Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

RBF-PU method for pricing options under the jump–diffusion model with local volatility

Reza Mollapourasl [a,b,*], Ali Fereshtian [a], Hengguang Li [c], Xun Lu [d]

[a] School of Mathematics, Shahid Rajaee Teacher Training University, Lavizan, Tehran 16788, Iran
[b] Department of Mathematics, Oregon State University, Corvallis, OR 97331, USA
[c] Department of Mathematics, Wayne State University, Detroit, MI 48202, USA
[d] School of Mathematics and Computational Science, Xiangtan University, Hunan Province, Xiangtan 411105, PR China

ARTICLE INFO

Article history:
Received 3 October 2017

Keywords:
Radial basis functions
Partition of unity
Option pricing
Jump–diffusion
Merton and Kou models

ABSTRACT

Meshfree methods based on radial basis functions (RBFs) are of general interest for solving partial differential equations (PDEs) because they can provide high order or spectral convergence for smooth solutions in complex geometries. For global RBF methods, one of the major disadvantages is the computational cost associated with the dense linear systems that arise. Therefore, this paper is currently directed toward localized RBF approximations known as the RBF partition of unity (RBF-PU) method for partial integro-differential equation (PIDE) arisen in option pricing problems in jump–diffusion model. RBF-PU method produces algebraic systems with sparse matrices which have small condition number. Also, for comparison, some stable time discretization schemes are combined with the operator splitting method to get a fully discrete problem. Numerical examples are presented to illustrate the convergence and stability of the proposed algorithms for pricing European and American options with Merton and Kou models.

Published by Elsevier B.V.

1. Introduction

The Black–Scholes [1] and stochastic volatility models [2] are examples of diffusion models, where the sample paths of the process are continuous. Stochastic volatility models are able to generate heavy tails in the return distribution for larger time intervals, which is a result of the accumulation of small moves over a sufficiently long time. However, large sudden changes under diffusion models are next to impossible. The addition of jumps into the model generates heavy tails in returns for short time intervals, and allows large sudden changes in the underlying asset. This is particularly important from a risk management perspective, since the implication is that large losses are possible even in a short time interval. Similarly, as in the case of stochastic volatility, markets under jump–diffusion models are incomplete.

It is recognized that the assumption of log normal stock diffusion with constant volatility in derivation of Black–Scholes model for pricing option is not consistent with real stock price behavior. Jumps are regularly observed in the discrete movement of stock price and these jumps cannot be captured by the log normal distribution characteristic of the stock price in the Black–Scholes model. Therefore an alternative model is necessary to overcome these issues. To resolve these issues several models have been proposed in the literature. Among these, the jump– diffusion model introduced by Merton [3] and Kou [4] is one of the most used model. Merton proposed a log-normally distributed process for the jump-amplitudes, while Kou suggested logdouble-exponentially distributed process. These models have finite jump activity, unlike the more general

[*] Corresponding author at: School of Mathematics, Shahid Rajaee Teacher Training University, Lavizan, Tehran 16788, Iran.
E-mail address: mollapour@srttu.edu (R. Mollapourasl).

https://doi.org/10.1016/j.cam.2018.01.002
0377-0427/Published by Elsevier B.V.

approach with possibly infinite jump activity proposed in [5]. Another approach is to consider stochastic volatility models with jumps. The model proposed by Bates in [6] with jumps only in the value of the underlying asset is an example of such an approach. More general jump–diffusion models with stochastic volatility are considered in [7], for example.

In a pricing model where the underlying asset follows a Markov process with infinitesimal generator $\mathcal{L}$, the value $V(t, S)$ as a function of the time t and the underlying asset price S solves Kolmogorov backward equation

$$\frac{\partial V(t,S)}{\partial t} + \mathcal{L}V(t,S) = 0 \tag{1}$$

with appropriate boundary conditions which describe the payoff of the option. In the case of the Black–Scholes model, this pricing equation reduces to the Black–Scholes partial differential equation whose analytical solution leads to the famous Black–Scholes formula. When the random evolution of the underlying asset is driven by a Lévy process or more generally a time inhomogeneous jump–diffusion process, the operator $\mathcal{L}$ is an integro-differential operator, expressed as the sum of a second-order differential operator and an integral operator, and Eq. (1) becomes a partial integro-differential equation [8].

Except in the Black–Scholes model with constant volatility, solutions to option pricing problems are not known analytically in general and require numerical methods. A variety of techniques have been proposed to solve pricing equations. In the financial literature several numerical approaches have been proposed for pricing options under both models with stochastic volatility and models with jumps. In particular, in the case of models with jumps, we can mention the paper by Matache et al. [9], where a Galerkin discretization in logarithmic price using a wavelet basis is presented. Also, in [10], they develop an alternating direction implicit (ADI) finite difference method that is shown to be unconditionally stable and, if combined with Fast Fourier Transform (FFT) methods, the method is computationally efficient. An explicit–implicit finite difference scheme is used in [11] to price European and barrier options in jump–diffusion and exponential Lévy Models. In the paper by Forsyth et al. [12], an implicit discretization method is presented for pricing such American options. The jump– diffusion correlation integral term is computed using an iterative method coupled with an FFT while the American constraint is imposed by using a penalty method. In papers by Fang and Oosterlee [13,14], they present a pricing method based on Fourier-cosine expansions for American and European options.

Other numerical methods which we can mention are for examples in [15] authors proposed an unconditionally stable ADI method for its solution. Also, in [16] a fitted finite volume method for jump–diffusion process has been employed. Their method is based on fitted finite volume method spatial discretization and Crank– Nicolson scheme for temporal discretization was developed. Recently in [17,18] authors developed an efficient method for pricing Merton jump–diffusion option, combining the spectral domain decomposition method and the Laplace transform method.

In models with jumps, the non local integral term leads to dense matrices after discretization [11] and efficient numerical methods are required for pricing of complex contracts and for calibration of model parameters. This fact means that fully implicit schemes encounter difficulties such as dense matrix inversions whereas fully explicit schemes impose stability restrictions [19]. For some solutions to this problem in [20,21] authors have used an iterative procedure for solving the discretized equations, and more efficient approaches using implicit–explicit Runge–Kutta schemes which treat the integral term explicitly were later proposed by Cont and Voltchkova in [11] and by Briani et al. [19].

Meshfree methods based on radial basis functions (RBFs) are widely used for solving PDEs because they can provide spectral convergence for smooth solutions in complex geometries. In [22,23], meshfree methods based on RBF approximation have been shown to perform better than finite difference methods for option pricing problems in one and two spatial dimensions. Similar problems have also been solved in [24]. Forward Kolmogorov problems have been solved in [25,26] with promising results. However, all of these papers employ global RBF collocation methods, leading to dense linear systems, and computational costs that become prohibitive as the number of dimensions increase. In order to address the computational cost issues of the global RBF method, we need to introduce locality. An easy way to do that is to use compactly supported RBFs, such as the Wendland functions [27], but then the spectral convergence properties are lost.

Here we take another approach, where the infinitely smooth RBFs are still used in the approximation but over local subregions of the computational domain. The possibility of using RBFs in a partition of unity scheme was mentioned in [28], further discussed in [29], implemented for interpolation on the sphere, in the plane, and in three-dimensional space in [30,31]. RBF-PU method is considered for parabolic PDEs of convection–diffusion type in [32], and stability and accuracy of RBF-PU method are investigated partly theoretically and partly numerically. In [33], we price American options under Heston's stochastic volatility model using RBF-PU method applied to a linear complementary formulation of the free boundary partial differential equation problem. Also, for pricing options under regime switching model, we applied local RBF method based on finite difference method, and derived an stable and well-conditioned discretized system of equations in [34].

In this paper, we develop a RBF-PU based algorithm for numerically pricing European and American options under jump– diffusion model and study on the accuracy and efficiency of the proposed algorithm. After space discretization by using RBF-PU approximation, for European option, Crank–Nicolson, Leap-Frog (CNLF), Crank–Nicolson, Adams–Bashforth (CNAB) and Backward-difference formula of second order (BDF-2) schemes are employed for time discretization, also for American option these time step schemes are combined with an operator splitting technique [35] which is applied to the linear complementarity problem (LCP). In Section 2, a partial integro-differential equation for pricing European options and a linear complementary problem for pricing the American options are introduced. In Sections 3 and 4, the basics of radial basis functions, the partition of unity method based on RBFs and the convergence results of RBF-PU approximation are presented.

In Section 5, differential and integral operators are introduced to our problem and their discretization are given. In Section 6, time discretization with CNLF, CNAB, and BDF-2 schemes are applied to European options, and these schemes are combined with the operator splitting method for evaluation of American options. The stability of these schemes are also discussed. Finally, in Section 7, the accuracy and efficiency of the proposed method are numerically investigated for European and American put options.

2. Option pricing model

Let $(S_t)_{t\in[0,T]}$ be the stock price process in financial market on a probability space $(\Omega, \mathcal{F}, \mathcal{F}_t, \mathbb{P})$ with filtration $\mathcal{F}_t$. Under the assumption of no-arbitrage there exists a measure $\mathbb{Q}$ equivalent to $\mathbb{P}$ under which $(e^{-rt}S_t)$ is a martingale where r is interest rate. In exponential jump–diffusion models, the stock price process of (S_t) under $\mathbb{Q}$ is represented as the exponential of a jump–diffusion process $S_t = S_0 \exp(rt + X_t)$ where X_t is a jump–diffusion process defined by

$$X_t = bt + \sigma W_t + \sum_{i=1}^{N_t} Y_i,$$

where b and $\sigma > 0$ are real constants, Y_i are independent and identically distributed random variables, and W_t, N_t, and Y_i are mutually independent.

Now, let V_t be a European put option price process under the risk-neutral measure $\mathbb{Q}$ defined by

$$V_t = \mathbb{E}[e^{-r(T-t)}G(S_T)|\mathcal{F}_t],$$

where $\mathbb{E}[\cdot]$ is the expectation operator and $G(S_T) = \max(K - S_T, 0)$ is a payoff function, K is the strike price, and T is the time of maturity. By the Markov property, we have the identity $V_t = V(t, S_t)$, where

$$V(t, S_t) = \mathbb{E}[e^{-r(T-t)}G(S_T)|S_t = S],$$

and leads to PIDE

$$\begin{aligned}&\frac{\partial V}{\partial t}(t,S) + \frac{1}{2}(\sigma(t,S)S)^2\frac{\partial^2 V}{\partial S^2}(t,S) + rS\frac{\partial V}{\partial S}(t,S) - rV(t,S)\\ &+\lambda\int_{-\infty}^{\infty}\left[V(t,Se^y) - V(t,S) - S(e^y-1)\frac{\partial V}{\partial S}(t,S)\right]f(y)dy = 0\end{aligned}$$

on $[0, T) \times (0, \infty)$, where $\sigma(t, S)$ is the known local volatility function, and PIDE satisfies the end condition

$$V(T, S) = G(S), \quad S > 0.$$

By using transformations $S = Ke^x$ and $t = T - \tau$, and let $u(\tau, x) = V(T - \tau, Ke^x)$ and $\tilde{\sigma}(\tau, x) = \sigma(T - \tau, Ke^x)$ so computation of the option value requires solving the following PIDE

$$\frac{\partial u}{\partial \tau} = \mathcal{L}u = \frac{1}{2}\tilde{\sigma}^2\frac{\partial^2 u}{\partial x^2} + (r - \frac{\tilde{\sigma}^2}{2} - \lambda\kappa)\frac{\partial u}{\partial x} - (r+\lambda)u + \lambda\int_{-\infty}^{\infty} u(\tau,y)f(y-x)dy \tag{2}$$

on $(0, T] \times (-\infty, \infty)$ with constant coefficients, and the initial condition

$$u(0, x) = H(x), \quad for \;\; x \in (-\infty, \infty) \tag{3}$$

where $H(x) = G(Ke^x)$, λ is the intensity of jumps, $\kappa = \int_{-\infty}^{\infty}(e^y - 1)f(y)dy$, and $f(y)$ is a density function.

In this paper, we focus on two popular jump–diffusion models with finite activity, Merton [3] and Kou [4] model. The jump size in Merton model follows a lognormal distribution and for Kou model jump size follows a double exponential distribution. Hence, the density function and corresponding mean are respectively given by

$$\text{Merton:}\;\; f(y) = \frac{1}{\gamma\sqrt{2\pi}}\exp\left(-\frac{(y-\mu)^2}{2\gamma^2}\right) \qquad \kappa = \exp(\mu + \frac{\gamma^2}{2}) - 1, \tag{4}$$

$$\text{Kou:}\;\; f(y) = p\alpha_1 e^{-\alpha_1 y}\mathcal{H}(y) + (1-p)\alpha_2 e^{\alpha_2 y}\mathcal{H}(-y) \qquad \kappa = p\frac{\alpha_1}{\alpha_1 - 1} + (1-p)\frac{\alpha_2}{\alpha_2+1} - 1, \tag{5}$$

with $\gamma > 0, \mu \in \mathbb{R}, \alpha_1 > 1, \alpha_2 > 0, 0 < p < 1$ and $\mathcal{H}(\cdot)$ is the Heaviside function.

The asymptotic behavior of the European put option is defined by

$$\lim_{x\to-\infty}[u(\tau,x) - (Ke^{-rt} - Ke^x)] = 0, \quad \lim_{x\to\infty} u(\tau,x) = 0.$$

In the following, we will give a PIDE formulation to price an American option under jump–diffusion model. An American option has the early exercise feature, so the optimal exercise boundary is a free boundary and separates the stopping and

Table 1
Some well-known functions that generate globally supported RBFs.

Function name	Definition
Gaussian (GA)	$\exp(-\epsilon^2 r^2)$
Multiquadrics (MQ)	$\sqrt{1+(\epsilon r)^2}$
Inverse multiquadrics (IMQ)	$\frac{1}{\sqrt{1+(\epsilon r)^2}}$
Conical splines	r^{2k+1}
Thin plate splines (TPS)	$(-1)^{k+1} r^{2k}\log(r)$

continuation region. Let $V(t,S)$ denote the fair value of an American option at time t if the asset price at that time is $S_t = S$, so $V(S,t)$ satisfy the following free boundary value problem

$$\begin{cases} \dfrac{\partial V}{\partial t}(t,S) + \dfrac{1}{2}(\sigma(t,S)S)^2 \dfrac{\partial^2 V}{\partial S^2}(t,S) + rS\dfrac{\partial V}{\partial S}(t,S) - rV(t,S) & S > S_f(t) \\ +\lambda \displaystyle\int_{-\infty}^{\infty} \left[V(t, Se^y) - V(t,S) - S(e^y - 1)\dfrac{\partial V}{\partial S}(t,S) \right] f(y)dy = 0, & \\ V(t,S) = G(S), & 0 \le S \le S_f(t) \end{cases} \tag{6}$$

where $G(S)$ is the payoff function, and $S_f(t)$ denotes as an unknown free moving exercise boundary of the option. Similar to European case, by using transformations $S = Ke^x$ and $t = T - \tau$, and let $u(\tau,x) = V(T-\tau, Ke^x)$ and $\tilde{\sigma}(\tau,x) = \sigma(T-\tau, Ke^x)$, so solving (6) and computation of the American option value requires solving the following linear complementary problem (LCP) [36]

$$\begin{cases} u_\tau(\tau,x) - \mathcal{L}u(\tau,x) \ge 0, \\ u(\tau,x) \ge H(x), \\ (u_t(\tau,x) - \mathcal{L}u(\tau,x))(u(\tau,x) - H(x)) = 0, \end{cases} \tag{7}$$

for all $(\tau,x) \in (0,T] \times (-\infty,\infty)$, and $\mathcal{L}u(\tau,x)$ and $H(x)$ are defined by (2) and (3).

In this case, we impose in addition the following initial and boundary conditions

$$u(0,x) = H(x). \tag{8}$$

$$\lim_{x\to-\infty} u(\tau,x) = K, \quad \lim_{x\to\infty} u(\tau,x) = 0. \tag{9}$$

3. RBF approximation

To avoid mesh generation, in recent years meshless techniques have attracted the attention of researchers. In a meshless method a set of scattered nodes is used instead of meshing the domain of the problem. The technique of RBFs is one of the most recently developed meshless methods that has attracted attention of many researchers in recent years. No need for a mesh or triangulation, simple implementation of boundary conditions and dimension independence are the main advantages of RBF methods.

Let $\phi : \Omega \times \Omega \to \mathbb{R}$ be a kernel with the property $\phi(\mathbf{x},\mathbf{y}) := \phi(\|\mathbf{x}-\mathbf{y}\|)$ for $\mathbf{x},\mathbf{y} \in \Omega$, and $\|\cdot\|$ is the Euclidean norm. Kernels with this property known as radial functions. In Table 1 some globally supported RBFs are listed which are commonly employed in the literature. The positive constant ϵ appearing in RBFs is called the shape parameter which dictates the flatness of the radial basis function and also has a key role on the convergence rate of the approximations and the condition number of the coefficient matrices. For more details about basic properties and types of radial basis functions, compactly and globally supported and also their wide applications in scattered date interpolations, the interested reader would be referred to recent works in this topic [37–40].

Now, let $\Omega \subset \mathbb{R}^d$ as a spatial domain, and consider a set of distinct points $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ in Ω. The RBF interpolant for a continuous target function $u : \Omega \to \mathbb{R}$ known at the nodes in X takes the form

$$s_u(\mathbf{x}) = \sum_{j=1}^{N} \lambda_j \phi(\|\mathbf{x} - \mathbf{x}_j\|). \tag{10}$$

The interpolation coefficients are determined by collocating the interpolant $s_u(\mathbf{x})$ to satisfy the interpolation condition $s_u(\mathbf{x}_i) = u(\mathbf{x}_i)$ for $i = 1, 2, \ldots, N$. This results in a symmetric system of linear equations

$$\mathbf{A}\lambda = \mathbf{u}, \tag{11}$$

where the elements of $\mathbf{A}$ are $a_{i,j} = \phi(\|\mathbf{x}_i - \mathbf{x}_j\|)$, $\lambda = [\lambda_1, \lambda_2, \ldots, \lambda_N]^\top$ (here $\top$ means transpose) and $\mathbf{u} = [u(\mathbf{x}_1), u(\mathbf{x}_2), \ldots, u(\mathbf{x}_N)]^\top$.

When the points in X are chosen to be distinct and ϕ is a positive definite or an order one conditionally positive definite on $\mathbb{R}^d$, the coefficient matrix $\mathbf{A}$ is guaranteed to be nonsingular, see [41]. For a nonsingular coefficient matrix $\mathbf{A}$, solving (11) for λ and substituting in (10) leads to

$$s_u(\mathbf{x}) = \Phi(\mathbf{x})\mathbf{A}^{-1}\mathbf{u} \tag{12}$$

with $\Phi(\mathbf{x}) = [\phi(\|\mathbf{x} - \mathbf{x}_1\|), \phi(\|\mathbf{x} - \mathbf{x}_2\|), \ldots, \phi(\|\mathbf{x} - \mathbf{x}_N\|)]$.

For simple implementation of the boundary conditions for PDEs, it is preferable to express the interpolation in Lagrange form, i.e., using cardinal basis functions. The cardinal basis functions, $\psi_j(\mathbf{x}), j = 1, 2, \ldots, N$, have the property

$$\psi_j(\mathbf{x}_i) = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \quad i = 1, 2, \ldots, N, \tag{13}$$

leading to the alternative formulation for the interpolant

$$s_u(\mathbf{x}) = \Psi(\mathbf{x})\mathbf{u}, \tag{14}$$

where $\Psi(\mathbf{x}) = [\psi_1(\mathbf{x}), \psi_2(\mathbf{x}), \ldots, \psi_N(\mathbf{x})]$.

Comparing (12) and (14), it is clear that the following relation holds between the cardinal basis and the original radial basis:

$$\Psi(\mathbf{x}) = \Phi(\mathbf{x})\mathbf{A}^{-1}. \tag{15}$$

When we approximate a time dependent function $u(t, \mathbf{x})$ that is a solution to a PDE problem, we let $\lambda_j, j = 1, 2, \ldots, N$, be time dependent such that

$$s_u(t, \mathbf{x}) = \sum_{j=1}^{N} \lambda_j(t)\phi(\|\mathbf{x} - \mathbf{x}_j\|), \quad \mathbf{x} \in \Omega, \ t \geq 0,$$

or, equivalently, when using the Lagrange form, we use the interpolant

$$s_u(t, \mathbf{x}) = \sum_{j=1}^{N} \psi_j(\mathbf{x})u_j(t) = \Phi(\mathbf{x})\mathbf{A}^{-1}\mathbf{u}(t). \tag{16}$$

By interpolating the initial condition (or a final condition) such that $\mathbf{u}(0) = [u(0, \mathbf{x}_1), u(0, \mathbf{x}_2), \ldots, u(0, \mathbf{x}_N)]^T$ we get $s_{u,X}(0, \mathbf{x}_k) = u(0, \mathbf{x}_k)$ for all k, while for $t > 0$ we have $\mathbf{u}(t) \approx [u(t, \mathbf{x}_1), u(t, \mathbf{x}_2), \ldots, u(t, \mathbf{x}_N)]^T$ and hence $s_{u,X}(t, \mathbf{x}_k) \approx u(t, \mathbf{x}_k)$ for all k.

4. Approximation based on RBF-PU method

Approximation based on global RBFs has many advantages but generally it leads to dense and ill conditioned coefficient matrices, hence complexity of computations is expensive. In this section we introduce RBF-PU method which is one of the local methods and put a set of local approximation spaces together to produce a conforming global approximation.

Let $\Omega \subset \mathbb{R}^d$ be a bounded set, and let a covering $\{\Omega_j\}_{j=1}^M$ of the region Ω such that $\Omega \subset \bigcup_{j=1}^M \Omega_j$. Also, we define

$$\forall \mathbf{x} \in \Omega \quad I(\mathbf{x}) := \{j \mid \mathbf{x} \in \Omega_j\}, \quad \mathrm{card}(I(\mathbf{x})) \leq K,$$

where the constant K is independent of the number of patches M.

Definition 1. Let $\Omega \subset \mathbb{R}^d$ be a bounded set. Let $\{\Omega_j\}_{j=1}^M$ be an open and bounded covering of Ω. This means all Ω_j are open and bounded and Ω is contained in their union. Set $\delta_j = diam(\Omega_j) = \sup_{\mathbf{x},\mathbf{y} \in \Omega_j} \|\mathbf{x} - \mathbf{y}\|_2$. A family of non-negative functions $\{w_j(\mathbf{x})\}_{j=1}^M$ with $w_j(\mathbf{x}) \in C^k(\mathbb{R}^d)$ is a k-stable partition of unity respect to cover of Ω_j if:

1. $supp(w_j) \subseteq \overline{\Omega}_j$
2. $\sum_j w_j(\mathbf{x}) = 1$ for $\mathbf{x} \in \Omega$
3. For every $\alpha \in \mathbb{N}_0^d$ with $|\alpha| \leq k$ there exist a constant $C_\alpha > 0$ such that for all $1 \leq j \leq M$

$$\|D^\alpha w_j\|_{L_\infty(\Omega_j)} \leq \frac{C_\alpha}{\delta_j^{|\alpha|}}$$

The weight functions w_j are constructed by using Shepard's method [42] as follow:

$$w_j(\mathbf{x}) = \frac{\phi_j(\mathbf{x})}{\sum_{k\in I(\mathbf{x})} \phi_k(\mathbf{x})}, \quad j = 1, 2, \dots, M \tag{17}$$

where $\phi_j(\mathbf{x})$ are compactly supported functions with support on Ω_j. To guarantee non-negativity and compact support in Ω_j, we define in (17)

$$\phi_j(\mathbf{x}) = \phi\left(\frac{\|\mathbf{x}-\mathbf{c}_j\|}{r_j}\right), \quad j = 1, 2, \dots, M, \tag{18}$$

where $\{\mathbf{c}_j\}_{j=1}^M$ and $\{r_j\}_{j=1}^M$ are the centers and radiuses of the circular, spherical or hyper-spherical patches $\{\Omega_j\}_{j=1}^M$ and where ϕ is one of the compactly supported functions with minimal degree described in [27, Corollary 9.14]. Here we consider the Wendland compact supported function [41]

$$\phi(r) = (1-r)_+^4(4r+1),$$

which belongs to C^2 for the construction of the weight functions.

The global approximation function $s_u(\mathbf{x})$, with $\mathbf{x} \in \Omega$, to the function $u(\mathbf{x})$ is constructed as

$$s_u(\mathbf{x}) = \sum_{j=1}^{M} w_j(\mathbf{x}) s_{u,j}(\mathbf{x}) = \sum_{j\in I(\mathbf{x})} w_j(\mathbf{x}) s_{u,j}(\mathbf{x}),$$

where $s_{u,j}$ are local interpolants such that $s_{u,j}(\mathbf{x}_i) = u(\mathbf{x}_i)$ for each node $\mathbf{x}_i \in \Omega_j$. Then, the global PU approximant inherits the interpolation property, i.e. $s_u(\mathbf{x}_i) = u(\mathbf{x}_i)$. Using the cardinal basis functions (13) the local interpolant $s_{u,j}(\mathbf{x})$ is an RBF-approximant of type (14) on Ω_j.

For time dependent function $u(t, \mathbf{x})$, we construct the global approximant built up from local RBF interpolants of type (16). For $j \in \{1, \dots, M\}$, let $J(\Omega_j) := \{k \mid \mathbf{x}_k \in \Omega_j\}$ be the set of indices of the node points that belong to the patch Ω_j. For such patch Ω_j, the local RBF approximation is given by

$$s_{u,j}(t, \mathbf{x}) = \sum_{k\in J(\Omega_j)} \psi_k(\mathbf{x}) u_k(t)$$

where $s_{u,j}(t, \mathbf{x}_k) = u_k(t)$ for all nodes $\mathbf{x}_k \in \Omega_j$ and ψ_k are cardinal basis functions. Hence, in the RBF-PUM, we obtain the global approximant for the time-dependent function $u(t, \mathbf{x})$

$$s_u(t, \mathbf{x}) = \sum_{j\in I(\mathbf{x})} w_j(\mathbf{x}) s_{u,j}(t, \mathbf{x}) = \sum_{j\in I(\mathbf{x})} \sum_{k\in J(\Omega_j)} w_j(\mathbf{x}) \psi_k(\mathbf{x}) u_k(t). \tag{19}$$

Now, for deriving an error estimation for partition of unity method, we need the following definitions

Definition 2. For a bounded region Ω fill distance on $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ define as following

$$h_{X,\Omega} = \sup_{\mathbf{x}\in\Omega} \min_{\mathbf{x}_j\in X} \|\mathbf{x}-\mathbf{x}_j\|_2,$$

which can be explained as largest distance that for every $\mathbf{x} \in \Omega$ there is one point $\mathbf{x}_j$ with this distance.

Definition 3. A subdomain $\Omega_j \subseteq \mathbb{R}^d$ satisfies an interior cone condition if there exists an angle $\theta \in (0, \frac{\pi}{2})$ and a radius $\gamma > 0$ such that, for all $\mathbf{x} \in \Omega_j$, a unit vector $\xi(\mathbf{x})$ exists such that the cone

$$\mathbf{C}(\mathbf{x}, \xi(\mathbf{x}), \theta, \gamma) = \{\mathbf{x} + \lambda\mathbf{y} : \mathbf{y} \in \mathbb{R}^d, \|\mathbf{y}\|_2 = 1, \mathbf{y}^T\xi(\mathbf{x}) \geq \cos(\theta), \lambda \in [0, \gamma]\}$$

is contained in Ω_j.

Definition 4. Suppose $\Omega \subseteq \mathbb{R}^d$ is bounded and $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\} \subseteq \mathbb{R}^d$ are given. An open and bounded covering $\{\Omega_j\}_{j=1}^M$ is called regular for (X, Ω) if the following properties are satisfied

1. For each $x \in \Omega$, the number of subdomains Ω_j, with $x \in \Omega_j$ is bounded by a global constant C,
2. Each subdomain Ω_j satisfies an interior cone condition,
3. The local fill distances $h_{X_{N_j},\Omega_j}$ are uniformly bounded by the global fill distance $h_{X,\Omega}$.

For each positive definite function $\phi \in C^k(\mathbb{R}^d)$ and each area $\Omega \subset \mathbb{R}^d$ there is a function space $\mathcal{N}_\phi(\Omega)$, the native Hilbert space [43]. The smoothness of ϕ inherited from the native space via $\mathcal{N}_\phi(\Omega) \subseteq C^{\lceil \frac{k}{2} \rceil}(\Omega)$. Also, for getting the full approximation order, the weak form of Holder continuity idea is used, and we define space $C_v^k(\mathbb{R}^d)$ as space of all functions such that their derivatives of order k satisfy $D^\alpha u(x) = \mathcal{O}(\|x\|_2^v)$ for $\|x\|_2 \to 0$. By using the above definitions the following convergence theorem is derived from [41].

Theorem 1. *Suppose $\phi \in C_v^k(\mathbb{R}^d)$ is conditionally positive definite of order m, also, let $\Omega \subseteq \mathbb{R}^d$ be open and bounded and $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\} \subseteq \Omega$. Let $\{\Omega_j\}_{j=1}^M$ be a regular covering for (Ω, X) and let $\{w_j\}_{j=1}^M$ be k-stable for $\{\Omega_j\}_{j=1}^M$. Then the error between $u \in \mathcal{N}_\phi(\Omega)$ and its partition of unity interpolant is bounded by*

$$|D^\alpha u(\mathbf{x}) - D^\alpha s_u(\mathbf{x})| \leq C_1 h_{X,\Omega}^{\frac{k+v}{2}-|\alpha|} |u|_{\mathcal{N}_\phi(\Omega)}$$

for all $\mathbf{x} \in \Omega$ and all $|\alpha| \leq \frac{k}{2}$.

5. Spatial discretization

The first step in this direction is to split the PIDE operator $\mathcal{L}$ into two parts as follows

$$\mathcal{L}u(\tau, x) = \mathcal{D}u(\tau, x) + \lambda(\mathcal{I}u(\tau, x) - u(\tau, x)),$$

where $\mathcal{D}$ is a differential operator and $\mathcal{I}$ is an integral operator defined by

$$\mathcal{D}u(\tau, x) = \frac{1}{2}\tilde{\sigma}^2 \frac{\partial^2 u}{\partial x^2} + (r - \frac{\tilde{\sigma}^2}{2} - \lambda\kappa)\frac{\partial u}{\partial x} - ru$$

$$\mathcal{I}u(\tau, x) = \int_{-\infty}^{\infty} u(\tau, y)f(y - x)dy.$$

To use the RBF-PU approximation for solving a PIDE problem (2), we need to compute the effect of applying a spatial differential operator $\mathcal{D}$ at the interior node points. Using Leibniz's rule, a derivative term of order α in the differential operator can be applied to the global approximation (19) as

$$\begin{aligned}\frac{\partial^{|\alpha|}}{\partial x^\alpha} u(\tau, x) &= \sum_{i \in I(x)} \sum_{k \in J(\Omega_i)} \frac{\partial^{|\alpha|}}{\partial x^\alpha}\left[w_i(x)\psi_k(x)\right] u_k(\tau) \\ &= \sum_{i \in I(x)} \sum_{k \in J(\Omega_i)} \left[\sum_{\beta \leq \alpha} \binom{\alpha}{\beta} \frac{\partial^{|\alpha-\beta|}}{\partial x^{\alpha-\beta}} w_i(x) \frac{\partial^{|\beta|}}{\partial x^\beta}\psi_k(x)\right] u_k(\tau). \end{aligned} \tag{20}$$

For composite linear operators like $\mathcal{D}$, we sum up the contributions from each term, and denote the global differentiation matrix under operator $\mathcal{D}$ by $\mathbf{D}$.

To approximate the integral operator $\mathcal{I}$ numerically, we replace the unbounded domain $(-\infty, \infty)$ for x with a bounded one $\Omega = [x_{\min}, x_{\max}]$ where the values $x_{\min}$ and $x_{\max}$ will be chosen based on standard financial arguments, and zero belongs to $[x_{\min}, x_{\max}]$. So, we divide the integral into two parts on Ω and $\mathbb{R} \setminus \Omega$, so we have

$$\mathcal{I}u(\tau, x) = \int_\Omega u(\tau, y)f(y - x)dy + \int_{\mathbb{R}\setminus\Omega} u(\tau, y)f(y - x)dy. \tag{21}$$

We define $R(\tau, x) = \int_{\mathbb{R}\setminus\Omega} u(\tau, y)f(y-x)dy$ and by using the boundary conditions for European put option we can calculate $R(\tau, x)$ by

$$R(\tau, x) = \int_{\mathbb{R}\setminus\Omega} (Ke^{-r\tau} - Ke^y)f(y - x)dy,$$

where for Merton model we have

$$R(\tau, x) = Ke^{-r\tau}\mathcal{N}(\frac{x_{\min} - x - \mu}{\gamma}) - Ke^{x+\mu+\frac{\gamma^2}{2}}\mathcal{N}\left(\frac{x_{\min} - x - \mu - \gamma^2}{\gamma}\right)$$

where $\mathcal{N}(\cdot)$ is the cumulative normal distribution, and for Kou model we have

$$R(\tau, x) = K(1 - p)\left(e^{-r\tau+\alpha_2(x_{\min}-x)} - \frac{\alpha_2}{\alpha_2 + 1}e^{-\alpha_2 x+(\alpha_2+1)x_{\min}}\right).$$

For American put option we can calculate $R(\tau, x)$ by

$$R(\tau, x) = \int_{\mathbb{R}\setminus\Omega} (K - Ke^y)f(y - x)dy,$$

where for Merton model we have

$$R(\tau, x) = K\mathcal{N}(\frac{x_{\min} - x - \mu}{\gamma}) - Ke^{x+\mu+\frac{\gamma^2}{2}}\mathcal{N}\left(\frac{x_{\min} - x - \mu - \gamma^2}{\gamma}\right)$$

and for Kou model we have

$$R(\tau,x)=K(1-p)\left(e^{\alpha_2(x_{\min}-x)}-\frac{\alpha_2}{\alpha_2+1}e^{-\alpha_2 x+(\alpha_2+1)x_{\min}}\right).$$

For approximating $\int_\Omega u(\tau,y)f(y-x)dy$ we use the classical numerical integration method known as trapezoidal rule by

$$\int_\Omega u(\tau,y)f(y-x)dy\approx\frac{\triangle y}{2}\left(\sum_{j=1}^{N_x}\omega_j u(\tau,y_j)f(y_j-x)\right)$$

where $y_j=j\triangle y$ with $\triangle y=\frac{x_{\max}-x_{\min}}{N_x}$ where N_x is the number of grids in x direction, and $\omega_j=1$ for $j=1,N_x$ and $\omega_j=2$ for $j=2,\dots,N_x-1$. Finally, by using approximate value for the first integral and an exact value for the second integral in (21), the integral matrix under operator $\mathcal{I}$ is derived and denoted by $\mathbb{I}$.

6. Time discretization

Let $\{0=\tau_0<\tau_1<\cdots<\tau_{M_\tau}=T,\delta\tau=\tau_m-\tau_{m-1},1\le m\le M_\tau\}$ as a partition for interval $[0,T]$ and $\{x_{\min}=x_1<x_2<\cdots<x_{N_x}=x_{\max}\}$ be partition of interval $\Omega=[x_{\min},x_{\max}]$. To evaluate an European option, let $\mathbf{U}_m:=[u(\tau_m,x_1),u(\tau_m,x_2),\dots,u(\tau_m,x_{N_x})]^\top$ (here $\top$ means transpose) as an approximate value of the solution which can be obtained by using the following time stepping schemes

- Crank–Nicolson, Leap-Frog (CNLF)
$$\frac{\mathbf{U}_{m+1}-\mathbf{U}_{m-1}}{2}=\delta\tau(\mathbf{D}-\lambda\mathbf{I})(\frac{\mathbf{U}_{m+1}+\mathbf{U}_{m-1}}{2})+\delta\tau\lambda\mathbb{I}\mathbf{U}_m \tag{22}$$
- Crank–Nicolson, Adams–Bashforth (CNAB)
$$\mathbf{U}_{m+1}-\mathbf{U}_m=\delta\tau(\mathbf{D}-\lambda\mathbf{I})(\frac{\mathbf{U}_{m+1}+\mathbf{U}_m}{2})+\delta\tau\lambda\mathbb{I}(\frac{3\mathbf{U}_m-\mathbf{U}_{m-1}}{2}) \tag{23}$$
- Backward-difference formula of second order (BDF-2)
$$\frac{3\mathbf{U}_{m+1}-4\mathbf{U}_m+\mathbf{U}_{m-1}}{2}=\delta\tau(\mathbf{D}-\lambda\mathbf{I})\mathbf{U}_{m+1}+\delta\tau\lambda\mathbb{I}(2\mathbf{U}_m-\mathbf{U}_{m-1}) \tag{24}$$

for $1\le m\le M_\tau-1$ where $\mathbf{D}$ is the differentiation matrix associated with the differential operator and $\mathbb{I}$ is integral matrix corresponding to the integral operator. In [44,45] stability of these schemes has been studied. Also, for $m=1$ the implicit Euler scheme is taken to avoid possible oscillations due to the non-smooth final value as the payoff function. In the following, we consider the stability analysis of implicit–explicit CNLF, CNAB and BDF-2 schemes derived from [46] for the linear test problem

$$u'(\tau)=\nu_B u(\tau)+\nu_C u(\tau)$$

where ν_B and ν_C are the complex eigenvalues of the explicit and implicit part of the scheme, respectively. By using the spatial discretization presented in the previous section for differential and integral operators, then we obtain a semi-discrete linear system of ODEs

$$\mathbf{u}'(\tau)=\mathbf{D}\mathbf{u}(\tau)+\mathbb{I}\mathbf{u}(\tau)-\lambda\mathbf{u}(\tau),\quad \tau\ge 0,$$

so, the following stability results are derived for the linear test problem.

Proposition 1. *If $\mathbb{I}$ is strictly positive, then for real and nonpositive eigenvalues of $\mathbf{D}$ the CNLF scheme is conditionally stable for $\lambda\delta\tau<1$.*

Proposition 2. *If $\mathbb{I}$ is strictly positive, then for real and nonpositive eigenvalues of $\mathbf{D}$ the CNAB scheme is conditionally stable for $\lambda\delta\tau<\frac{1}{2}$.*

Proposition 3. *If $\mathbb{I}$ is strictly positive, then for real and nonpositive eigenvalues of $\mathbf{D}$ the BDF-2 scheme is conditionally stable for $\lambda\delta\tau<\frac{2}{3}$.*

Under Merton and Kou models $\mathbb{I}$ is strictly positive for any quadrature with positive weights, so we have the following corollaries.

Corollary 1. *If eigenvalues of $\mathbf{D}$ are nonpositive, then the CNLF scheme is conditionally stable under Kou and Merton models for all $\lambda\delta\tau<1$.*

Corollary 2. *If eigenvalues of* **D** *are nonpositive, then the CNAB scheme is conditionally stable under Kou and Merton models for all* $\lambda\delta\tau < \frac{1}{2}$.

Corollary 3. *If eigenvalues of* **D** *are nonpositive, then the BDF-2 scheme is conditionally stable under Kou and Merton models for all* $\lambda\delta\tau < \frac{2}{3}$.

For RBF-PU discretization, we could not prove that eigenvalues of **D** are nonpositive or equivalently matrix **D** is an M-matrix which is known as a sufficient condition, so these are remain as the open problems, but in numerical experiments, we checked the nonpositivity of eigenvalues of **D** numerically for different presented discretization parameters, and derived that eigenvalues of **D** are nonpositive.

For American option, we use operator splitting method for solving LCP (7). The operator splitting method is introduced by Ikonen and Toivanen in [35] to evaluate the price of the American put option under the Black–Scholes model, and the method is studied by Toivanen in [47] under the Kou model. The operator splitting method is based on the formulation with an auxiliary variable Λ such that $u_\tau(\tau, x) - (\mathcal{D} - \lambda)u(\tau, x) - \lambda\mathcal{I}u(\tau, x) = \Lambda(\tau, x)$. Thus the LCP (7) for the American put option is reformulated as

$$\begin{cases} u_\tau(\tau, x) - (\mathcal{D} - \lambda)u(\tau, x) - \lambda\mathcal{I}u(\tau, x) = \Lambda(\tau, x), \quad \Lambda(\tau, x) \geq 0 \\ u(\tau, x) \geq H(x), \\ \Lambda(\tau, x)(u(\tau, x) - H(x)) = 0 \end{cases} \tag{25}$$

Similar as the European put option, we apply the RBF-PU method for spatial discretization of differential and integral operators in N_x collocation points $\{x_{\min} = x_1 < x_2 < \cdots < x_{N_x} = x_{\max}\}$, then corresponding differential and integral matrices **D** and $\mathbb{I}$ are derived. To evaluate an American put option price, we have to solve for $\mathbf{U}_m := [u(\tau_m, x_1), u(\tau_m, x_2), \ldots, u(\tau_m, x_{N_x})]^\top$ (here $\top$ means transpose) for $0 \leq m \leq M_\tau$ as a solution to the time stepping schemes

- Crank–Nicolson, Leap-Frog (CNLF)

$$\begin{cases} \dfrac{\widetilde{\mathbf{U}}_{m+1} - \mathbf{U}_{m-1}}{2} - \left(\delta\tau(\mathbf{D} - \lambda\mathbf{I})(\dfrac{\widetilde{\mathbf{U}}_{m+1} + \mathbf{U}_{m-1}}{2}) + \delta\tau\lambda\mathbb{I}\mathbf{U}_m\right) = \delta\tau\mathbf{\Lambda}_m, \\ \mathbf{\Lambda}_{m+1} = \mathbf{\Lambda}_m + \dfrac{\mathbf{U}_{m+1} - \widetilde{\mathbf{U}}_{m+1}}{2\delta\tau}, \\ \mathbf{\Lambda}_{m+1} \geq 0, \quad \mathbf{U}_{m+1} \geq \mathbf{H}, \quad (\mathbf{\Lambda}_{m+1})^\top(\mathbf{U}_{m+1} - \mathbf{H}) = 0 \end{cases} \tag{26}$$

- Crank–Nicolson, Adams–Bashforth (CNAB)

$$\begin{cases} \widetilde{\mathbf{U}}_{m+1} - \mathbf{U}_m - \left(\delta\tau(\mathbf{D} - \lambda\mathbf{I})(\dfrac{\widetilde{\mathbf{U}}_{m+1} + \mathbf{U}_m}{2}) + \delta\tau\lambda\mathbb{I}(\dfrac{3\mathbf{U}_m - \mathbf{U}_{m-1}}{2})\right) = \delta\tau\mathbf{\Lambda}_m, \\ \mathbf{\Lambda}_{m+1} = \mathbf{\Lambda}_m + \dfrac{\mathbf{U}_{m+1} - \widetilde{\mathbf{U}}_{m+1}}{2\delta\tau}, \\ \mathbf{\Lambda}_{m+1} \geq 0, \quad \mathbf{U}_{m+1} \geq \mathbf{H}, \quad (\mathbf{\Lambda}_{m+1})^\top(\mathbf{U}_{m+1} - \mathbf{H}) = 0 \end{cases} \tag{27}$$

- Backward-difference formula of second order (BDF-2)

$$\begin{cases} \dfrac{3\widetilde{\mathbf{U}}_{m+1} - 4\mathbf{U}_m + \mathbf{U}_{m-1}}{2} - \left(\delta\tau(\mathbf{D} - \lambda\mathbf{I})\widetilde{\mathbf{U}}_{m+1} + \delta\tau\lambda\mathbb{I}(2\mathbf{U}_m - \mathbf{U}_{m-1})\right) = \delta\tau\mathbf{\Lambda}_m, \\ \mathbf{\Lambda}_{m+1} = \mathbf{\Lambda}_m + \dfrac{\mathbf{U}_{m+1} - \widetilde{\mathbf{U}}_{m+1}}{2\delta\tau}, \\ \mathbf{\Lambda}_{m+1} \geq 0, \quad \mathbf{U}_{m+1} \geq \mathbf{H}, \quad (\mathbf{\Lambda}_{m+1})^\top(\mathbf{U}_{m+1} - \mathbf{H}) = 0 \end{cases} \tag{28}$$

where $\widetilde{\mathbf{U}}_{m+1}$ is an intermediate solution vector, $\mathbf{\Lambda}_m := [\Lambda(\tau_m, x_1), \Lambda(\tau_m, x_2), \ldots, \Lambda(\tau_m, x_{N_x})]^\top$ is the auxiliary function $\Lambda(\tau, x)$ evaluated at the discretization points and $\mathbf{H} := [H(x_1), H(x_2), \ldots, H(x_{N_x})]^\top$ with $H(\cdot)$ is the payoff function.

Each time step is split into two parts. Starting from the initial vector $\mathbf{U}_0 = \mathbf{H}$ and $\mathbf{\Lambda}_0 = \mathbf{0}$, first, the intermediate solution vector $\widetilde{\mathbf{U}}_{m+1}$ is solved from the modified system of linear equations

- Crank–Nicolson, Leap-Frog (CNLF)

$$\frac{\widetilde{\mathbf{U}}_{m+1} - \mathbf{U}_{m-1}}{2} - \left(\delta\tau(\mathbf{D} - \lambda\mathbf{I})(\frac{\widetilde{\mathbf{U}}_{m+1} + \mathbf{U}_{m-1}}{2}) + \delta\tau\lambda\mathbb{I}\mathbf{U}_m\right) = \delta\tau\mathbf{\Lambda}_m,$$

- Crank–Nicolson, Adams–Bashforth (CNAB)

$$\widetilde{\mathbf{U}}_{m+1} - \mathbf{U}_m - \left(\delta\tau(\mathbf{D} - \lambda\mathbf{I})(\frac{\widetilde{\mathbf{U}}_{m+1} + \mathbf{U}_m}{2}) + \delta\tau\lambda\mathbb{I}(\frac{3\mathbf{U}_m - \mathbf{U}_{m-1}}{2})\right) = \delta\tau\mathbf{\Lambda}_m,$$

- Backward-difference formula of second order (BDF-2)

$$\frac{3\widetilde{\mathbf{U}}_{m+1}-4\mathbf{U}_m+\mathbf{U}_{m-1}}{2}-\left(\delta\tau(\mathbf{D}-\lambda\mathbf{I})\widetilde{\mathbf{U}}_{m+1}+\delta\tau\lambda\mathbb{I}(2\mathbf{U}_m-\mathbf{U}_{m-1})\right)=\delta\tau\mathbf{\Lambda}_m.$$

Second, the intermediate solution $\widetilde{\mathbf{U}}_{m+1}$ is projected to be feasible, and $\mathbf{\Lambda}_m$ is updated by

$$\mathbf{\Lambda}_{m+1}=\mathbf{\Lambda}_m+\frac{\mathbf{U}_{m+1}-\widetilde{\mathbf{U}}_{m+1}}{2\delta\tau}$$

to satisfy

$$\mathbf{\Lambda}_{m+1}\geq 0,\quad \mathbf{U}_{m+1}\geq\mathbf{H},\quad (\mathbf{\Lambda}_{m+1})^{\top}(\mathbf{U}_{m+1}-\mathbf{H})=0.$$

The update step can be performed at each spatial grid point independently with the formulas

$$\mathbf{U}_{m+1}=\max(\mathbf{H},\widetilde{\mathbf{U}}_{m+1}-2\delta\tau\mathbf{\Lambda}_m),\quad \mathbf{\Lambda}_{m+1}=\max(\mathbf{\Lambda}_m+\frac{\mathbf{H}-\widetilde{\mathbf{U}}_{m+1}}{2\delta\tau},0).$$

7. Numerical results

In this section, we present numerical results to compute the prices of the American and European options using the RBF-PU method. Although the scheme works for all radial basis functions, we will use a positive definite one known as inverse multi-quadric radial basis function on different experimental setups. All computations are carried in MATLAB with a 3.6 GHz Corei3 processor. We will use the following measure for the error

$$\text{Error}:=\left|\frac{V(0,S)-V_{ref}(0,S)}{V_{ref}(0,S)}\right|$$

where $V(0,S)$ is the approximated value and $V_{ref}(0,S)$ is the reference value of option price. For European option under the Kou and Merton model, there exists an analytical closed form solution [4,3], so $V_{ref}(0,S)$ in the definition of Error refers to the exact solution, but for American option, analytical solution is not available, so $V_{ref}(0,S)$ refers to the numerical solutions used in literature as reference value with high mesh refinement. The jump distribution parameters are not changed between experiments. For Kou's model, the jump parameters are given by $\alpha_1=3.0465, \alpha_2=3.0775$ and $p=0.3445$, and for Merton's model are given by $\mu=-0.9$ and $\gamma=0.45$ in all Examples except Example 3.

Example 1. In this example, we consider European put option under Kou and Merton model with the following parameters

$$\sigma=0.15,\quad r=0.05,\quad T=0.25,\quad K=100,\quad \lambda=0.1.$$

These parameters are adopted from [20,48]. The reference value for European put option under Kou model is 9.430457 at $S=90$, 2.731259 at $S=100$, and 0.552363 at $S=110$. The reference values for European put option under Merton model are 9.285418 at $S=90$, 3.149026 at $S=100$, and 1.401186 at $S=110$.

In this example, the computation interval is $\Omega=[x_{\min},x_{\max}]=[-4,4]$, and numerical results including errors at different asset prices, cpu time in second and condition number of linear system of equations are shown in Tables 2 and 3 for different space N_x and time M_τ steps and different time discretization schemes Crank–Nicolson, Leap-Frog (CNLF), Crank–Nicolson, Adams–Bashforth (CNAB) and Backward-difference formula of second order (BDF-2). Regarding the results presented in Table 2 of [20], and Tables 7.1, 7.3 and 7.5 of [48], it is easy to derive that RBF-PU method combined with CNLF, CNAB and BDF-2 are faster and accurate than methods developed in [20,48]. Also, the option value, Delta and Gamma as the Greeks are plotted in Fig. 1.

Example 2. In this example, we consider American put option under Kou and Merton model with the following parameters

$$\sigma=0.15,\quad r=0.05,\quad T=0.25,\quad K=100,\quad \lambda=0.1.$$

These parameters are adopted from [49,47]. The reference value [49] for American put option under Kou model is 10.005071 at $S=90$, 2.807879 at $S=100$, and 0.561876 at $S=110$. The reference values [49] for American put option under Merton model are 10.003815 at $S=90$, 3.241215 at $S=100$, and 1.419796 at $S=110$.

The computation interval is $\Omega=[x_{\min},x_{\max}]=[-4,4]$, and numerical results including errors at different asset prices, cpu time in second and condition number of linear system of equations are shown in Tables 4 and 5 for different space N_x and time M_τ steps and different time discretization schemes. Numerical results for American options by these parameters are presented in Tables 3 and 6 in [49]. By comparing results, it is clear that RBF-PU method with CNLF is accurate than the iterative method developed in [49]. Also, the option value, Delta and Gamma as the Greeks are plotted in Fig. 2.

Table 2
Numerical results for European put option under the Kou model for Example 1.

M_τ	N_x	$S = 90$		$S = 100$		$S = 110$		Time (s)	cond
		Price	Error	Price	Error	Price	Error		
CNLF									
64	128	9.42664177	4.0e−4	2.745998655	5.4e−3	0.5479142263	8.1e−3	0.006	2.24
128	512	9.42471690	6.1e−4	2.733174916	7.0e−4	0.5542181415	3.3e−3	0.043	2.41
256	1024	9.43045652	5.1e−8	2.731256352	9.7e−7	0.5523252997	6.8e−5	0.243	5.66
CNAB									
64	128	9.42675126	3.9e−4	2.745841883	5.3e−3	0.5479419939	8.0e−3	0.010	1.28
128	512	9.42474519	6.0e−4	2.733134566	6.9e−4	0.5542258776	3.4e−3	0.050	1.63
256	1024	9.43046363	7.0e−7	2.731246184	4.7e−6	0.5523272883	6.5e−5	0.273	3.20
BDF-2									
64	128	9.42672776	3.9e−4	2.745862104	5.3e−3	0.547915764	8.1e−3	0.010	2.03
128	512	9.42473887	6.1e−4	2.733140158	6.9e−4	0.554218938	3.4e−3	0.060	2.57
256	1024	9.43046202	5.3e−7	2.731247613	4.2e−6	0.552325540	6.8e−5	0.376	4.53

Table 3
Numerical results for European put option under the Merton model for Example 1.

M_τ	N_x	$S = 90$		$S = 100$		$S = 110$		Time (s)	cond
		Price	Error	Price	Error	Price	Error		
CNLF									
64	128	9.27393905	1.2e−3	3.162594916	4.3e−3	1.392284639	6.4e−3	0.016	2.24
128	512	9.28049314	5.3e−4	3.150930743	6.0e−4	1.402503276	9.4e−4	0.063	2.41
256	1024	9.28542608	8.7e−7	3.149026452	1.4e−7	1.401149857	2.6e−5	0.276	5.77
CNAB									
64	128	9.27411418	1.2e−3	3.162453523	4.3e−3	1.3922872191	6.3e−3	0.023	1.29
128	512	9.28053795	5.2e−4	3.150894291	5.9e−4	1.4025045132	9.4e−4	0.073	1.64
256	1024	9.28543733	2.1e−6	3.149017251	2.8e−6	1.4011502052	2.5e−5	0.340	3.29
BDF-2									
64	128	9.27409209	1.2e−3	3.162475554	4.3e−3	1.392266254	6.4e−3	0.026	2.04
128	512	9.28053178	5.3e−4	3.150900321	5.9e−4	1.402498954	9.8e−4	0.086	2.57
256	1024	9.28543575	1.9e−6	3.149018788	2.3e−6	1.401148808	2.6e−5	0.433	4.57

Table 4
Numerical results for American put option under the Kou model for Example 2.

M_τ	N_x	$S = 90$		$S = 100$		$S = 110$		Time (s)	cond
		Price	Error	Price	Error	Price	Error		
CNLF									
64	128	10.0076266	2.5e−4	2.823742587	5.6e−3	0.557440105	7.9e−3	0.006	2.24
128	512	10.0046835	3.9e−5	2.809225746	4.8e−4	0.563643337	3.1e−3	0.040	2.41
256	1024	10.0046929	3.8e−5	2.807848651	1.1e−5	0.561816040	1.1e−4	0.256	5.66
CNAB									
64	128	9.98509195	1.9e−3	2.80926409	4.9e−4	0.5570629962	8.6e−3	0.010	1.29
128	512	10.0041366	9.3e−5	2.79307744	5.3e−3	0.5620876906	3.8e−4	0.040	1.63
256	1024	10.0000751	4.9e−4	2.78947936	6.5e−3	0.5603682064	2.7e−3	0.266	3.37
BDF-2									
64	128	9.98502386	2.0e−3	2.809086344	4.3e−4	0.556943155	8.8e−3	0.010	2.03
128	512	10.0041819	8.9e−5	2.792737034	5.4e−3	0.562068442	3.4e−4	0.056	2.57
256	1024	10.0000758	4.9e−4	2.789427905	6.8e−3	0.560358254	2.7e−3	0.370	4.53

Example 3. In this example, we price European put option under Kou model with the following parameters

$$\sigma = 0.2, \quad r = 0, \quad T = 0.2, \quad K = 1, \quad \lambda = 0.2, \quad \alpha_1 = 3, \quad \alpha_2 = 2, \quad p = 0.5.$$

These parameters are adopted from [47]. The reference value for European put option under Kou model is 0.042647805 at $S = 1$. In this example the computation interval is $\Omega = [x_{\min}, x_{\max}] = [-4, 4]$, and numerical results including errors at different asset prices, cpu time in second and condition number of linear system of equations are shown in the Table 6 for different space N_x and time M_τ steps and different time discretization schemes. The numerical results are in accordance with those found in [47]. Also, the option value, Delta and Gamma as the Greeks are plotted in Fig. 3.

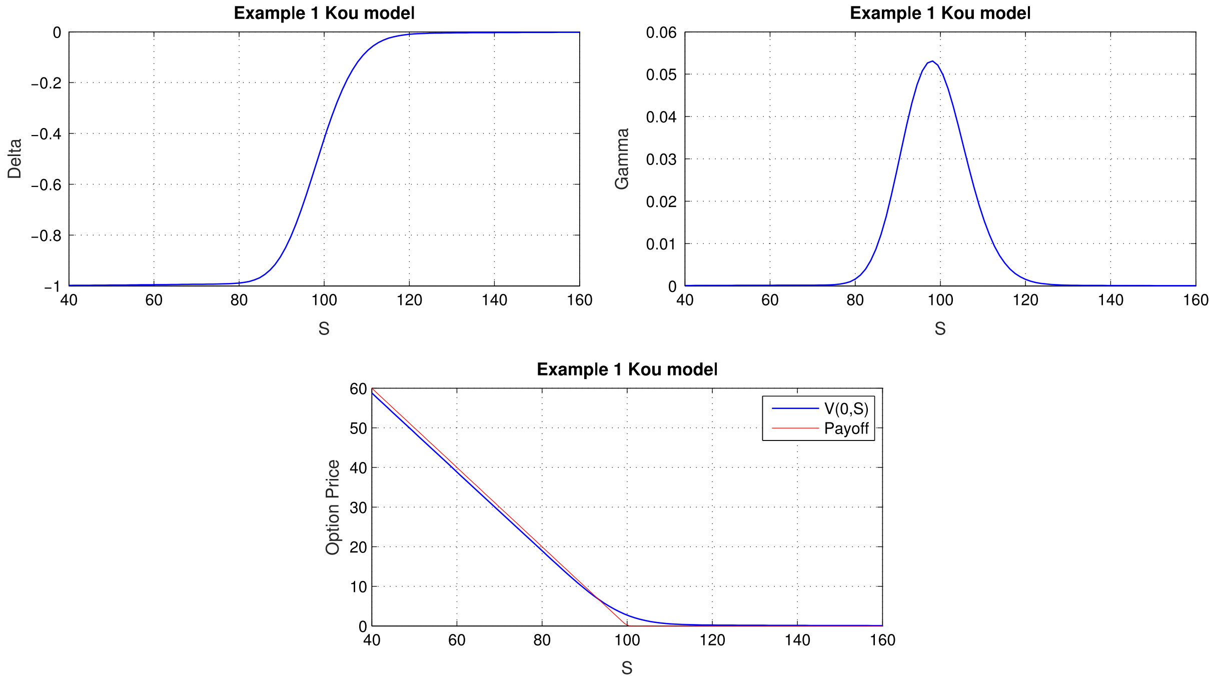

Fig. 1. The option value, Delta and Gamma functions for Example 1 and Kou model.

Table 5
Numerical results for American put option under the Merton model for Example 2.

M_τ	N_x	$S = 90$		$S = 100$		$S = 110$		Time (s)	cond
		Price	Error	Price	Error	Price	Error		
CNLF									
64	128	10.0095536	5.7e−4	3.255032508	4.7e−3	1.41178599	5.6e−3	0.016	2.26
128	512	10.0016200	2.2e−4	3.242254687	3.2e−4	1.42109123	9.1e−4	0.063	2.41
256	1024	10.0024419	1.4e−4	3.241222821	2.4e−6	1.1.419752	3.0e−5	0.306	5.77
CNAB									
64	128	9.98193166	2.2e−3	3.238885096	7.2e−4	1.411660715	5.7e−3	0.023	1.29
128	512	10.0049260	1.1e−4	3.219978857	6.5e−3	1.419613174	1.3e−4	0.073	1.63
256	1024	9.99995971	3.8e−4	3.218582923	6.9e−3	1.418371237	1.0e−3	0.330	3.25
BDF-2									
64	128	9.98185622	2.2e−3	3.238575749	8.1e−4	1.411531249	5.8e−3	0.023	2.04
128	512	10.0049693	1.1e−4	3.219585889	6.7e−3	1.419600374	1.4e−4	0.090	2.57
256	1024	9.99995862	3.8e−4	3.218424649	7.0e−3	1.418362133	1.0e−3	0.436	4.57

Example 4. In this Example, we price European put options under Kou model with the same parameters as in Example 1 except that the local volatility is given by the function

$$\sigma(t,S) = 0.15 + 0.15(0.5+2t)\frac{(\frac{S}{100}-1.2)^2}{(\frac{S}{100})^2+1.44}.$$

The reference value [47] for European put option under Kou model is 9.456392 at $S = 90$, 2.767587 at $S = 100$, and 0.561144 at $S = 110$. In this example the computation interval is $\Omega = [x_{\min}, x_{\max}] = [-4, 4]$, and numerical results including errors at different asset prices, cpu time in second and condition number of linear system of equations are shown in Table 7 for different space N_x and time M_τ steps and different time discretization schemes. Numerical results show that at the strike price $S = K$, RBF-PU with CNLF is accurate than method of [47], but in points far from strike price results presented in [47] are efficient. Also, the option value, Delta and Gamma as the Greeks are plotted in Fig. 4.

Example 5. In this Example, for more discussion about accuracy and stability of RBF-PU method, we consider Merton model for European put option with large jump intensity. So, we price European put options under Merton model with the same parameters as in Example 1 except that the jump intensity λ is chosen to be 25 and 50.

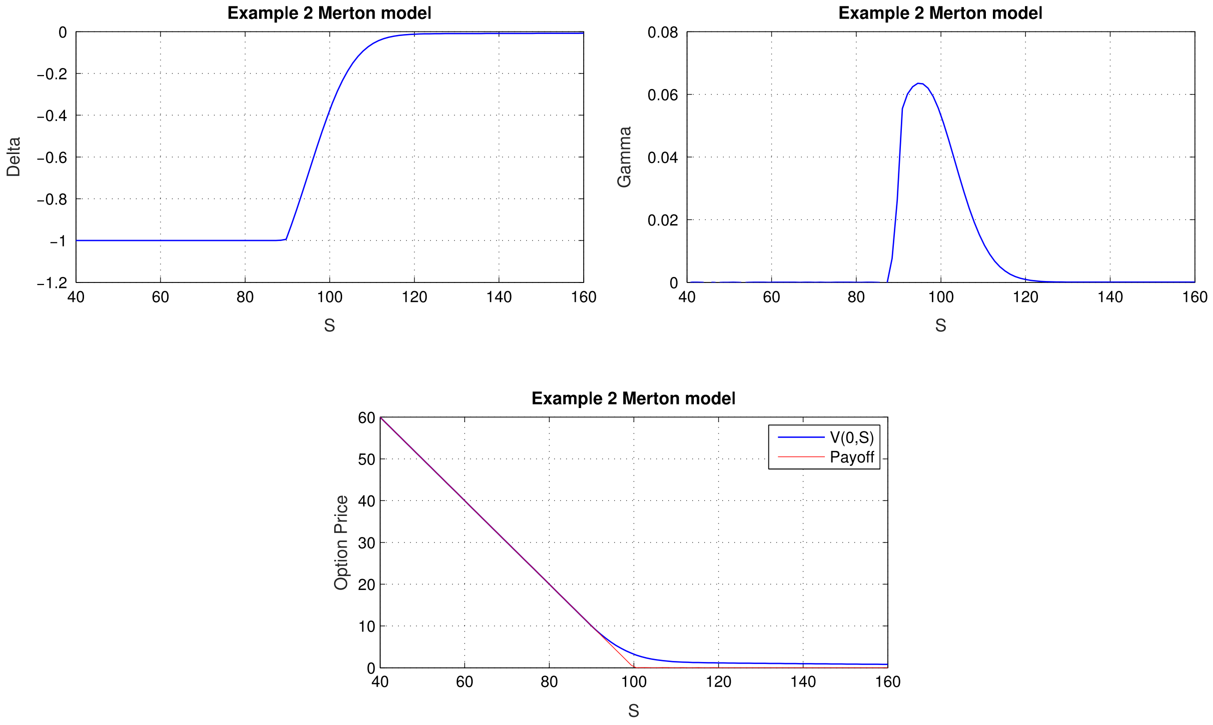

Fig. 2. The option value, Delta and Gamma functions for Example 2 and Merton model.

Table 6
Numerical results for European put option under the Kou model for Example 3.

M_τ	N_x	$S = 1$		Time (s)	cond
		Price	Error		
CNLF					
64	128	0.04269837301	1.2e−3	0.006	2.30
128	512	0.04264461463	7.5e−5	0.040	2.68
256	1024	0.04264770068	2.4e−6	0.286	7.73
CNAB					
64	128	0.04269630337	1.1e−3	0.006	1.04
128	512	0.04264408798	8.7e−5	0.040	1.88
256	1024	0.04264756838	5.5e−6	0.276	4.39
BDF-2					
64	128	0.04269649354	1.1e−3	0.010	2.26
128	512	0.04264414022	8.6e−5	0.056	3.03
256	1024	0.04264758172	5.2e−6	0.356	5.93

The reference value [50] for $\lambda = 25$ is 66.07703584 at $S = 90$, 64.56736614 at $S = 100$, and 63.18682693 at $S = 110$, and for $\lambda = 50$ is 81.59880768 at $S = 90$, 80.78966717 at $S = 100$, and 80.03974097 at $S = 110$. In this example the computation interval is $\Omega = [x_{\min}, x_{\max}] = [-21, 21]$, and numerical results including errors at different asset prices, cpu time in second and condition number of linear system of equations are shown in Tables 8 and 9 for different space N_x and time M_τ steps and different time discretization schemes.

Example 6. In this example, we price American put options under Kou and Merton models with parameters

$$\sigma = 0.1, \quad r = 0.1, \quad T = 1, \quad K = 100, \quad \lambda = 0.5.$$

For numerical technique, we set $\Omega = [x_{\min}, x_{\max}] = [-4, 4]$. The reference values of the American put option under the Merton model are 19.948906 at $S = 90$, 18.246332 at $S = 100$ and 16.666925 at $S = 110$. The reference values of the

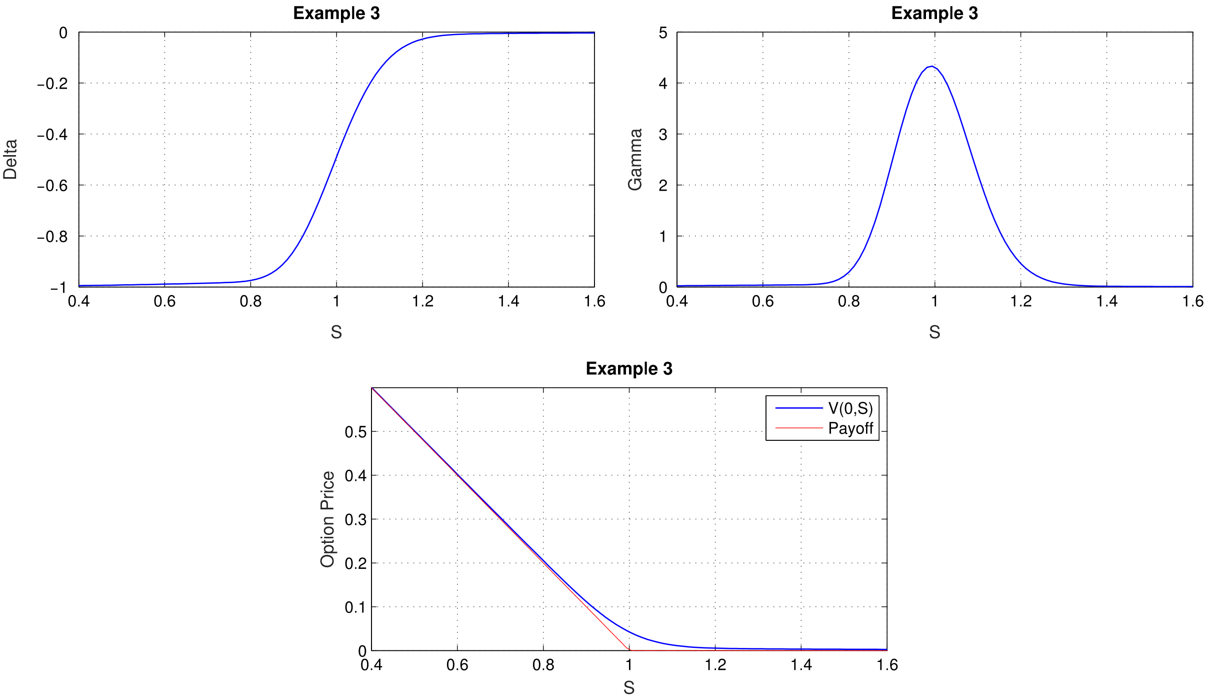

Fig. 3. The option value, Delta and Gamma functions for Example 3 and Kou model.

Table 7
Numerical results for European put option under the Kou model with local volatility for Example 4.

M_τ	N_x	$S = 90$		$S = 100$		$S = 110$		Time (s)	cond
		Price	Error	Price	Error	Price	Error		
CNLF									
64	128	9.45568716	7.4e−5	2.781672466	5.1e−3	0.555162007	1.1e−2	0.200	2.94
128	512	9.45295859	3.6e−4	2.769369241	6.4e−4	0.561817963	1.2e−3	2.123	5.47
256	1024	9.45895936	2.7e−4	2.767595832	3.2e−6	0.559939518	2.1e−3	21.126	20.18
CNAB									
64	128	9.45579308	6.3e−5	2.781519088	5.0e−3	0.5551908807	1.1e−2	0.270	2.01
128	512	9.45298456	3.6e−4	2.769329228	6.3e−4	0.5618260645	1.2e−3	3.200	3.23
256	1024	9.45896458	2.7e−4	2.767585195	6.5e−7	0.5599416666	2.1e−3	27.986	10.19
BDF-2									
64	128	9.45761797	1.3e−4	2.762329601	1.9e−3	0.543860912	3.1e−2	0.146	3.55
128	512	9.45385417	2.7e−4	2.759619407	2.9e−3	0.556036672	9.1e−3	1.993	4.86
256	1024	9.45938225	3.2e−4	2.762717795	1.7e−3	0.557050480	7.3e−3	16.253	15.43

American put option under the Kou model are 10.698208 at $S = 90$, 6.417275 at $S = 100$ and 4.624099 at $S = 110$. These values can be found in [51]. Results are reported in Tables 10 and 11.

To discuss about convergence of the time discretization, we numerically investigate the behavior of the global temporal errors as a function of $\delta\tau$ which is defined by

$$\text{Temporal error} = \sqrt{\frac{1}{N_x}\sum_{j=1}^{N_x}\left(\frac{V^k(0,S_j) - V^{10\,000}(0,S_j)}{V^{10\,000}(0,S_j)}\right)^2}, \tag{29}$$

where V^k is the numerical solution of European and American options at the spatial nodes $S_j \in (\frac{K}{2}, \frac{3K}{2})$ after k time steps, and $V^{10\,000}$ is the corresponding solution for $k = 10\,000$ used as an approximation for the exact solution. Figs. 5–7 display the global temporal errors versus $\delta\tau$ for CNLF, CNAB and BDF-2 schemes and for sequence of six increasing time steps k, namely 8, 16, 32, 64, 128, 256 and $N_x = 512$ spatial nodes for Examples 1–6.

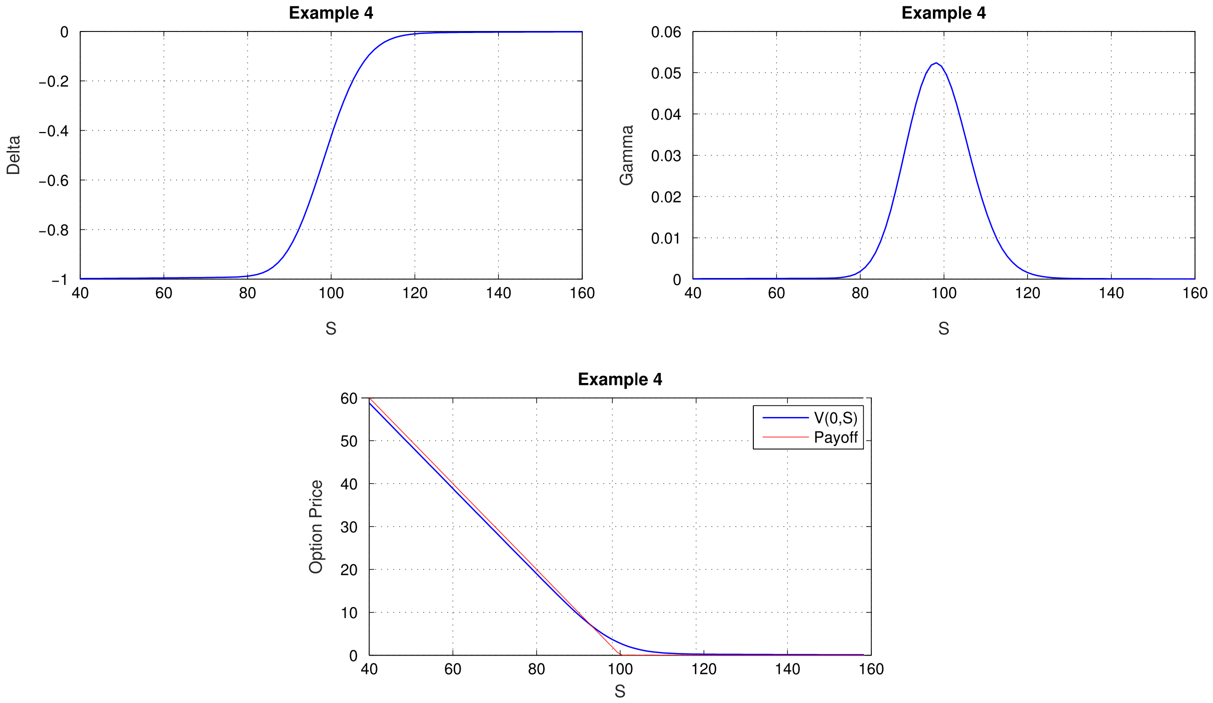

Fig. 4. The option value, Delta and Gamma functions for Example 4 and Kou model.

Table 8
Numerical results for European put option under the Merton model with $\lambda = 25$ for Example 5.

M_τ	N_x	$S = 90$		$S = 100$		$S = 110$		Time (s)	cond
		Price	Error	Price	Error	Price	Error		
CNLF									
64	128	66.2795466	3.1e−3	64.65695038	1.4e−3	62.9859099	3.2e−3	0.013	2.62
128	512	65.2474607	1.2e−2	64.63965706	1.1e−3	62.5704635	9.7e−3	0.060	3.25
256	1024	66.0216217	8.4e−4	64.59551682	4.3e−4	63.1699677	2.7e−4	0.321	3.59
CNAB									
64	128	66.2689964	2.9e−3	64.6446335	1.2e−3	62.97271527	3.4e−3	0.020	1.40
128	512	65.2447347	1.2e−2	64.6362690	1.1e−3	62.56721822	9.8e−3	0.066	1.79
256	1024	66.0208252	8.5e−4	64.5947285	4.2e−4	63.16909336	2.8e−4	0.353	2.17
BDF-2									
64	128	66.2719099	2.9e−3	64.64868214	1.2e−3	62.9774841	3.3e−3	0.023	2.28
128	512	65.2446846	1.2e−2	64.63640293	1.1e−3	62.5672883	9.8e−3	0.083	2.79
256	1024	66.0208494	8.5e−4	64.59472989	4.2e−4	63.1691225	2.8e−4	0.446	3.08

As a first main observation, the global temporal errors decrease monotonically as M_τ increases or equivalently $\delta\tau$ decreases. Concerning the actual convergence behavior, it is easy to see from Figs. 5–7 that the temporal errors as a function of $\delta\tau$ are bounded from above in each case by $C(\delta\tau)^p$ with some moderate constants C where $p \approx 2$ for more cases.

The accuracy of RBF methods highly depends upon the shape parameter ϵ of the basis functions, which is responsible for the flatness of the functions. For smooth problems, the best accuracy is typically achieved when ϵ is small, but then the condition number of the linear system becomes very large. In this part of the paper, we try to find the best compromise for the size of ϵ for our problem. Fig. 8 displays the dependence of the Error defined by

$$\text{Error} = \sqrt{\frac{1}{N_{eval}} \sum_{i=1}^{N_{eval}} \left(\frac{V(0, S_i) - V_{ref}(0, S_i)}{V_{ref}(0, S_i)} \right)^2}, \tag{30}$$

on the size of the shape parameter for the CNLF scheme and American and European options presented in Examples 1–6. In this definition, N_{eval} means number of evaluation points. In Example 3, $N_{eval} = 1$ and evaluation is performed at $S = 1$, and in all other examples $N_{eval} = 3$ and evaluation are performed at $S = 90, 100, 110$. Also, V_{ref} is the reference value

Table 9
Numerical results for European put option under the Merton model with $\lambda = 50$ for Example 5.

M_τ	N_x	$S = 90$		$S = 100$		$S = 110$		Time (s)	cond
		Price	Error	Price	Error	Price	Error		
CNLF									
64	128	81.2690315	4.0e−3	80.3395912	5.6e−3	79.30881743	9.1e−3	0.016	3.04
128	512	80.5666462	1.3e−2	80.9017254	1.4e−3	79.29764274	9.3e−3	0.060	4.00
256	1024	81.5569489	5.1e−4	80.8505427	7.5e−4	80.04428815	5.7e−5	0.306	5.68
CNAB									
64	128	81.2527329	4.2e−3	80.32191224	5.8e−3	79.28992521	9.4e−3	0.026	1.81
128	512	80.5613454	1.3e−2	80.89580517	1.3e−3	79.29148446	9.3e−3	0.070	2.59
256	1024	81.5555658	5.3e−4	80.84905509	7.3e−4	80.04270278	3.7e−5	0.353	3.35
BDF-2									
64	128	81.2708377	4.1e−3	80.34095998	5.5e−3	79.30983826	9.1e−3	0.026	3.21
128	512	80.5639230	1.3e−2	80.89861484	1.3e−3	79.29443347	9.3e−3	0.090	3.97
256	1024	81.5562002	5.2e−4	80.84973873	7.4e−4	80.04342847	4.6e−5	0.456	4.63

Table 10
Numerical results for American put option under the Kou model for Example 6.

M_τ	N_x	$S = 90$		$S = 100$		$S = 110$		Time (s)	cond
		Price	Error	Price	Error	Price	Error		
CNLF									
64	128	10.64084579	5.4e−3	6.412075743	8.1e−4	4.593321237	6.6e−3	0.006	2.36
128	512	10.69138383	6.4e−4	6.417126926	2.3e−5	4.625021498	2.0e−4	0.040	3.14
256	1024	10.69588043	2.2e−4	6.417371451	1.1e−5	4.624253797	3.3e−5	0.256	10.21
CNAB									
64	128	10.12211356	5.4e−2	6.386864933	4.7e−3	4.585504179	8.3e−3	0.006	1.52
128	512	10.39503274	2.8e−2	6.385103756	5.0e−3	4.618203872	1.3e−3	0.053	2.12
256	1024	10.35392468	3.2e−2	6.386614237	4.8e−3	4.617585966	1.4e−3	0.090	5.57
BDF-2									
64	128	10.12303279	5.4e−2	6.386563118	4.8e−3	4.58544356	8.3e−3	0.010	2.51
128	512	10.39976975	1.8e−2	6.384902961	5.0e−3	4.61812649	1.3e−3	0.056	3.52
256	1024	10.36344012	3.1e−2	6.386560969	4.8e−3	4.61755460	1.4e−3	0.373	7.48

Table 11
Numerical results for American put option under the Merton model for Example 6.

M_τ	N_x	$S = 90$		$S = 100$		$S = 110$		Time (s)	cond
		Price	Error	Price	Error	Price	Error		
CNLF									
64	128	19.89116273	2.9e−3	18.21208449	1.9e−3	16.61800262	2.9e−3	0.016	2.47
128	512	19.94640105	1.2e−4	18.24433217	1.1e−4	16.66528098	9.9e−5	0.060	3.93
256	1024	19.94894104	1.7e−6	18.24636221	1.6e−6	16.66695116	1.6e−6	0.320	13.23
CNAB									
64	128	19.99192112	2.1e−3	18.19828606	2.6e−3	16.60712916	3.6e−3	0.033	1.64
128	512	19.90569885	2.2e−3	18.23087976	8.5e−4	16.64962592	1.0e−3	0.076	2.51
256	1024	19.90519493	2.2e−3	18.23464176	6.4e−4	16.65171553	9.1e−4	0.340	6.55
BDF-2									
64	128	19.99256264	2.2e−3	18.19802769	2.6e−3	16.60670145	3.6e−3	0.030	2.79
128	512	19.90366105	2.3e−3	18.23075321	8.5e−4	16.64957617	1.0e−3	0.090	4.07
256	1024	19.90564461	2.2e−3	18.23462274	6.4e−4	16.65169651	9.1e−4	0.446	8.73

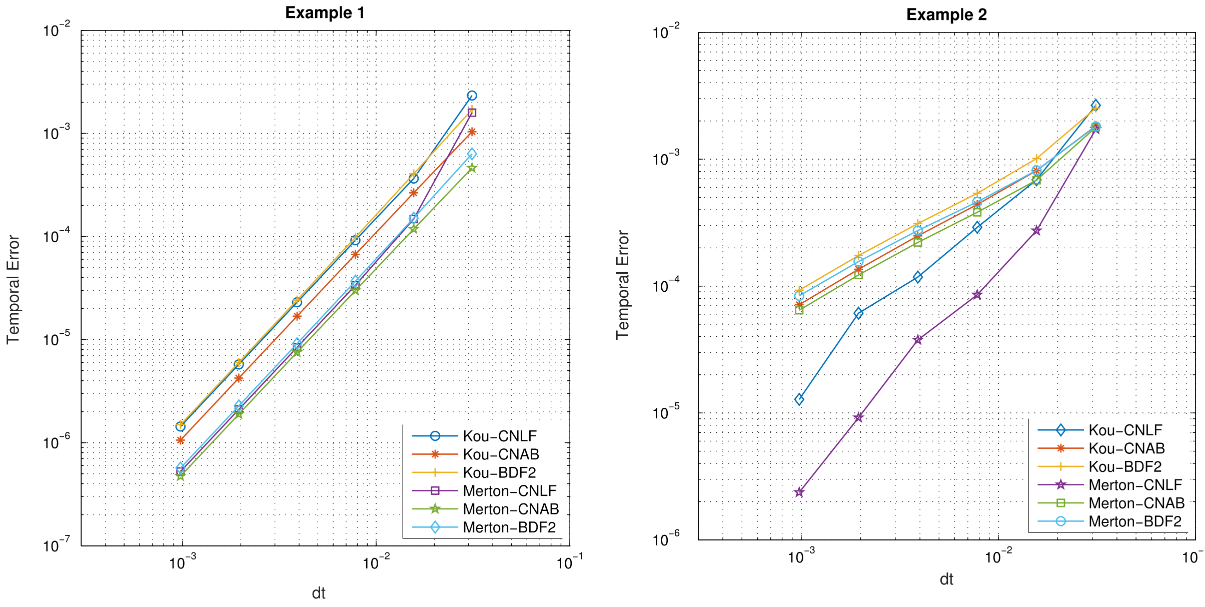

Fig. 5. Temporal error vs. $\delta\tau$ for the CNLF, CNAB and BDF-2 schemes with $N_x = 512$ spatial nodes for Examples 1 and 2.

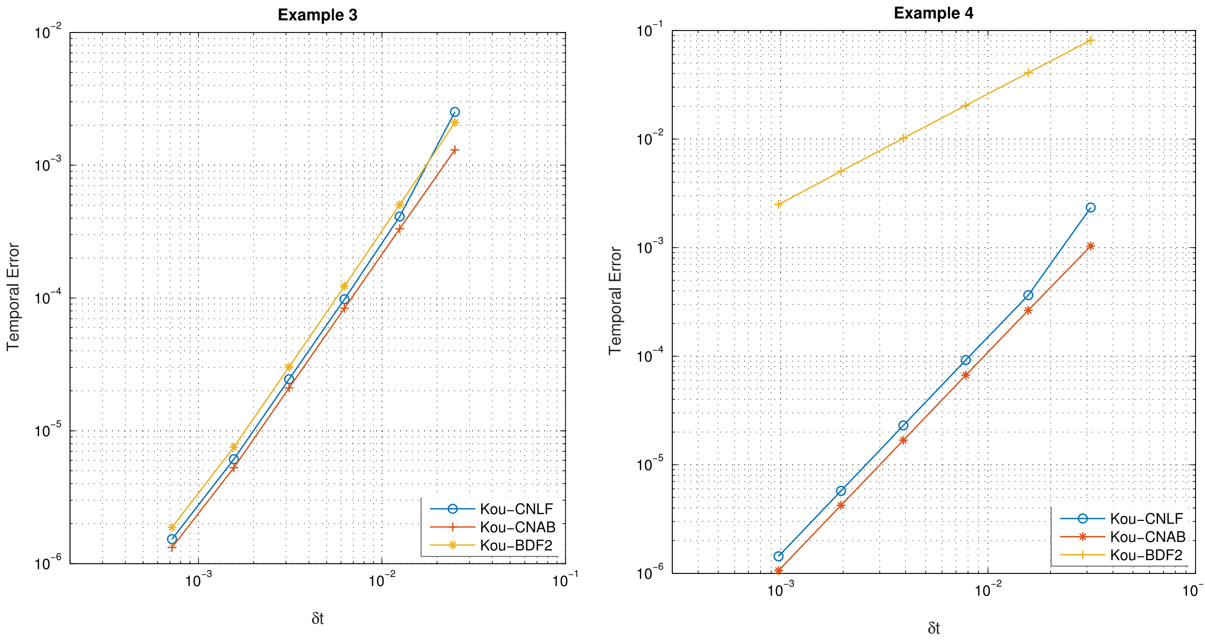

Fig. 6. Temporal error vs. $\delta\tau$ for the CNLF, CNAB and BDF-2 schemes with $N_x = 512$ spatial nodes for Examples 3 and 4.

presented in each example separately. According to Fig. 8, we choose $\epsilon = 10$ for the European and American option pricing experiments.

In the previous section, we presented some propositions and corollaries related to stability analysis of CNLF, CNAB and BDF-2 time discretizations schemes. Corollaries 1–3 guarantee the stability of CNLF, CNAB and BDF-2 schemes if all real eigenvalues of differentiation matrix **D** corresponding to the differential operator $\mathcal{D}$ are nonpositive. Figs. 9–11 show real part of the largest eigenvalue of matrix **D** for different values of h known as space step size and ϵ known as shape parameter of RBFs for different examples.

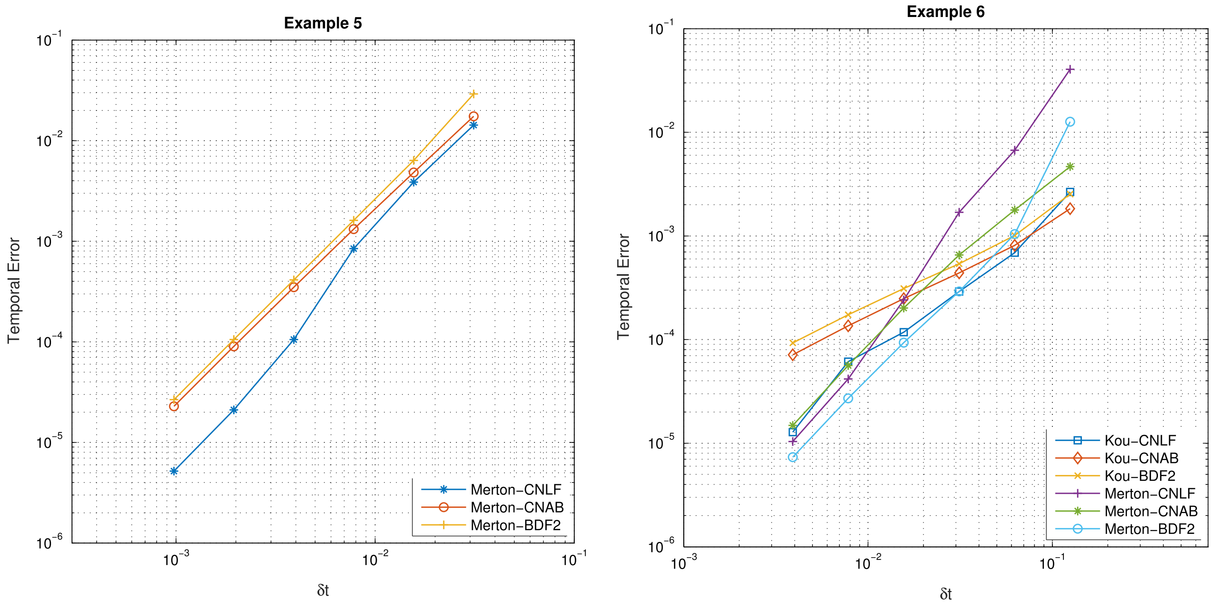

Fig. 7. Temporal error vs. $\delta\tau$ for the CNLF, CNAB and BDF-2 schemes with $N_x = 512$ spatial nodes for Examples 5 and 6.

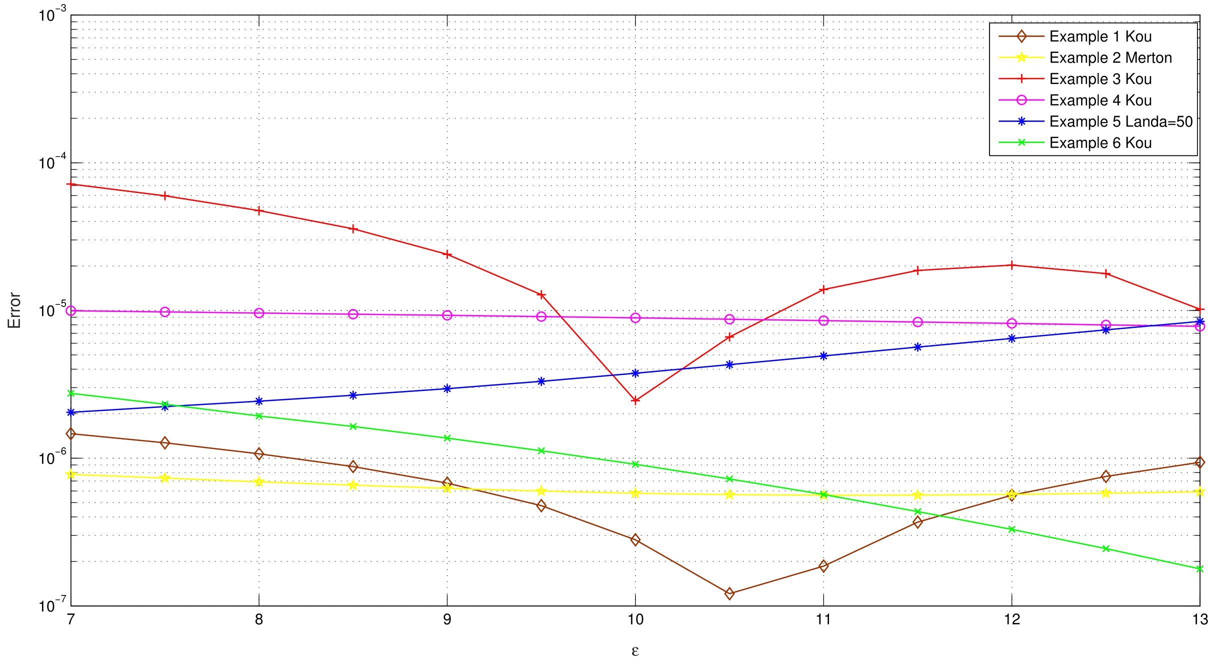

Fig. 8. Error in the price of the different option models against the shape parameter ϵ.

8. Conclusion

We proposed RBF-PU method to price American and European options under the jump–diffusion model. The free boundary problem formulated as a PDE was transformed into an LCP problem. The RBF-PU method was used for the spatial discretization. Then, CNLF, CNAB and BDF-2 time discretizations were combined with an operator splitting method. These

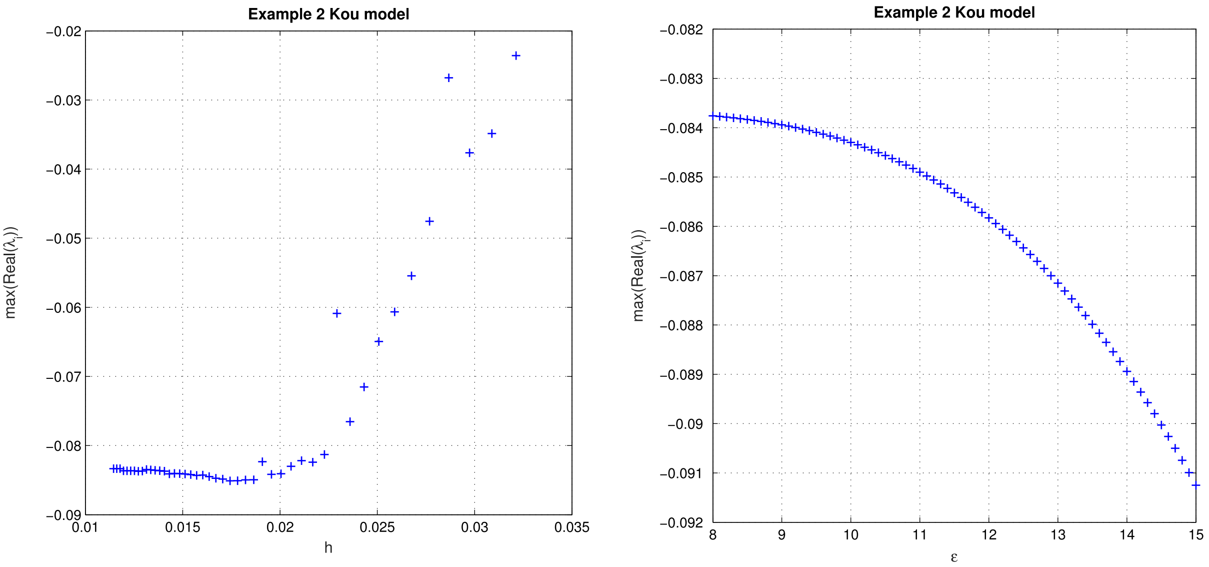

Fig. 9. Real part of the largest eigenvalue of matrix **D** vs. space step size h (left) and ϵ (right) for Example 2.

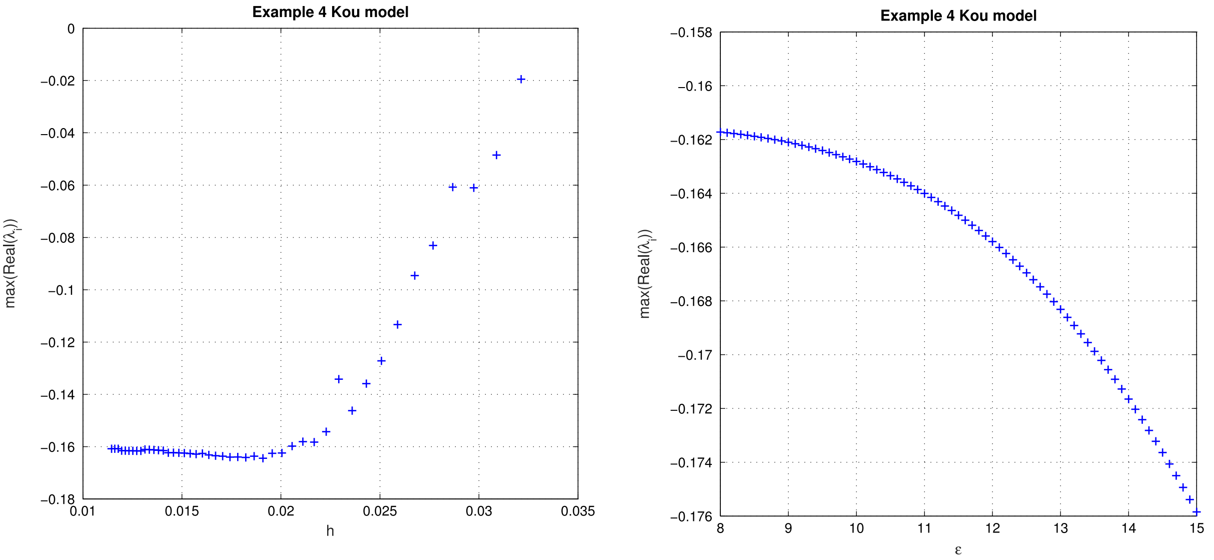

Fig. 10. Real part of the largest eigenvalue of matrix **D** vs. space step size h (left) and ϵ (right) for Example 4.

result in a linear algebraic system with a sparse matrix that has a small condition number. The shape parameter in the RBF affects the accuracy and stability of the method. Fig. 8 confirms that RBF-PU method is less sensitive to the change of shape parameter. The numerical experiments presented in tables confirm that CNLF time discretization performs slightly better and faster than CNAB and BDF-2 schemes. An increase of the number of nodal points and correspondingly an increase of the number of patches also leads to an improvement of the approximation. The effect of the time discretization is measured by studying the temporal error. For the American and European option cases where the CNLF, CNAB and BDF-2 schemes are combined with an operator splitting method, we conclude that the rate of convergence is of second order for more cases. The experiments also demonstrated that European and American option prices with error around $1.0e-4$ can be computed in less than one second on a PC. Thus, the developed method is very efficient and accurate.

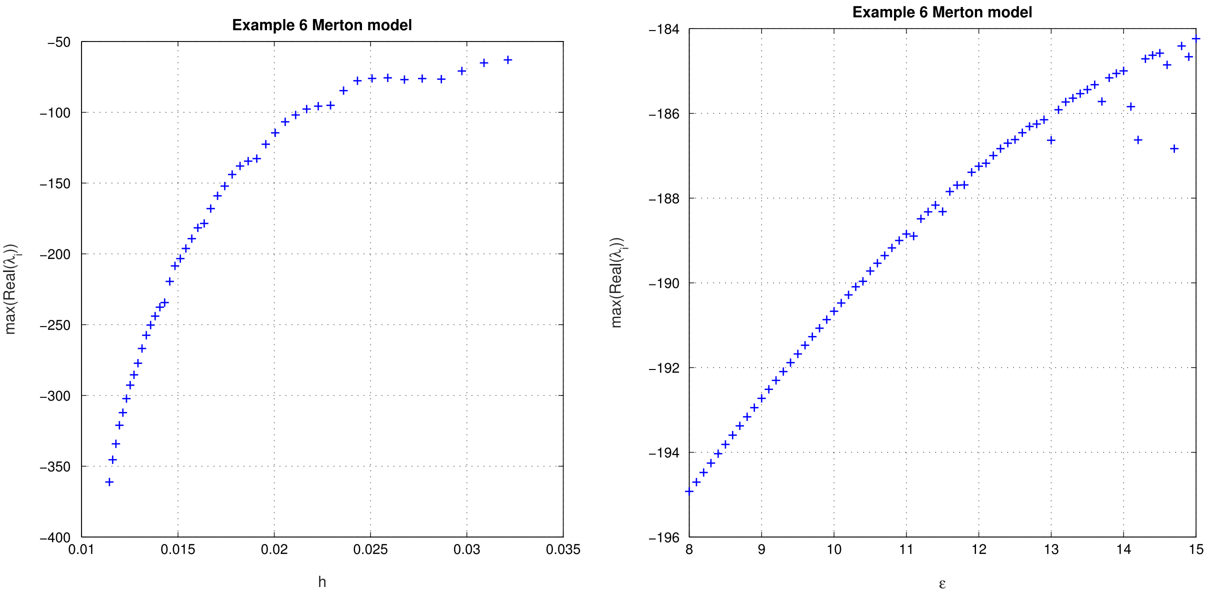

Fig. 11. Real part of the largest eigenvalue of matrix **D** vs. space step size h (left) and ϵ (right) for Example 6.

Acknowledgment

The authors would like to thank professor Alfa R.H. Heryudono for some fruitful comments. Hengguang Li was partially supported by the NSF Grant DMS-1418853, by the Natural Science Foundation of China (NSFC) Grant 11628104, and by the Wayne State University Grants Plus Program.

References

[1] F. Black, M. Scholes, The pricing of options and corporate liabilities, J. Polit. Econ. 81 (3) (1973) 637–654.
[2] G.P.J.P. Fouque, K.R. Sircar, Derivatives in Financial Markets with Stochastic Volatility, Cambridge University Press, Cambridge, 2000.
[3] R. Merton, Option pricing when underlying stock returns are discontinuous, J. Financ. Econ. 3 (1–2) (1976) 125–144.
[4] S. Kou, A jump-diffusion model for option pricing, Manage. Sci. 48 (8) (2002) 1086–1101.
[5] P. Carr, H. Geman, The fine structure of asset returns: An empirical investigation, J. Bus. 75 (2) (2002) 305–332.
[6] D. Bates, Jumps and stochastic volatility: Exchange rate processes implicit in Deutsche mark options, Rev. Financ. Stud. 9 (1) (1996) 69–107.
[7] B. Eraker, M. Johannes, N. Polson, The impact of jumps in volatility and returns, J. Finance 58 (3) (2003).
[8] R. Cont, P. Tankov, Financial Modelling with Jump Processes, Chapman & Hall, Cambridge, 2004.
[9] A.-M. Matache, P.-A. Nitsche, C. Schwab, Wavelet Galerkin pricing of American options on Lévy driven assets, Quant. Finance 5 (4) (2005) 403–424.
[10] L. Andersen, J. Andreasen, Jump-diffusion processes: Volatility smile fitting and numerical methods for option pricing, Rev. Deriv. Res. 4 (3) (2000) 231–262.
[11] R. Cont, E. Voltchkova, A finite difference scheme for option pricing in jump diffusion and exponential Lévy models, SIAM J. Numer. Anal. 43 (4) (2005) 1596–1626.
[12] Y. d'Halluin, P. Forsyth, G. Labahn, A penalty method for American options with jump diffusion processes, Numer. Math. 97 (2) (2004) 321–352.
[13] F. Fang, C.W. Oosterlee, A novel pricing method for European options based on Fourier-Cosine series expansions, SIAM J. Sci. Comput. 31 (2) (2009) 826–848.
[14] F. Fang, C.W. Oosterlee, Pricing early-exercise and discrete barrier options by fourier-cosine series expansions, Numer. Math. 114 (1) (2009) 27.
[15] A. Almendral, C.W. Oosterlee, Numerical valuation of options with jumps in the underlying, Appl. Numer. Math. 53 (1) (2005) 1–18.
[16] K. Zhang, S. Wang, Pricing options under jump diffusion processes with fitted finite volume method, Appl. Math. Comput. 201 (12) (2008) 398–413.
[17] E. Ngounda, K.C. Patidar, E. Pindza, Contour integral method for European options with jumps, Commun. Nonlinear Sci. Numer. Simul. 18 (3) (2013) 478–492.
[18] E. Pindza, K. Patidar, E. Ngounda, Robust spectral method for numerical valuation of European options under Merton's jump-diffusion model, Numer. Methods Partial Differential Equations 30 (4) (2014).
[19] M. Briani, R. Natalini, G. Russo, Implicit–explicit numerical schemes for jump–diffusion processes, Calcolo 44 (1) (2007) 33–57.
[20] Y. d'Halluin, P.A. Forsyth, K.R. Vetzal, Robust numerical methods for contingent claims under jump diffusion processes, IMA J. Numer. Anal. 25 (1) (2005) 87.
[21] R. Lord, F. Fang, F. Bervoets, C.W. Oosterlee, A fast and accurate FFT-based method for pricing early-exercise options under Lévy processes, SIAM J. Sci. Comput. 30 (4) (2008) 1678–1705.
[22] L. Ballestra, G. Pacelli, Pricing European and American options with two stochastic factors: A highly efficient radial basis function approach, J. Econom. Dynam. Control 37 (6) (2013) 1142–1167.
[23] G. Fasshauer, A. Khaliq, D. Voss, Using meshfree approximation for multiasset American options, J. Chin. Inst. Eng. 27 (4) (2004) 563–571.
[24] Z. Wu, Y. Hon, Convergence error estimate in solving free boundary diffusion problem by radial basis functions method, Eng. Anal. Bound. Elem. 27 (1) (2003) 73–79.

[25] L. Ballestra, G. Pacelli, Computing the survival probability density function in jump-diffusion models: A new approach based on radial basis functions, Eng. Anal. Bound. Elem. 35 (9) (2011) 1075–1084.

[26] L. Ballestra, G. Pacelli, A radial basis function approach to compute the first-passage probability density function in two-dimensional jump-diffusion models for financial and other applications, Eng. Anal. Bound. Elem. 36 (11) (2012) 1546–1554.

[27] H. Wendland, Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree, Adv. Comput. Math. 4 (1) (1995) 389–396.

[28] I. Babuška, J. Melenk, The partition of unity method, Internat. J. Numer. Methods Engrg. 40 (4) (1997) 727–758.

[29] H. Wendland, Fast evaluation of radial basis functions: Methods based on partition of unity, in: C. Chui, L. Schumaker, J. Stöckler (Eds.), Approximation Theory X: Wavelets, Splines, and Applications, Vanderbilt University Press, Nashville, 2002, pp. 473–483.

[30] R. Cavoretto, A.D. Rossi, Spherical interpolation using the partition of unity method: An efficient and flexible algorithm, Appl. Math. Lett. 25 (10) (2012) 1251–1256.

[31] R. Cavoretto, A.D. Rossi, A meshless interpolation algorithm using a cell-based searching procedure, Comput. Math. Appl. 67 (5) (2014) 1024–1038.

[32] A. Safdari-Vaighani, A. Heryudono, E. Larsson, A radial basis function partition of unity collocation method for convection–diffusion equations arising in financial applications, J. Sci. Comput. 64 (2) (2015) 341–367.

[33] R. Mollapourasl, A. Fereshtian, M. Vanmaele, Radial basis functions with partition of unity method for American options with stochastic volatility, Comput. Econ. (2017).

[34] H. Li, R. Mollapourasl, M. Haghi, A local radial basis function method for pricing options under the regime switching model, J. Sci. Comput. (2018) under review.

[35] S. Ikonen, J. Toivanen, Operator splitting methods for American option pricing, Appl. Math. Lett. 17 (7) (2004) 809–814.

[36] R. Seydel, Tools for Computational Finance, fourth ed., Springer, Berlin, Heidelberg, 2009.

[37] G. Fasshauer, Meshfree Approximation Methods with Matlab, World Scientific Publishing Co, Singapore, 2007.

[38] M. Powell, The theory of radial basis functions approximation in 1990, in: W. Light (Ed.), Advances in Numerical Analysis. Vol. II. Wavelets, Subdivision Algorithms and Radial Basis Functions, Oxford Univ. Press, London, 1992, pp. 105–210.

[39] M. Buhmann, Radial basis functions, Acta Numer. 9 (2000) 1–38.

[40] M. Buhmann, Radial Basis Functions: Theory and Implementations, Cambridge University Press, New York, 2003.

[41] H. Wendland, Scattered Data Approximation, in: Cambridge Monographs on Applied and Computational Mathematics, no. 17, Cambridge University Press, New York, 2005.

[42] D. Shepard, A two-dimensional interpolation function for irregularly-spaced data, in: Proceedings of the 1968 23rd ACM National Conference, ACM'68, ACM, New York, NY, USA, 1968, pp. 517–524.

[43] R. Schaback, A unified theory of radial basis functions, J. Comput. Appl. Math. 121 (1) (2000) 165–177.

[44] U.M. Ascher, S.J. Ruuth, B.T.R. Wetton, Implicit-explicit methods for time-dependent partial differential equations, SIAM J. Numer. Anal. 32 (3) (1995) 797–823.

[45] J. Frank, W. Hundsdorfer, J. Verwer, On the stability of implicit-explicit linear multistep methods, Appl. Numer. Math. 25 (2) (1997) 193–205.

[46] S. Salmi, J. Toivanen, IMEX schemes for pricing options under jump-diffusion models, Appl. Numer. Math. 84 (2014) 33–45.

[47] J. Toivanen, Numerical valuation of European and American options under Kou's jump-diffusion model, SIAM J. Sci. Comput. 30 (4) (2008) 1949–1970.

[48] Y. Kwon, Y. Lee, A second-order finite difference method for option pricing under jump-diffusion models, SIAM J. Numer. Anal. 49 (6) (2011) 2598–2617.

[49] S. Salmi, J. Toivanen, An iterative method for pricing American options under jump-diffusion models, Appl. Numer. Math. 61 (7) (2011) 821–831.

[50] M.K. Kadalbajoo, L.P. Tripathi, A. Kumar, Second order accurate IMEX methods for option pricing under Merton and kou jump-diffusion models, J. Sci. Comput. 65 (3) (2015) 979–1024.

[51] S. Salmi, J. Toivanen, Comparison and survey of finite difference methods for pricing American options under finite activity jump-diffusion models, Int. J. Comput. Math. 89 (9) (2012) 1112–1134.