2.072

[a] N is the number of interpolation points. $\hat{x}_i = \log S_i$ is any evaluation point ranging from $S = 0.05$ to 2, of which there are 1950. [b] M_0 is the number of time steps. [c] T is the maturity time.

Table 16. E_∞ and E_2 of the cubic spline interpolation for pricing an American put under the Kou model. The parameters are as follows: $r = 0$, $q = 0$, $\sigma = 0.2$, $\alpha_1 = 3$, $\alpha_2 = 2$, $\lambda = 0.2$, $p = 0.5$, $K = 1$ and $T = 0.2$. The parameters are taken from the literature [2]. The order of convergence is 2 in space and 1 in time.

N[a]	M_0[b]	$E_\infty(\hat{x}_i, T)$[c]	R_∞	$E_2(\hat{x}_i, T)$[c]	R_2
225	10	1.508321E-03	N/A	5.589125E-04	N/A
450	40	7.233939E-04	1.060	1.759571E-04	1.667
900	160	1.958968E-04	1.885	4.733738E-05	1.894
1800	640	5.243753E-05	1.901	1.271703E-05	1.896
3600	2560	1.374207E-05	1.932	3.405083E-06	1.901

[a] N is the number of interpolation points. $\hat{x}_i = \log S_i$ is any evaluation point ranging from $S = 0.05$ to 2, of which there are 1950. [b] M_0 is the number of time steps. [c] T is the maturity time.

Table 17. E_∞ and E_2 of the cubic spline interpolation for pricing an American put under the Kou model. The parameters are as follows: $r = 0.05$, $q = 0$, $\sigma = 0.15$, $\alpha_1 = 3.0465$, $\alpha_2 = 3.0465$, $\lambda = 0.1$, $p = 0.3445$, $K = 1$ and $T = 0.25$. The parameters are taken from [14]. The order of convergence is 2 in space and 1 in time.

N[a]	M_0[b]	$E_\infty(\hat{x}_i, T)$[c]	R_∞	$E_2(\hat{x}_i, T)$[c]	R_2
225	10	1.933354E-03	N/A	8.983577E-04	N/A
450	40	8.487095E-04	1.188	2.783005E-04	1.691
900	160	2.497213E-04	1.765	7.257535E-05	1.939
1800	640	6.843085E-05	1.868	1.933309E-05	1.908
3600	2560	1.827216E-05	1.905	5.119491E-06	1.917

[a] N is the number of interpolation points. $\hat{x}_i = \log S_i$ is any evaluation point ranging from $S = 0.05$ to 2, of which there are 1950. [b] M_0 is the number of time steps. [c] T is the maturity time.

Table 18. E_∞ and E_2 of the cubic spline interpolation for pricing an American put under the Kou model. The parameters are as follows: $r = 0.04$, $q = 0.03$, $\sigma = 1$, $\alpha_1 = 4$, $\alpha_2 = 4$, $\lambda = 0.3$, $p = 0.6$, $K = 1$ and $T = 1$, whereas the parameter $\sigma = 1$ is selected to stress our numerical algorithm. The order of convergence is 2 in space and 1 in time.

N[a]	M_0[b]	$E_\infty(\hat{x}_i, T)$[c]	R_∞	$E_2(\hat{x}_i, T)$[c]	R_2
225	10	3.839148E-03	N/A	1.095217E-03	N/A
450	40	9.616353E-04	1.997	2.458977E-04	2.155
900	160	2.405238E-04	1.999	6.111403E-05	2.008
1800	640	6.013812E-05	2.000	1.508359E-05	2.019
3600	2560	1.490999E-05	2.012	3.768285E-06	2.001

[a] N is the number of interpolation points. $\hat{x}_i = \log S_i$ is any evaluation point ranging from $S = 0.05$ to 2, of which there are 1950. [b] M_0 is the number of time steps. [c] T is the maturity time.

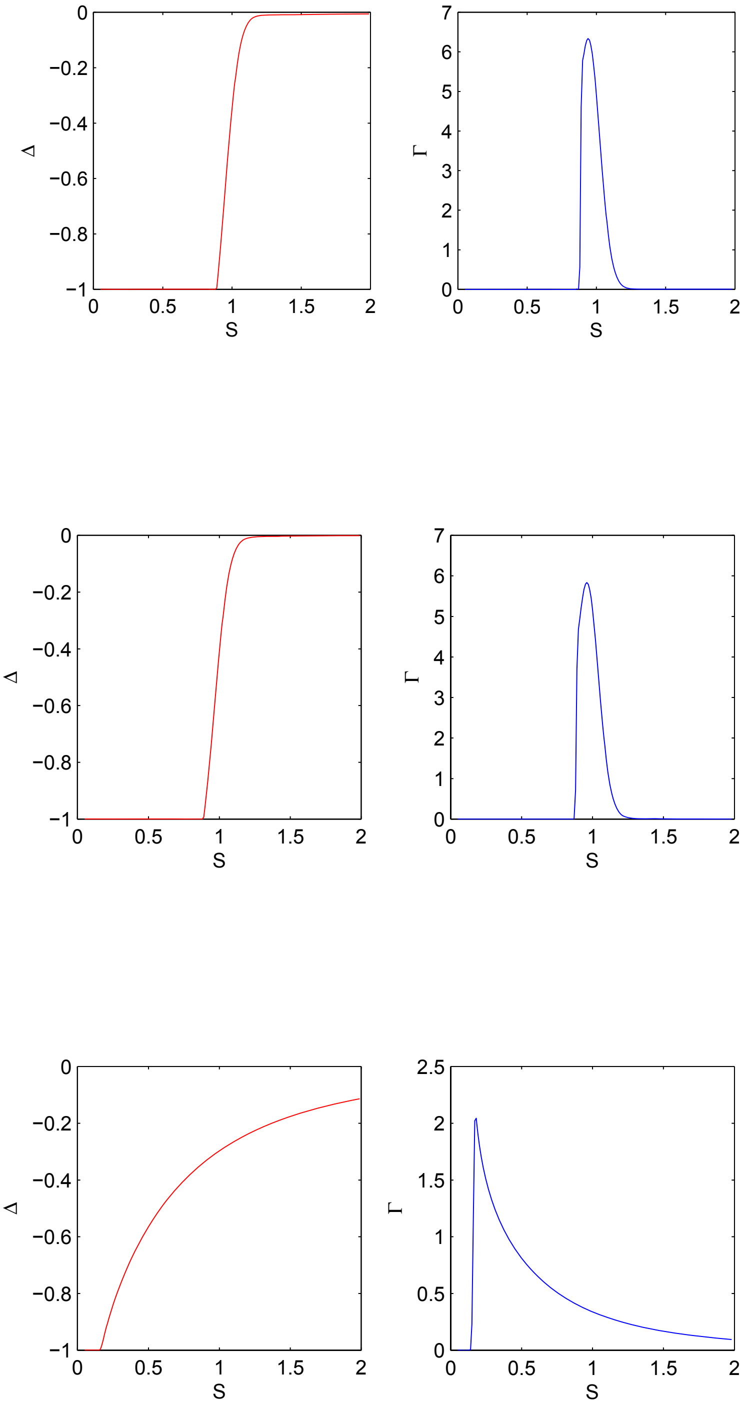


Figure 7. Put option delta Δ (left) and gamma Γ (right)in the Kou model. The number of interpolation points N is 1800, and the number of time steps M_0 is 640. The number of evaluation points ranging from $S = 0.05$ to 2 is 1950. The input parameters are provided in the caption for Table 18.

Appendix A. A Finite Computational Range in the Jump-diffusion Model

In the Merton model, suppose that a domain $\Omega \in \mathbb{R}$ for the European option price $u(x, \tau)$ satisfies the Lipchitz inequality such that

$$|u(x_1, \tau) - u(x_2, \tau)| \leq L|x_1 - x_2|, \ \forall x_1, x_2 \in \Omega.$$

Then, we choose a parameter $\epsilon > 0$ and select the bounded intervals $[y_{-\epsilon}, y_\epsilon]$ as the set of all points y that verify

$$k(y) = \frac{1}{\sqrt{2\pi}\sigma_J} e^{-\frac{(y-\mu_J)^2}{2\sigma_J^2}} \geq \epsilon.$$

Given the symmetry of $k(y)$, we set $y_{-\epsilon} = -y_\epsilon$. Then, the truncation of the integral domain yielding an error in the approximation of the problem can be estimated by

$$\left| \int_{-\infty}^{\infty} (u(x+y) - u(x))k(y)\,\mathrm{d}y - \int_{-y_\epsilon}^{y_\epsilon} (u(x+y) - u(x))k(y)\,\mathrm{d}y \right|$$

$$\leq L \left| \int_{-\infty}^{\infty} (x+y-x)k(y)\,\mathrm{d}y - \int_{-y_\epsilon}^{y_\epsilon} (x+y-x)k(y)\,\mathrm{d}y \right| \tag{A1a}$$

$$\leq L \left(\int_{-\infty}^{-y_\epsilon} |y|k(y)\,\mathrm{d}y + \int_{y_\epsilon}^{\infty} |y|k(y)\,\mathrm{d}y \right) \tag{A1b}$$

$$= 2\int_{y_\epsilon}^{\infty} y \frac{1}{\sqrt{2\pi}\sigma_J} \exp(-\frac{(y-\mu_J)^2}{2\sigma_J^2})\,\mathrm{d}y \tag{A1c}$$

$$= 2\int_{y_\epsilon-\mu_J}^{\infty} (y+\mu_J) \frac{1}{\sqrt{2\pi}\sigma_J} \exp(-\frac{y^2}{2\sigma_J^2})\,\mathrm{d}y \tag{A1d}$$

$$= 2\int_{y_\epsilon-\mu_J}^{\infty} (y+\mu_J) \frac{1}{\sqrt{2\pi}\sigma_J} \exp(-\frac{y^2}{2\sigma_J^2})\,\mathrm{d}y \tag{A1e}$$

$$\leq 2\int_{y_\epsilon-\mu_J}^{\infty} (y+y) \frac{1}{\sqrt{2\pi}\sigma_J} \exp(-\frac{y^2}{2\sigma_J^2})\,\mathrm{d}y \tag{A1f}$$

$$= \frac{4\sigma_J}{\sqrt{2\pi}} \exp(-\frac{(y_\epsilon-\mu_J)^2}{2\sigma_J^2}) \tag{A1g}$$

$$= 2\sigma_J^2\epsilon. \tag{A1h}$$

Thus, by using (A1g) and (A1h),

$$y_\epsilon = \sqrt{-2\sigma_J^2 \log(\epsilon\sigma_J\sqrt{2\pi}/2)} + \mu_J \tag{A2}$$

. We use the aforementioned arguments to determine the finite computational range $[y_{-\epsilon}, y_\epsilon]$ in the Kou model. We carry out the reasoning for the positive semi-axis (the reasoning is similar to that for the negative semi-axis) and set $k(y) = p\alpha_1 e^{-\alpha_1 y}$ for $y \geq 0$ $\big((1-p)\alpha_2 e^{\alpha_2 x}$ for $y < 0\big)$. Then, y_ϵ can be determined by the following equations:

$$\left|\int_0^\infty (u(x+y)-u(x))\lambda f(y)\,\mathrm{d}y - \int_0^{y_\epsilon} (u(x+y)-u(x))\lambda f(y)\,\mathrm{d}y\right|$$

$$\leq L\left|\int_0^\infty (x+y-x)\lambda f(y)\,\mathrm{d}y - \int_0^{y_\epsilon} (x+y-x)\lambda f(y)\,\mathrm{d}y\right| \tag{A3a}$$

$$\leq L\int_{y_\epsilon}^\infty |y| f(y)\,\mathrm{d}y \tag{A3b}$$

$$= \int_{y_\epsilon}^\infty |y| p\alpha_1 e^{-\alpha_1 y}\,\mathrm{d}y \tag{A3c}$$

$$= p\alpha_1 e^{-y_\epsilon \alpha_1}\left(\frac{1}{\alpha_1^2} + \frac{y_\epsilon}{\alpha_1}\right) \tag{A3d}$$

[27, equation 3.351]

$$= \frac{p}{\alpha_1} e^{-y_\epsilon \alpha_1}(1 + y_\epsilon \alpha_1) \tag{A3e}$$

$$\leq \frac{p}{\alpha_1} e^{-y_\epsilon \alpha_1}\alpha_1 e^{y_\epsilon} \tag{A3f}$$

$$= p e^{y_\epsilon(1-\alpha_1)} \tag{A3g}$$

$$= \epsilon, \tag{A3h}$$

resulting in

$$y_\epsilon = \log(\epsilon/p)/(1-\alpha_1). \tag{A4}$$

Similar arguments can be applied to $y < 0$. Thus,

$$y_{-\epsilon} = -\log\big(\epsilon/(1-p)\big)/(1-\alpha_2). \tag{A5}$$

References

[1] A. Almendra, *Numerical valuation of American options under the CGMY process*, in *Exotic Option Pricing and Advanced Lévy Models*, W. Schoutens, A. Kyprianou, and P. Wilmott, eds., Wiley, UK, 2004.

[2] A. Almendral and C. Oosterlee, *Numerical valuation of options with jumps in the underlying*, Applied Numerical Mathematics 53 (2005), pp. 1 – 18.

[3] A. Almendral and C.W. Oosterlee, *On American options under the Variance Gamma process*, Tech. rep., Delft University of Technology, 2004, to appear in Applied Mathematical Finance (not yet available).

[4] ———, *Highly accurate evaluation of European and American options under the Variance Gamma process*, Journal of Computational Finance 10 (2006), pp. 21–42.

[5] ———, *Accurate evaluation of European and American options under the CGMY process*, SIAM Journal on Scientific Computing 29 (2007), pp. 93–117.

[6] L. Andersen and J. Andreasen, *Jump-diffusion processes: Volalitility smile fitting and numerical methods for option pricing*, Review of Derivatives Research 4 (2000), pp. 231 – 262.

[7] L. Bos and K. Salkauskas, *On the matrix* $[|x_i - x_j|^3]$ *and the cubic spline continuity equations*, Journal of Approximation Theory 51 (1987), pp. 81 – 88.

[8] S.I. Boyarchenko and S.Z. Levendorskiĭ, *Non-Gaussian Merton-Black-Scholes Theory*, *Advanced Series on Statistical Science & Applied Probability*, vol. 9, World Scientific Publishing Co. Inc., River Edge, NJ (2002).

[9] M. Briani, R. Natalini, and G. Russo, *Implicit-explict numerical schemes for Jump-diffusion processes*, Calcolo 44 (2007), pp. 33 – 57.

[10] D. Brown, L. Ling, E. Kansa, and J. Levesley, *On approximate cardinal preconditioning methods for solving PDEs with radial basis functions*, Engineering Analysis with Boundary Elements 29 (2005), pp. 343–353.

[11] R. Brummelhuis and R.T. Chan, *An RBF scheme for option pricing in exponential lévy models*, Tech. rep., Birkbeck, University of London, 2011, sumitted to Journal of Applied Mathematical Finance.
[12] P. Carr and D.B. Madan, *Option valuation using the Fast Fourier Transform*, Journal of Computational Finance 2 (1999), pp. 61–73.
[13] P. Carr, D.B. Madan, and E.C. Chang, *The Variance Gamma process and option pricing*, European Finance Review 2 (1998), pp. 79–105.
[14] P. Carr and A. Mayo, *On the numerical evaluation of option prices in Jump Diffusion processes*, The European Journal of Finance 13 (2007), pp. 353 – 372.
[15] P. Carr, H. Geman, D.B. Madan, and M. Yor, *The fine structure of asset returns: An empirical investigation*, Journal of Business 75 (2002), pp. 305–332.
[16] R. Cont and P. Tankov, *Financial Modelling With Jump Processes*, Chapman & Hall/CRC Financial Mathematics Series, Chapman & Hall/CRC, Boca Raton, Fla., London (2004).
[17] R. Cont and E. Voltchkova, *A finite difference scheme for option pricing in jump diffusion and exponential Lévy models*, SIAM Journal on Numerical Analysis 43 (2005), pp. 1596 – 1626.
[18] Y. d'Halluin, P. Forsyth, and K. Vetzalz, *Robust numerical methods for contingent claims under Jump Duffusion process*, IMA J. Num. Anal. 25 (2005), pp. 87 – 112.
[19] Y. d'Halluin, P.A. Forsyth, and G. Labahn, *A penalty method for American options with Jump Diffusion processes*, Numerische Mathematik 97 (2004), pp. 321 – 352.
[20] T. Driscoll and B. Fornberg, *Interpolation in the limit of increasingly flat radial basis functions*, Comput. Math. Appl. 43 (2002), pp. 413 – 422.
[21] G.E. Fasshauer, *Meshfree Approximation Methods with MATLAB*, *Interdisciplinary Mathematical Sciences*, vol. 6, World Scientific Sciences, Hackensack, N.J. (2007).
[22] G.E. Fasshauer and M.J. Mccourt, *Stable evaluation of gaussian rbf interpolants* (2010), http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.188.7442.
[23] G.E. Fausshauer, A.Q.M. Khaliq, and D.A. Voss, in *Proceedings of A Parallel Time Stepping Approach Using Meshfree Approximations for Pricing Options with Non-smooth Payoffs*, July, Third World Congress of the Bachelier Finance Society, Chicago, 2004.
[24] ———, *Using meshfree approximation for multi-asset American option problems*, J. Chinese Institute Engineers 27 (2004), pp. 563 – 571.
[25] B. Fornberg and G. Wright, *Stable computation of multiquadric interpolants for all values of the shape parameter*, Comput. Math. Appl. 47 (2004), pp. 497–523.
[26] M. Giles and R. Carter, *Convergence analysis of Crank-Nicolson and Rannacher time-marching*, Journal of Computational Finance 9 (2006), pp. 89–112.
[27] I.S. Gradshteyn and I. Ryzhik, *Table of Integrals, Series, and Products*, 5th ed., Academic Press, Inc., London (1994).
[28] A. Hirsa and D.B. Madan, *Pricing American options under Variance Gamma*, Journal of Computational Finance 7 (2004).
[29] Y.C. Hon and X.Z. Mao, *A radial basis function method for solving options pricing model*, Financial Engineering 8 (1999), pp. 31–49.
[30] A. Iserles, *A First Course in the Numerical Analysis of Differential Equations*, Cambridge Texts in Applied Mathematics (2009).
[31] K.R. Jackson, J. Sebastian, and S. Vladimir, *Fourier space time-stepping for option pricing with Lévy models*, the Journal of Computational Finance 12 (2008), pp. 1–29.
[32] E.J. Kansa, *Multiquadrics - a scattered data approximation scheme with applications to computational fluid dynamics - i.surface approximations and partial derivatives estimates*, Comput. Math. Appl. 19 (1990), pp. 127–145.
[33] ———, *Multiquadricsa scattered data approximation scheme with applications to computation fluid dynamics: Ii. solution to parabolic, hyperbolic and elliptic partial differential equations*, Comput. Math. Appl. 19 (1990), pp. 147–161.
[34] S.G. Kou, *A jump diffusion model for option pricing*, Management Science 48 (2002), pp. 1086–1101.
[35] S.G. Kou and H. Wang, *Option pricing under a double exponential jump diffusion model*, Tech. rep., Columbia University, 2001, working paper.
[36] E. Larsson, K. Åhlander, and A. Hall, *Multi-dimensional option pricing using radial basis functions and the Generalized Fourier Transform*, Journal of Computational and Applied Mathematics 222 (2008), pp. 175–192.
[37] E. Larsson and B. Fornberg, *Theorectical and computational aspects of multivariate interpolation with increasingly flat radial basis functions*, Comput. Math. Appl. 49 (2005), pp. 103 – 130.
[38] A.L. Lewis, *A simple option formula for general Jump-diffusion and other exponential Lévy processes* (2001), http://www.optioncity.net/pubs/ExpLevy.pdf.
[39] L. Ling and E. Kansa, *Preconditioning for radial basis functions with domain decomposition methods*, Math. and Compt. Modelling 40 (2004), pp. 1413–1427.
[40] ———, *A least-squares preconditioner for radial basis functions collocation methods*, Adv. in Comput. Math 23 (2005), pp. 31–54.
[41] D.B. Madan and F. Milne, *Option pricing with V. G. martingale components*, Mathematical Finance 1 (1991), pp. 39–55.
[42] A.M. Matache, P.A. Nitsche, and C. Schwab, *Wavelet galerkin pricing of American options on Lévy driven assets*, 2003. working paper.
[43] A.M. Matache, C. Schwab, and T.P. Wihler, *Fast numerical solution of parabolic integrodifferential equations with applications in finance*, SIAM Journal on Scientific Computing 27 (2005), pp. 369–393.
[44] R.C. Merton, *Option pricing when underlying stock returns are discontinuous.*, Journal of Financial Economics 3 (1976), pp. 125–144.
[45] U. Pettersson, E. Larsson, G. Marcusson, and J. Persson, *Improved radial basis function methods for multi-dimensional option pricing*, Journal of Computational and Applied Mathematics 222 (2008), pp. 82–93.
[46] H. Pham, *Optimal stopping, free boundary, and American option in a Jump-diffusion model*, Appl.

Math. Optim. 35 (1997), pp. 125 – 144.

[47] K.I. Sato, *Lévy Processes and Infinitely Divisible Distributions*, Cambridge University Press, Cambridge, U.K., New York (1999).

[48] W. Schoutens, *Lévy Processes in Finance : Pricing Financial Derivatives*, Wiley Series in Probability and Mathematical Statistics, Chichester : Wiley (2003).

[49] ———, *Exotic options under Lévy models: An overview*, Journal of Computational and Applied Mathematics 189 (2006), pp. 526 – 538.

[50] L.F. Shampine, *Vectorized adaptive quadrature in matlab*, Journal of Computational and Applied Mathematics 211 (2008), pp. 131 – 140.

[51] P. Tankov and E. Voltchkova, *Jump-diffusion models: A practitioners guide* (2009), `http://people.math.jussieu.fr/~tankov/tankov_voltchkova.pdf`.

[52] I.R. Wang, J.W.L. Wan, and P.A. Forsyth, *Robust numerical valuation of European and American options under the CGMY process*, Journal of Computational Finance 10 (2007), pp. 31–69.

[53] H. Wendland, *Scattered Data Approximation,*, *Cambridge Monographs on Applied and Computational Mathematics*, vol. 17, Cambridge University Press, Cambridge (2005).