Open access • Journal Article • DOI:10.1137/060674697

Numerical Valuation of European and American Options under Kou's Jump-Diffusion Model — Source link

Jari Toivanen

Institutions: Stanford University

Published on: 01 Apr 2008 - SIAM Journal on Scientific Computing (Society for Industrial and Applied Mathematics)

Topics: Valuation of options, Binomial options pricing model, Jump diffusion, Partial differential equation and Penalty method

Numerical Valuation of European and American Options under Kou's Jump-Diffusion Model*

Jari Toivanen†

Abstract

Numerical methods are developed for pricing European and American options under Kou's jump-diffusion model which assumes the price of the underlying asset to behave like a geometrical Brownian motion with a drift and jumps whose size is log-double-exponentially distributed. The price of a European option is given by a partial integro-differential equation (PIDE) while American options lead to a linear complementarity problem (LCP) with the same operator. Spatial differential operators are discretized using finite differences on nonuniform grids and time stepping is performed using the implicit Rannacher scheme. For the evaluation of the integral term easy to implement recursion formulas are derived which have optimal computational cost. When pricing European options the resulting dense linear systems are solved using a stationary iteration. For American options two ways to solve the LCPs are described: an operator slitting method and a penalty method. Numerical experiments confirm that the developed methods are very efficient as fairly accurate option prices can be computed in a few milliseconds on a PC.

Keywords: option pricing, jump-diffusion model, partial integro-differential equation, linear complementarity problem, finite difference method, operator splitting method, penalty method

1 Introduction

The amount of financial option trading has grown to enormous scale since the pioneering work by Black and Scholes [7] and Merton [33] on the pricing of options in 1973. During the last two decades it has become evident that their assumption that the price of underlying asset behaves like a geometric Brownian motion with a drift and a constant volatility cannot explain the market prices of options with various

*This research was supported by the Academy of Finland, grant #207089

†Department of Mathematical Information Technology, University of Jyväskylä, PO Box 35 (Agora), FI-40014 University of Jyväskylä, Finland, Jari.Toivanen@mit.jyu.fi

strike prices and maturities. Already 1976 Merton proposed to add jumps which have normally distributed size to the behavior of asset prices [34]. During the last ten years the research on models with jumps have become very active. A large number of such models have been proposed; see [12] and references therein. The model proposed by Kou in [27] assumes the distribution of jump sizes to be a log-double-exponential function; see also [28]. In a more rich CGMY model [9] the asset price is a Lévy process with possibly infinite jump activity. Another generalization is to assume the volatility to be stochastic like in [20], for example. A popular approach is to use a deterministic volatility model where the volatility is a function of time and the price of the underlying asset [17]. This approach was combined with Merton's jump-diffusion model in [4]. The most generic models assume stochastic volatility with jumps in the asset price and possibly also in volatility like the ones in [6], [16].

The most basic options are call and put options which give the right to buy and sell, respectively, the underlying asset with a strike price. These options can be European or American which means that they can be exercised only at the expiry date or anytime before it, respectively. Under many models it is possible to derive formulas for the price of European options, but American options have to be priced usually numerically. There is a large number of methods to price options. In this paper we consider approach which solves numerically a partial integro-differential equation (PIDE) or inequality derived for the price of the option. The European options lead to equations while American options lead inequalities which have a form of a linear complementarity problem (LCP). In option pricing context, most common way to discretized the differential operators have been finite difference methods; see [1] and [37], for example. The treatment of the integral term associated to jumps in models is more challenging. Their discretization leads to full matrices. The multiplication of a vector by such a $n \times n$ matrix requires $\mathcal{O}(n^2)$ operations when implemented in a straightforward manner. With an implicit time discretization it is necessary to solve a problem with a full matrix which is computationally more expensive than a multiplication.

Here we review earlier works on pricing options under jump models using finite difference/element methods. Zhang in [39] priced American options with Merton's model. The integral term was treated explicitly in time while the differential terms were treated implicitly. This lead to a first-order accurate method in time and a stability restriction for the size of time step. Tavella and Randall in [37] considered pricing European options. They used an implicit time discretization which leads to the solution of problems with a full matrix. They proposed a stationary iterative method to solve these dense problems which converges fairly rapidly. Each iteration requires a solution with a tridiagonal matrix and a multiplication of a vector by a full matrix. Andersen and Andreasen in [4] proposed for European options an ADI-type operator splitting method with two fractional steps. In the first step the spatial differential operator is treated implicitly and the integral operator is treated explicitly and in the second step their roles are exchanged. This method is second-order accurate in time and unconditionally stable. Furthermore, they used a well-known fast technique for evaluating convolution integrals. It is uses the fast Fourier

transform (FFT) and it requires $\mathcal{O}(n \log n)$ operations. Thus, their approach lead to an essential reduction of computational cost.

In [14] d'Halluin, Forsyth, and Labahn price American options using a penalty method. For the resulting nonlinear nonsmooth problems they propose an approximate semismooth Newton method. They use the Crank-Nicolson method with adaptive time steps and FFT for evaluating the integral term. d'Halluin, Forsyth, and Vetzal in [15], develop a method for European options under Merton's and Kou's model. They use nonuniform spatial grids and in order to use FFT for evaluating the integral term on a uniform grid they perform interpolations back and forth nonuniform and uniform grids. The resulting dense linear systems are solved using the iterative method proposed in [37]. In [2] Almendral and Oosterlee price European options under Merton's and Kou's models. They use uniform grids and the BDF2 method for time discretization. The integrations are performed using FFT and the iterative method in [37] is employed for linear systems.

Cont and Voltchkova study pricing of European options under Merton's model and infinite jump intensity models in [13]. In the finite jump intensity case the time discretization is implicit for the spatial differential operator and explicit for the integral operator. This leads to a first-order accurate method in time with a stability restriction on the size of time steps. Briani, La Chioma, and Natalini propose a fully explicit time stepping for European options in [8] which leads to a more severe stability restriction. In [32] Matache, von Petersdorff, and Schwab price European options under models with finite and infinite jump intensity. They perform wavelet compression for the resulting full matrix and then the multiplication of a vector requires $\mathcal{O}(n \log n)$ operations. The time discretization is performed using the θ-scheme and the resulting linear problems are solved iteratively using the GMRES method. Matache, Schwab, and Wihler use a discontinuous Galerkin time discretization for pricing European options in [31]. Matache, Nitsche, and Schwab develop a method for American options which uses the implicit Euler time discretization and a fixed point iteration for the resulting LCPs in [30]. Recently Almendral and Oosterlee in [3] and Wang, Wan, and Forsyth in [38] consider pricing options under the CGMY process. In [3] the integral part is transform to a Volterra equation while in [38] the integral with a singular integrand is decomposed and resulting parts are treated separately. Both approaches lead to multiplication by full matrices which is performed using FFT.

In this paper, we price options under Kou's jump-diffusion model and the volatility can be given by a deterministic volatility model like in [4]. The spatial derivatives are discretized using central finite differences with possibly some locally added volatility in order to obtain an M-matrix. The time discretization is performed using the Rannacher time-stepping scheme [36]. The time-space grids are nonuniform with refinement near the strike price and expiry. The main contribution of this paper is the treatment of the integral term. It is based on the observation that due to the log-double-exponential form of the kernel in Kou's model it is possible to derive recursion formulas for evaluating the integrals. There are several benefits from this: The computational cost is reduced to $\mathcal{O}(n)$ operations per time step. The

resulting formulas are easy to implement and there is no difficulty to use nonuniform grids. With the FFT based approach and Kou's model it is difficult to maintain second-order convergence in space with a nonuniform grid [15] while there is no difficulty in this with the proposed approach. For American options we consider two alternative ways to treat the resulting LCPs. The first one is the operator splitting method introduced in [22], [23] and the second approach is the penalty method by d'Halluin, Forsyth, and Vetzal in [14]. Our numerical experiments demonstrate that the proposed approach is very fast as option prices with sufficient accuracy for most purposes can be obtained in a few milliseconds on a PC.

The outline of this paper is the following. Section 2 introduces Kou's model and option pricing problems for European and American put options. Section 3 describes the discretization of the partial integro-differential equation and the corresponding inequality using finite differences. Results are given on the accuracy of approximations. The recursion formulas are described for the integral term. Section 4 describes the stationary iteration for solving the dense linear systems resulting from pricing European options. Furthermore, an operators splitting method and a penalty method are considered for LCPs giving the price of American options. Section 5 presents numerical experiments with European and American options under a constant volatility and a deterministic volatility function. The paper ends with conclusions in Section 6.

2 Kou's model and option pricing problems

Kou's jump-diffusion model [27] for the value of the underlying asset x is given by

$$\frac{dx(t)}{x(t-)} = \mu dt + \sigma dW(t) + d\left(\sum_{j=1}^{N(t)} V_j\right), \tag{1}$$

where $W(t)$ is a standard Brownian motion and $N(t)$ is a Poisson process with rate λ. The set $\{V_j\}$ is a sequence of independent identically distributed random variables from a distribution with the log-double-exponential density

$$f(y) = \begin{cases} q\alpha_2 y^{\alpha_2 - 1}, & y < 1 \\ p\alpha_1 y^{-\alpha_1 - 1}, & y \geq 1, \end{cases} \tag{2}$$

where p, q, $\alpha_1 > 1$, and α_2 are positive constants such that $p + q = 1$. Following Andersen and Andreasen in [4], we allow the volatility σ in (1) to be a function of t and $x(t-)$. This volatility function makes it possible to calibrate the model to correspond the prevailing market prices.

The owner of a European put option has the right to sell the underlying asset for the strike price K regardless of the value of it at the expiry time T while the owner of an American put option can sell the asset for K anytime up to the expiry. The corresponding call options give the right to buy the underlying asset with the

strike price. In the rest of the paper, we consider the pricing of put options while call options can be priced in the same manner.

When the value of the underlying asset follows Kou's model the value of a European option v satisfies a final value problem defined by a partial integro-differential equation (PIDE)

$$v_t = Lv = -\frac{1}{2}(\sigma(t,x)x)^2 v_{xx} - (r - \lambda\zeta)xv_x + (r+\lambda)v - \lambda \int_{\mathbb{R}_+} v(t,xy) f(y)\, dy \qquad (3)$$

for all $(t,x) \in [0,T) \times \mathbb{R}_+$; see [12], for example. In (3), r is the risk free interest rate and the coefficient ζ is given by $\zeta = \frac{p\alpha_1}{\alpha_1 - 1} + \frac{q\alpha_2}{\alpha_2 + 1} - 1$. The final value of v is given by

$$v(T,x) = g(x), \qquad x \in \mathbb{R}_+, \qquad (4)$$

where $g(x)$ is the payoff function of the European option contract. For a put option, it is

$$g(x) = \max\{K - x, 0\}, \qquad (5)$$

where K is the strike price. The behavior of the value of the European put option on the boundaries is given by

$$v(t,0) = Ke^{-r(T-t)} \qquad \text{and} \qquad \lim_{x\to\infty} v(t,x) = 0, \qquad t \in [0,T]. \qquad (6)$$

The value of an American put option v satisfies a linear complementarity problem (LCP)

$$\begin{cases} (v_t - Lv) \geq 0, \quad v \geq g, \\ (v_t - Lv)\big(v - g\big) = 0 \end{cases} \qquad (7)$$

with the same final value as in (4); see [12], for example. The behavior of the value of the American put option on the boundaries is given by

$$v(t,0) = K \qquad \text{and} \qquad \lim_{x\to\infty} v(t,x) = 0, \qquad t \in [0,T]. \qquad (8)$$

3 Discretization

3.1 Truncation and time-space grid

For a finite difference discretization, we truncate the infinite domain $\mathbb{R}_+$ for x to be $[0,X]$ with a sufficiently large X. The truncation error with respect to X has been shown to decrease exponentially pointwise in [13] and exponentially in L_2-norm in [31]. For put options, we set the value of v at X to be zero.

We perform the discretization using a nonuniform grid. In option pricing, a more common approach has been to perform coordinate transforms and then use a uniform grid which approach effectively concentrates grid points in the original coordinates; see [11], [37], for example. For the computational domain $[0,T) \times [0,X]$, we define a $(m+1) \times (n+1)$ grid

$$(t_k, x_i) \in \{0 = t_0, \ldots, t_m = T\} \times \{0 = x_0, \ldots, x_n = X\}. \qquad (9)$$

In the following we propose one possible choice for grids. We remark that the numerical methods developed in the following sections are derived for generic nonuniform grids, that is, they do not assume this specific choice. The solution of the PIDE (3) changes more rapidly close to the expiry T and, hence, it is beneficial to take smaller time steps near the expiry. This is accomplished by choosing the approximation times t_k according to

$$t_k = \left(\frac{a^{-k/(m-2)} - 1}{a^{-1} - 1} \right) T, \qquad k = 0, 1, \ldots, m-4, \tag{10}$$

and

$$t_k = \left(\frac{a^{-(k+m-4)/(2m-4)} - 1}{a^{-1} - 1} \right) T, \qquad k = m-3, \ldots, l, \tag{11}$$

where α is a constant greater than one. By increasing α the time steps near the expiry become smaller. The reason to make the last four time steps especially small is due to the use of the Rannacher time-stepping explained in Section 3.4.

The space grid is defined by a grid function h as

$$x_i = h(s_i), \qquad s_i = i/n, \qquad i = 0, 1, \ldots, n. \tag{12}$$

It is beneficial to have a finer space grid near the strike price K. Such a grid is obtained by choosing h to be

$$h(s) = \left(1 + \frac{\sinh(\beta(s - \gamma))}{\sinh(\beta\gamma)} \right) K, \tag{13}$$

where the constant β is solved numerically from the equation $h(1) = X$. The constant γ controls the amount of refinement near the strike price K. By choosing $\gamma = i/n$ the grid point x_i will be exactly at K.

In the following we describe a central finite difference discretization of the spatial derivatives and a quadrature rule for the integral term. We show that these lead to an M-matrix and we propose recursion formulas for performing the numerical integrations. Then we describe the Rannacher time-stepping for the time discretization.

3.2 Finite difference discretization of spatial derivatives

Here we describe the discretization of the spatial derivative terms together with the zeroth-order term. This leads to a tridiagonal matrix which we denote by $\boldsymbol{D}$. We remark that due to the time dependent volatility $\sigma(t, x)$ the matrix $\boldsymbol{D}$ is also time dependent. We approximate the spatial derivative terms in (3) with central finite differences. We add artificial volatility, that is, artificial diffusion at some grid points if otherwise the discretization would lead $\boldsymbol{D}$ to have positive off-diagonal elements and, thus, to be a non M-matrix.

The central finite difference scheme for the derivative terms in (3) gives off-diagonal elements

$$\boldsymbol{D}_{i,i-1} = \frac{-(\hat{\sigma}_i(t)x_i)^2 + (r - \lambda\zeta)x_i\Delta x_i}{\Delta x_{i-1}(\Delta x_{i-1} + \Delta x_i)} \tag{14}$$

and

$$\boldsymbol{D}_{i,i+1} = \frac{-(\hat{\sigma}_i(t)x_i)^2 - (r-\lambda\zeta)x_i\Delta x_{i-1}}{\Delta x_i(\Delta x_{i-1}+\Delta x_i)}, \tag{15}$$

where $\Delta x_{i-1} = x_i - x_{i-1}$ and $\Delta x_i = x_{i+1} - x_i$. The modified volatility $\hat{\sigma}_i(t)$ in (14) and (15) is defined by

$$(\hat{\sigma}_i(t))^2 = \max\left\{(\sigma(t,x_i))^2,\ (r-\lambda\zeta)\frac{\Delta x_i}{x_i},\ -(r-\lambda\zeta)\frac{\Delta x_{i-1}}{x_i}\right\}. \tag{16}$$

In the case of a uniform grid, it is easy to show that including artificial volatility is equivalent with the use of a combination of the central finite difference and an one-sided finite difference for the first-order spatial derivative.

The diagonal element of $\boldsymbol{D}$ includes the sum of the off-diagonal elements with a minus sign. Furthermore, $r+\lambda$ is added to it due to the zeroth-order term in (3). Thus, the ith diagonal element is given by

$$\boldsymbol{D}_{i,i} = r + \lambda - \boldsymbol{D}_{i,i-1} - \boldsymbol{D}_{i,i+1}. \tag{17}$$

The following result states that the finite difference discretization is second-order accurate under suitable assumptions. Due to a nonuniform grid, we require the grid function h to be sufficiently smooth; see [29], for example. The function h defined by (13) satisfies this requirement. The following results assume that no artificial diffusion is added. If it is necessary to add some diffusion then often the order of accuracy is reduced.

Theorem 1 *For a given $t \in [0,T)$ let $v(t,x)$ be three times continuously differentiable with respect to x in $[0,X]$ and let $v_{xxxx}(t,x)$ be bounded for $x \in (0,X)$. Furthermore, let the grid function h be continuously differentiable in $[0,1]$ and let h_{xx} be bounded in $(0,1)$. If $\hat{\sigma}_i(t) = \sigma(t,x_i)$ then the truncation error of the finite difference approximation at (t,x_i) is $\mathcal{O}(\max\{(\Delta x_{i-1})^2, (\Delta x_i)^2, (1/n)^2\})$.*

Proof. Taylor's theorem gives us the expressions

$$\begin{aligned} v(t,x_{i-1}) = v(t,x_i) - v_x(t,x_i)\Delta x_{i-1} + \frac{1}{2}v_{xx}(t,x_i)(\Delta x_{i-1})^2 - \frac{1}{6}v_{xxx}(t,x_i)(\Delta x_{i-1})^3 \\ + \mathcal{O}((\Delta x_{i-1})^4) \end{aligned}$$

and

$$v(t,x_{i+1}) = v(t,x_i) + v_x(t,x_i)\Delta x_i + \frac{1}{2}v_{xx}(t,x_i)(\Delta x_i)^2 + \frac{1}{6}v_{xxx}(t,x_i)(\Delta x_i)^3 + \mathcal{O}((\Delta x_i)^4).$$

In the following, we assume that $\hat{\sigma}_i(t) = \sigma(t,x_i)$. For the finite difference discretization we obtain using the above expressions a truncation error

$$\begin{aligned} &\boldsymbol{D}_{i,i-1}v(t,x_{i-1}) + \boldsymbol{D}_{i,i}v(t,x_i) + \boldsymbol{D}_{i,i+1}v(t,x_{i-1}) \\ &- \left(-\frac{1}{2}(\sigma(t,x_i)x_i)^2 v_{xx}(t,x_i) - (r-\lambda\zeta)x_i v_x(t,x_i) + (r+\lambda)v(x_i)\right) \\ &= -\frac{1}{6}\left((\hat{\sigma}_i(t)x_i)^2(\Delta x_i - \Delta x_{i-1}) + (r-\lambda\zeta)x_i\Delta x_{i-1}\Delta x_i\right)v_{xxx}(t,x_i) \\ &+ \mathcal{O}(\max\{(\Delta x_{i-1})^2, (\Delta x_i)^2\}). \end{aligned}$$

Since $\Delta x_{i-1}\Delta x_i$ is dominated by $\mathcal{O}(\max\{(\Delta x_{i-1})^2, (\Delta x_i)^2\})$, it is sufficient to study only the magnitude of the term $\Delta x_i - \Delta x_{i-1}$. We have

$$\Delta x_i - \Delta x_{i-1} = x_{i-1} - 2x_i + x_{i+1} = h((i-1)/n) - 2h(i/n) + h((i+1)/n).$$

Taylor's theorem applied to $h((i-1)/n)$ and $h((i-1)/n)$ yields together with easy manipulations that $\Delta x_i - \Delta x_{i-1} = \mathcal{O}((1/n)^2)$. Substituting this to the above truncation error completes the proof. □

The following result states that the matrix $\boldsymbol{D}$ has desirable properties.

Theorem 2 *For a positive interest rate $r > 0$ the matrix $\boldsymbol{D} - \lambda\boldsymbol{I}$ is a strictly diagonally dominant M-matrix, that is,*

$$\sum_j (\boldsymbol{D} - \lambda\boldsymbol{I})_{i,j} > 0 \qquad \textit{and} \qquad \boldsymbol{D}_{i,j} \le 0 \quad \forall j \neq i$$

for $i = 1, \ldots, n-1$.

Proof. First we show that the off-diagonal $\boldsymbol{D}_{i,i-1}$ in (14) is non positive. According to the definition and (16), we have

$$\begin{aligned}\boldsymbol{D}_{i,i-1} &= \frac{-\max\left\{(\sigma(t,x_i))^2,\ (r-\lambda\zeta)\frac{\Delta x_i}{x_i},\ -(r-\lambda\zeta)\frac{\Delta x_{i-1}}{x_i}\right\} x_i^2 + (r-\lambda\zeta)x_i\Delta x_i}{\Delta x_{i-1}(\Delta x_{i-1} + \Delta x_i)} \\ &= \frac{-\max\left\{(\sigma(t,x_i)x_i)^2 - (r-\lambda\zeta)x_i\Delta x_i,\ 0,\ -(r-\lambda\zeta)x_i(\Delta x_{i-1} + \Delta x_i)\right\}}{\Delta x_{i-1}(\Delta x_{i-1} + \Delta x_i)} \le 0.\end{aligned}$$

Similarly the off-diagonal $\boldsymbol{D}_{i,i+1}$ in (15) can be shown to be non positive. From the definition of $\boldsymbol{D}_{i,i}$ in (17) it is easy to see that the matrix $\boldsymbol{D} - \lambda\boldsymbol{I}$ has positive diagonal. Furthermore, it is strictly diagonal dominant, since

$$\sum_j (\boldsymbol{D} - \lambda\boldsymbol{I})_{i,j} \ge r > 0.$$

This completes the proof. □

3.3 Recursion formulas for approximating integrals

The discretization of the integral term

$$-\lambda \int_{\mathbb{R}_+} v(t, xy) f(y)\, dy \tag{18}$$

in (3) leads to a full matrix which approximation we denote by $\boldsymbol{J}$. In the following solution procedures, we need to multiply vectors by $\boldsymbol{J}$ which corresponds to the evaluation of the integral term at each grid point. We do not form the matrix, but rather device a fast way to evaluate the integral term approximately.

By making the change of variable $y = z/x$, we obtain

$$I = \int_{\mathbb{R}_+} v(t, xy) f(y)\, dy = \int_{\mathbb{R}_+} v(t, z) f(z/x)/x\, dz. \tag{19}$$

We decompose the integral as $I = I^- + I^+$, where

$$I^- = \int_0^x v(t, z) f(z/x)/x\, dz = q\alpha_2 x^{-\alpha_2} \int_0^x v(t, z)\, z^{\alpha_2 - 1} dz$$

and

$$I^+ = \int_x^\infty v(t, z) f(z/x)/x\, dz = p\alpha_1 x^{\alpha_1} \int_x^\infty v(t, z)\, z^{-\alpha_1 - 1} dz.$$

In the following, we consider the approximation of the integrals of the type I^- while the other ones can the treated in the same way. At each grid point x_i, $i = 1, \ldots, n-1$, we need to approximate

$$I_i^- = q\alpha_2 x_i^{-\alpha_2} \int_0^{x_i} v(t, z)\, z^{\alpha_2 - 1} dz = \sum_{j=0}^{i-1} I_{i,j}^-, \tag{20}$$

where

$$I_{i,j}^- = q\alpha_2 x_i^{-\alpha_2} \int_{x_j}^{x_{j+1}} v(t, z)\, z^{\alpha_2 - 1} dz. \tag{21}$$

With second-order accurate finite differences it is typical to use the linear interpolation for v between grid points. This interpolation of $v(t, x)$ leads to the approximation

$$I_i^- \approx A_i^- = \sum_{j=0}^{i-1} A_{i,j}^-, \tag{22}$$

where

$$A_{i,j}^- = q\alpha_2 x_i^{-\alpha_2} \int_{x_j}^{x_{j+1}} \left(\frac{x_{j+1} - z}{\Delta x_j} v(t, x_j) + \frac{z - x_j}{\Delta x_j} v(t, x_{j+1}) \right) z^{\alpha_2 - 1} dz. \tag{23}$$

By performing the integration we obtain

$$\begin{aligned} A_{i,j}^- = \frac{q\, x_i^{-\alpha_2}}{(\alpha_2 + 1)\Delta x_j} \big[& \left(x_{j+1}^{\alpha_2 + 1} - (x_{j+1} + \alpha_2 \Delta x_j) x_j^{\alpha_2} \right) v(t, x_j) \\ & + \left(x_j^{\alpha_2 + 1} - (x_j - \alpha_2 \Delta x_j) x_{j+1}^{\alpha_2} \right) v(t, x_{j+1}) \big] \end{aligned} \tag{24}$$

for $j = 1, \ldots, i-1$ and

$$A_{i,0}^- = \frac{q\, x_i^{-\alpha_2}}{\alpha_2 + 1} \left[x_1^{\alpha_2} v(t, 0) + \alpha_2 x_1^{\alpha_2} v(t, x_1) \right]. \tag{25}$$

The computation of all A_i^-s, $i = 1, \ldots, n-1$, using (22), (24), and (25) requires $\mathcal{O}(n^2)$ operations. This high computational cost would slow down solution procedures essentially on finer grids. One possibility to speed up the computations is to

transform the integral in (19) to a convolution form and then use FFT in its evaluation. This approach has be advocated by Andersen and Andreasen in [4], Almendral and Oosterlee in [2], and d'Halluin, Forsyth, and Vetzal in [15], for example. A nonuniform grid and the usual discontinuity of the convolution kernel makes it difficult to maintain second-order accuracy with this approach. Furthermore, this approach is not easy to implement.

Fortunately with a log-double-exponential f it is possible to derive a fast and easy to implement recursion formula for computing the approximations A_i^- and also the corresponding approximations for I^+. In a different context, a similar formula has been employed in [5], for example. The observation that

$$A_{i+1,j}^- = \left(\frac{x_{i+1}}{x_i}\right)^{-\alpha_2} A_{i,j}^-$$

for $j = 0, \ldots, i-1$ together with (22) lead to the recursion formula

$$A_{i+1}^- = \left(\frac{x_i}{x_{i+1}}\right)^{\alpha_2} A_i^- + A_{i+1,i}^-. \tag{26}$$

Thus, this reduces the computational cost to be $\mathcal{O}(n)$ operations for all A_i^-s, $i = 1, \ldots, n-1$.

Similarly, for analogously defined A_i^+s we obtain a recursion formula

$$A_{i-1}^+ = \left(\frac{x_{i-1}}{x_i}\right)^{\alpha_1} A_i^+ + A_{i-1,i-1}^+, \quad i = 2, \ldots, n-1, \tag{27}$$

where we have

$$\begin{aligned} A_{i,j}^+ = \frac{p\, x_i^{\alpha_1}}{(\alpha_1 - 1)\Delta x_j} \big[&\left(x_{j+1}^{1-\alpha_1} - (x_{j+1} - \alpha_1 \Delta x_j) x_j^{-\alpha_1}\right) v(t, x_j) \\ &+ \left(x_j^{1-\alpha_1} - (x_j + \alpha_1 \Delta x_j) x_{j+1}^{-\alpha_1}\right) v(t, x_{j+1}) \big] \end{aligned} \tag{28}$$

for $j = i, \ldots, n-1$. The first value to be computed is A_{n-1}^+ and it is given by

$$A_{n-1}^+ = \frac{p\, x_{n-1}^{\alpha_1}}{(\alpha_1 - 1)\Delta x_{n-1}} \left(x_n^{1-\alpha_1} - (x_n - \alpha_1 \Delta x_{n-1}) x_{n-1}^{-\alpha_1}\right) v(t, x_{n-1}).$$

Now the multiplication of a vector $\boldsymbol{v}$ by $\boldsymbol{J}$ is defined by

$$(\boldsymbol{J}\boldsymbol{v})_i = -\lambda \left(A_i^- + A_i^+\right) \tag{29}$$

for $i = 1, \ldots, n-1$.

The following result states that our approximation of the integral is second-order accurate under a suitable regularity assumption on v.

Theorem 3 *For a given $t \in [0, T)$ let $v(t, x)$ be twice continuously differentiable with respect to x in $[0, X]$ and $\alpha_2 > 0$ and $\alpha_1 > 1$. The difference between $I_i^- + I_i^+$ and $A_i^- + A_i^+$ is $\mathcal{O}(\max_{0 \le j < n} (\Delta x_j)^2)$.*

Proof. In the following, we show that $|I_i^- - A_i^-| = \mathcal{O}(\max_{0\le j<i}(\Delta x_j)^2)$. It can be shown in the same way that $|I_i^+ - A_i^+| = \mathcal{O}(\max_{i\le j<n}(\Delta x_j)^2)$. Together from these the result follows.

We begin by considering an interval $[x_j, x_{j+1}]$ for some $j \in \{0, \dots, i-1\}$. The approximation in this interval is based on the linear interpolation of $v(t,z)$ which we denote by $\bar{v}(z)$. A well-known result states that under the regularity assumption on v the interpolation error for any $z \in [x_j, x_{j+1}]$ is

$$v(t,z) - \bar{v}(z) = v_{xx}(t,\xi)(z - x_j)(z - x_{j+1})/2$$

for some $\xi \in [x_j, x_{j+1}]$. Thus, we have

$$|v(t,z) - \bar{v}(z)| \le \frac{1}{8} C (\Delta x_j)^2,$$

where $C = \max_{x\in[0,X]} v_{xx}(t,\xi)$. Using this, we obtain

$$\begin{aligned}
|I_{i,j}^- - A_{i,j}^-| &= q\alpha_2 x_i^{-\alpha_2} \int_{x_j}^{x_{j+1}} |v(t,z) - \bar{v}(z)|\, z^{\alpha_2 - 1} dz \\
&\le \frac{1}{8} C q \alpha_2 x_i^{-\alpha_2} (\Delta x_j)^2 \int_{x_j}^{x_{j+1}} z^{\alpha_2 - 1} dz \\
&\le \frac{1}{8} C q\, x_i^{-\alpha_2} (\Delta x_j)^2 \left(x_{j+1}^{\alpha_2} - x_j^{\alpha_2} \right).
\end{aligned}$$

Summing over the intervals, we get

$$\begin{aligned}
|I_i^- - A_i^-| &\le \sum_{j=0}^{i-1} |I_{i,j}^- - A_{i,j}^-| \le \frac{1}{8} C q\, x_i^{-\alpha_2} \sum_{j=0}^{i-1} (\Delta x_j)^2 \left(x_{j+1}^{\alpha_2} - x_j^{\alpha_2} \right) \\
&\le \frac{1}{8} C q\, x_i^{-\alpha_2} \max_{0\le j<i} (\Delta x_j)^2 \sum_{j=0}^{i-1} \left(x_{j+1}^{\alpha_2} - x_j^{\alpha_2} \right) = \frac{1}{8} C q \max_{0\le j<i} (\Delta x_j)^2.
\end{aligned}$$

This completes the proof. □

The following results shows that the matrix $\boldsymbol{J} + \lambda \boldsymbol{I}$ has good properties.

Theorem 4 *The matrix $\boldsymbol{J} + \lambda \boldsymbol{I}$ is a diagonally dominant Z-matrix with a non negative diagonal, that is,*

$$\sum_j (\boldsymbol{J} + \lambda \boldsymbol{I})_{i,j} \ge 0 \qquad \textit{and} \qquad \boldsymbol{J}_{i,j} \le 0 \quad \forall j \ne i$$

for $i = 1, \dots, n-1$.

Proof. For $\lambda = 0$ we have $\boldsymbol{J} + \lambda \boldsymbol{I} = \boldsymbol{0}$ and, thus, it trivially satisfies the result. Let $\lambda > 0$ in the following.

The off-diagonal elements of $\boldsymbol{J}$ are negative, since the coefficients for $v(t, x_j)$ and $v(t, x_{j+1})$ appearing in (24) and (28) can be shown to be positive, and the result is multiplied by $-\lambda$.

The row sums of $\boldsymbol{J} + \lambda \boldsymbol{I}$ can be obtained by multiplying a vector $\boldsymbol{u}$ containing ones. In this case, for A_i^- the definitions (22) and (23) give

$$A_i^- = q\alpha_2 x_i^{-\alpha_2} \int_0^{x_i} z^{\alpha_2 - 1} dz = q.$$

Similarly can be shown that $A_i^+ \leq p$. Thus, we have

$$(\boldsymbol{J}\boldsymbol{u})_i = -\lambda \left(A_i^- + A_i^+ \right) \geq -\lambda(q + p) = -\lambda.$$

From this the first inequality in the results follows which completes the proof. □

Theorems 2 and 4 yield the following result.

Corollary 1 *For a positive interest rate $r > 0$ the matrix $\boldsymbol{A} = \boldsymbol{D} + \boldsymbol{J}$ is a strictly diagonally dominant M-matrix.*

Many recursion formulas are notoriously unstable and, thus, usually unusable. The following results guarantees that with the developed recursion formulas the accumulating error grows at most linearly with respect to the number of grid points n. This means that the formulas are sufficiently stable to be used in the computations.

Theorem 5 *Let the additional error introduced at each recursion step* (26) *and* (27) *be at most δ then the error on the computed value of $A_i^- + A_i^+$ is less than $n\delta$.*

Proof. We denote the maximum error in the computed value of A_i^- by δ_i^-. At most the first value A_1^- can have the error $\delta_1^- = \delta$. From (26) it follows that the maximum errors are

$$\begin{aligned}
\delta_2^- &= \left(\frac{x_1}{x_2}\right)^{\alpha_2} \delta_1^- + \delta = \left[\left(\frac{x_1}{x_2}\right)^{\alpha_2} + 1\right] \delta \\
\delta_3^- &= \left(\frac{x_2}{x_3}\right)^{\alpha_2} \delta_2^- + \delta = \left[\left(\frac{x_1}{x_3}\right)^{\alpha_2} + \left(\frac{x_2}{x_3}\right)^{\alpha_2} + 1\right] \delta \\
&\vdots \\
\delta_i^- &= \left[\sum_{j=1}^{i} \left(\frac{x_j}{x_i}\right)^{\alpha_2}\right] \delta < i\delta.
\end{aligned}$$

In the same way, it can be shown that the error of the computed value of A_i^+ is less than $(n - i)\delta$. By summing the maximum errors the result follows. □

3.4 Rannacher time-stepping scheme

After the discretization of the spatial derivative terms and the integral term in (3), for the value of the European option we obtain a semi-discrete linear problem

$$\boldsymbol{v}_t = \boldsymbol{A}\boldsymbol{v} + \boldsymbol{f} = \boldsymbol{D}\boldsymbol{v} + \boldsymbol{J}\boldsymbol{v} + \boldsymbol{f}, \tag{30}$$

where the vector $\boldsymbol{f}$ is due to the nonhomogenous boundary value $v(t,0)$. For the value of the American option we obtain a semi-discrete LCP

$$\begin{cases} \left(\boldsymbol{v}_t - \boldsymbol{A}\boldsymbol{v} - \boldsymbol{f}\right) \geq \boldsymbol{0}, \quad \boldsymbol{v} \geq \boldsymbol{g}, \\ \left(\boldsymbol{v}_t - \boldsymbol{A}\boldsymbol{v} - \boldsymbol{f}\right)^T \left(\boldsymbol{v} - \boldsymbol{g}\right) = 0. \end{cases} \tag{31}$$

The vector $\boldsymbol{g}$ contains the values of the payoff function g at the grid points, that is, $\boldsymbol{g}_i = g(x_i)$, $i = 1, \ldots, n-1$. For both problems (30) and (31) the final value is $\boldsymbol{v}(T) = \boldsymbol{g}$.

Usually option pricing problems have a nonsmooth payoff function and, thus, also a nonsmooth final value. In such cases the popular Crank-Nicolson method can lead to a numerical solution with oscillations due to the lack of L-stability. In order to obtain numerical solution without oscillations we propose to use the Rannacher time-stepping scheme [36]. It has been applied in the option pricing in [18], [19], [24], [35], for example.

The Rannacher time-stepping scheme performs a few first time steps with the implicit Euler method and after that it uses the Crank-Nicolson method. This leads to good damping properties and second-order accuracy. For the semi-discrete linear problem (30) the Rannacher time-stepping scheme reads

$$\boldsymbol{B}^{(k)} \boldsymbol{v}^{(k)} = \boldsymbol{b}^{(k)}, \qquad k = m-1, \ldots, 0, \tag{32}$$

where

$$\boldsymbol{B}^{(k)} = \boldsymbol{I} + \theta_k \Delta t_k \boldsymbol{A}, \qquad \boldsymbol{b}^{(k)} = \left(\boldsymbol{I} - (1-\theta_k)\Delta t_k \boldsymbol{A}\right) \boldsymbol{v}^{(k+1)} + \boldsymbol{f}^{(k)}, \tag{33}$$

and

$$\boldsymbol{f}^{(k)} = -\Delta t_k \boldsymbol{D}_{1,0} \left(\theta_k v(t_k, 0) + (1-\theta_k) v(t_{k+1}, 0)\right) \boldsymbol{u}^1. \tag{34}$$

In (33) and (34), the matrix $\boldsymbol{A}$ and the coefficient $\boldsymbol{D}_{1,0}$ defined by (14) are computed using the volatility at the time $t_{k+1} - \theta_k \Delta t_k$. The vector $\boldsymbol{u}^1$ is such that its first component is one and others are zero. The parameter θ_k is defined by

$$\theta_k = \begin{cases} 1, & k = m-1, m-2, m-3, m-4, \\ \frac{1}{2}, & k = m-5, \ldots, 0. \end{cases} \tag{35}$$

For the semi-discrete LCP (31) the time-stepping scheme reads

$$\begin{cases} \left(\boldsymbol{B}^{(k)} \boldsymbol{v}^{(k)} - \boldsymbol{b}^{(k)}\right) \geq \boldsymbol{0}, \quad \boldsymbol{v}^{(k)} \geq \boldsymbol{g}, \\ \left(\boldsymbol{B}^{(k)} \boldsymbol{v}^{(k)} - \boldsymbol{b}^{(k)}\right)^T \left(\boldsymbol{v}^{(k)} - \boldsymbol{g}\right) = 0, \end{cases} \qquad k = m-1, \ldots, 0. \tag{36}$$

4 Solution methods

4.1 Iterative method for European options

We use an iterative method propose by Tavella and Randall in [37]. d'Halluin, Forsyth, and Vetzal analyzed its convergence in [15]. For our presentation we adopt

the matrix formulation presented by Almendral and Oosterlee in [2]. We solve the resulting systems of linear equations (32) using a stationary iterative method based on a regular splitting of the matrix $\boldsymbol{B}^{(k)}$ given by

$$\boldsymbol{B}^{(k)} = \boldsymbol{T} - \boldsymbol{R}, \quad \text{where} \quad \boldsymbol{T} = \boldsymbol{I} + \theta_k \Delta t_k \boldsymbol{D} \quad \text{and} \quad \boldsymbol{R} = -\theta_k \Delta t_k \boldsymbol{J}. \tag{37}$$

With this choice the matrix $\boldsymbol{T}$ is tridiagonal. Then the iterative method for the linear system $\boldsymbol{B}^{(k)}\boldsymbol{v} = \boldsymbol{b}^{(k)}$ reads

$$\boldsymbol{v}^{l+1} = \boldsymbol{T}^{-1}\left(\boldsymbol{b}^{(k)} + \boldsymbol{R}\boldsymbol{v}^{l}\right), \qquad l = 0, 1, \ldots, \tag{38}$$

where $\boldsymbol{v}^0$ is the initial guess taken to be $\boldsymbol{v}^{(k+1)}$. Each iteration requires a solution with the tridiagonal $\boldsymbol{T}$ and the multiplication of a vector by $\boldsymbol{J}$ which corresponds to the evaluation of the integral term in the PIDE (3). The problem with $\boldsymbol{T}$ can be solved efficiently using $\boldsymbol{LU}$ decomposition. For the multiplication by $\boldsymbol{J}$ we can employ the fast recursion formulas in Section 3.3. Thus, one iteration requires $\mathcal{O}(n)$ operations.

After l iterations the residual vector $\boldsymbol{r}^{l+1}$ is given by

$$\boldsymbol{r}^{l+1} = \boldsymbol{b}^{(k)} - \boldsymbol{B}^{(k)}\boldsymbol{v}^{l+1} = \boldsymbol{b}^{(k)} - \boldsymbol{T}\,\boldsymbol{T}^{-1}\left(\boldsymbol{b}^{(k)} + \boldsymbol{R}\boldsymbol{v}^{l}\right) + \boldsymbol{R}\boldsymbol{v}^{l+1} = \boldsymbol{R}\boldsymbol{v}^{l+1} - \boldsymbol{R}\boldsymbol{v}^{l}. \tag{39}$$

The following result [15] shows that the iteration converges rapidly.

Theorem 6 *Let $\boldsymbol{e}^l$ be the error $\boldsymbol{e}^l = \boldsymbol{v}^l - \left(\boldsymbol{B}^{(k)}\right)^{-1}\boldsymbol{b}^{(k)}$ at the lth iteration. Then the error $\boldsymbol{e}^{l+1}$ at the iteration $l+1$ satisfies*

$$\left\|\boldsymbol{e}^{l+1}\right\|_\infty \leq \frac{\theta_k \Delta t_k \lambda}{1 + \theta_k \Delta t_k (r + \lambda)} \left\|\boldsymbol{e}^{l}\right\|_\infty. \tag{40}$$

Proof. Substituting $\boldsymbol{v}^l = \boldsymbol{e}^l + \left(\boldsymbol{B}^{(k)}\right)^{-1}\boldsymbol{b}^{(k)}$ and $\boldsymbol{v}^{l+1} = \boldsymbol{e}^{l+1} + \left(\boldsymbol{B}^{(k)}\right)^{-1}\boldsymbol{b}^{(k)}$ to (38). Then rearranging the terms gives us

$$\boldsymbol{T}\boldsymbol{e}^{l+1} = \boldsymbol{R}\boldsymbol{e}^{l}. \tag{41}$$

By using the properties of the matrix $\boldsymbol{D}$ it is easy to show that

$$\|\boldsymbol{T}\boldsymbol{w}\|_\infty \geq \left[1 + \theta_k \Delta t_k (r + \lambda)\right] \|\boldsymbol{w}\|_\infty$$

for any vector $\boldsymbol{w}$. Furthermore, it can be easily seen that

$$\|\boldsymbol{R}\boldsymbol{w}\|_\infty \leq \theta_k \Delta t_k \lambda \|\boldsymbol{w}\|_\infty$$

any vector $\boldsymbol{w}$. By using these inequalities to the terms in (41) and dividing by $1 + \theta_k \Delta t_k (r + \lambda)$ the inequality (40) is obtained. □

4.2 Operator splitting method for American options

The operator splitting method was introduced in [22] and further studied in [23], [24]. In order to describe this method we restate the LCP at the kth time step with a Lagrange multiplier $\boldsymbol{\lambda}^{(k)}$ as

$$\boldsymbol{B}^{(k)}\boldsymbol{v}^{(k)} = \boldsymbol{b}^{(k)} + \Delta t_k \boldsymbol{\lambda}^{(k)}, \quad \boldsymbol{v}^{(k)} \geq \boldsymbol{g}, \quad \boldsymbol{\lambda}^{(k)} \geq \boldsymbol{0}, \quad \left(\boldsymbol{\lambda}^{(k)}\right)^T \left(\boldsymbol{v}^{(k)} - \boldsymbol{g}\right) = 0.$$

The operator splitting method has the following two fractional time steps:

$$\boldsymbol{B}^{(k)}\hat{\boldsymbol{v}}^{(k)} = \boldsymbol{b}^{(k)} + \Delta t_k \boldsymbol{\lambda}^{(k+1)} \tag{42}$$

and

$$\boldsymbol{v}^{(k)} - \hat{\boldsymbol{v}}^{(k)} = \Delta t_k(\boldsymbol{\lambda}^{(k)} - \boldsymbol{\lambda}^{(k+1)}), \quad \boldsymbol{v}^{(k)} \geq \boldsymbol{g}, \quad \boldsymbol{\lambda}^{(k)} \geq \boldsymbol{0}, \quad \left(\boldsymbol{\lambda}^{(k)}\right)^T \left(\boldsymbol{v}^{(k)} - \boldsymbol{g}\right) = 0. \tag{43}$$

The first step (42) requires the solution of the linear system which can be efficiently obtained using the iteration (38). In the second step (43) the inequalities can be solved easily componentwise manner. Thus, the second step is computationally very cheap and the first step determines the computational cost of the solution procedure.

The above operator splitting introduces additional error to the numerical solution. In [23], it was shown that when the Crank-Nicolson method with a constant time step Δt is used for the time discretization and $\boldsymbol{B} = \boldsymbol{B}^{(k)}$ is an M-matrix then under the assumption that the time oscillations of the Lagrange multiplier are bounded the difference between the solutions obtained using the unsplit and split versions is $\mathcal{O}\left((\Delta t)^2\right)$. Thus, this result says that the order of accuracy of the operator splitting method is the same as for the fully discretized LCP when both are based on the Crack-Nicolson method.

4.3 Penalty method for American options

As a second approach for solving the LCPs we consider the penalty method proposed by d'Halluin, Forsyth, and Vetzal in [14]. This leads to the solution of nonlinear problems

$$\boldsymbol{B}^{(k)}\boldsymbol{v}^{(k)} = \boldsymbol{b}^{(k)} + \frac{1}{\epsilon}\max\left\{\boldsymbol{g} - \boldsymbol{v}^{(k)},\ \boldsymbol{0}\right\}, \tag{44}$$

where ϵ is a small penalty parameter. One possibility is to solve the nonlinear and nonsmooth problems (44) using a semismooth Newton method in [21]. We choose the initial guess to be $\boldsymbol{v}^0 = \boldsymbol{v}^{(k+1)}$. Then the $(l+1)$th iterant would be given by

$$\boldsymbol{v}^{l+1} = \boldsymbol{v}^l + \boldsymbol{d}^l, \tag{45}$$

where the vector $\boldsymbol{d}^l$ is the solution of the system of linear equations

$$\boldsymbol{C}^l\boldsymbol{d}^l = \boldsymbol{b}^{(k)} + \frac{1}{\epsilon}\max\left\{\boldsymbol{g} - \boldsymbol{v}^l,\ \boldsymbol{0}\right\} - \boldsymbol{B}^{(k)}\boldsymbol{v}^l. \tag{46}$$

The matrix $\boldsymbol{C}^l$ in (46) belongs to the generalized Jacobian [10] at x^l and it is chosen to be $\boldsymbol{C}^l = \boldsymbol{B}^{(k)} + \boldsymbol{P}^l$, where $\boldsymbol{P}^l$ is a diagonal matrix with its diagonal defined by

$$\boldsymbol{P}^l_{i,i} = \begin{cases} \frac{1}{\epsilon}, & \text{if} \quad \boldsymbol{v}^l_i < \boldsymbol{g}_i, \\ 0, & \text{otherwise.} \end{cases} \tag{47}$$

The matrix $\boldsymbol{C}^l$ is a full matrix and the systems of linear equations in (46) could be solved using an iteration similar to one in Section 4.1. Instead of this we employ an approximate semismooth Newton method proposed by d'Halluin, Forsyth, and Vetzal in [14] which approximates $\boldsymbol{C}^l$ with a tridiagonal matrix $\widehat{\boldsymbol{C}}^l = \boldsymbol{T} + \boldsymbol{P}^l$, where $\boldsymbol{T}$ is defined in (37). Replacing $\boldsymbol{C}^l$ by $\widehat{\boldsymbol{C}}^l$ in (45) and eliminating $\boldsymbol{d}^l$ gives us

$$\begin{aligned} \widehat{\boldsymbol{C}}^l \boldsymbol{v}^{l+1} &= \widehat{\boldsymbol{C}}^l \boldsymbol{v}^l + \boldsymbol{b}^{(k)} + \frac{1}{\epsilon} \max\left\{\boldsymbol{g} - \boldsymbol{v}^l,\, \boldsymbol{0}\right\} - \boldsymbol{B}^{(k)} \boldsymbol{v}^l \\ &= (\boldsymbol{T} + \boldsymbol{P}^l)\boldsymbol{v}^l + \boldsymbol{b}^{(k)} + \boldsymbol{P}^l(\boldsymbol{g} - \boldsymbol{v}^l) - (\boldsymbol{T} - \boldsymbol{R})\boldsymbol{v}^l \\ &= \boldsymbol{b}^{(k)} + \boldsymbol{R}\boldsymbol{v}^l + \boldsymbol{P}^l \boldsymbol{g}, \end{aligned} \tag{48}$$

where we have used the identities

$$\boldsymbol{B}^{(k)} = \boldsymbol{T} - \boldsymbol{R} \qquad \text{and} \qquad \boldsymbol{P}^l(\boldsymbol{g} - \boldsymbol{v}^l) = \frac{1}{\epsilon} \max\left\{\boldsymbol{g} - \boldsymbol{v}^l,\, \boldsymbol{0}\right\}. \tag{49}$$

Multiplying (48) by $\left(\widehat{\boldsymbol{C}}^l\right)^{-1}$ leads to

$$\boldsymbol{v}^{l+1} = \left(\boldsymbol{T} + \boldsymbol{P}^l\right)^{-1} \left(\boldsymbol{b}^{(k)} + \boldsymbol{R}\boldsymbol{v}^l + \boldsymbol{P}^l \boldsymbol{g}\right), \tag{50}$$

which includes two additional terms with $\boldsymbol{P}^l$ when compared to (38). In [14] it has been shown that the iteration (50) converges to the unique solution of (44).

The penalty formulation (44) leads to a solution which always violates the early exercise constraint by a small amount controlled by the penalty parameter ϵ. By adding a Lagrange multiplier to the formulation like in [21], [25], and [26] it is possible to guarantee that the constraint is always satisfied.

5 Numerical results

5.1 European put option

In our first numerical example, we price a European put option defined by the parameters

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

These same parameters were used by Almendral and Oosterlee in [2]. Furthermore, they provide an integral presentation for the option price. Our reference price

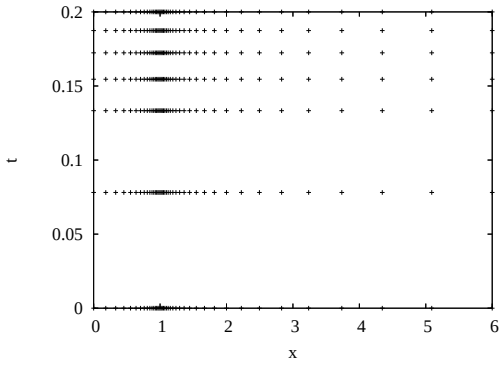

Figure 1: The nonuniform time-space grid with $m = 6$ and $n = 40$.

Table 1: The pricing error, the ratio of l_2 errors, the total number of iterations, and the CPU time in milliseconds for the European put option.

m	n	error at K	rate	iter.	time
6	40	-5.1×10^{-4}		19	0.1
10	80	-1.2×10^{-4}	4.3	28	0.2
18	160	-2.9×10^{-5}	4.0	45	1.0
34	320	-7.4×10^{-6}	4.0	70	2.9
66	640	-1.9×10^{-6}	4.0	132	10.1

$v^*(0,1) = 0.042647805$ is computed using their formula with Simpson's quadrature rule in the interval $[0, 100]$ with 10000 subintervals. We choose the truncation boundary to be at $X = 6$. The time-space grids are generated using $\alpha = 4$ in (10), (11), and $\gamma = 0.4$ in (12). Figure 1 shows a 5×41 time-space grid.

Table 1 gives the pricing errors at the money, that is, at $x = K$. The rate in the table is defined as the ratio between the error on the coarser grid and the error on the current grid. The reported numbers of iterations (38) are obtained using the stopping criterion $\|\boldsymbol{r}^{l+1}\|_2 < 10^{-9}\|\boldsymbol{b}^{(k)}\|_2$, where the residual vector $\boldsymbol{r}^{l+1}$ is defined by (39). The CPU times are given in CPU milliseconds on a PC with 3.8 GHz Intel Xeon processor.

Based on Table 1 the discretization is second-order accurate when the number of time and space steps are increased at the same rate. On the finer grids the number of iterations per one time step is two. Thus, in this case the iteration (38) is a predictor-corrector time integration scheme. The errors and the CPU times are a few orders of magnitude smaller than in [2]. Thus, there is a clear advantage to use nonuniform grids and the fast recursion formulas for the integration rather than uniform grids for $\log x$ and FFT for evaluating the integrals as has been done in [2].

Table 2: The pricing errors, the ratio of l_2 errors, the total number of iterations, and the CPU time in milliseconds for the European call options with $\sigma = 0.15$.

m	n	error at 90	error at 100	error at 110	rate	iter.	time
6	40	4.4×10^{-3}	-1.6×10^{-2}	-6.1×10^{-3}		19	0.1
10	80	6.0×10^{-4}	-3.2×10^{-3}	-1.8×10^{-3}	4.8	26	0.4
18	160	6.2×10^{-4}	-6.9×10^{-4}	-1.7×10^{-4}	4.0	43	1.0
34	320	-1.4×10^{-4}	-1.5×10^{-4}	-2.1×10^{-4}	3.2	68	3.0
66	640	-7.2×10^{-6}	-3.6×10^{-5}	-3.7×10^{-5}	5.6	132	10.3

5.2 European call option

In the second example, we price European call options when the model parameters are given by

$$\begin{aligned} &\sigma = 0.15, \quad r = 0.05, \quad T = 0.25, \quad K = 100, \\ &\lambda = 0.1, \quad \alpha_1 = 3.0465, \quad \alpha_2 = 3.0775, \quad \text{and} \quad p = 0.3445, \end{aligned} \tag{51}$$

which are also used by d'Halluin, Forsyth, and Vetzal in [15]. Instead of pricing a call option directly we compute the price of a put option with the same strike price K and then we use the put-call parity for European options given by

$$v_c(t,x) = v_p(t,x) + x - Ke^{-r(T-t)}, \tag{52}$$

where v_c is the value of the call option and v_p is the value of the put option. The term $Ke^{-r(T-t)}$ is the value of the strike price K at the time t discounted from the expiry time T. The truncation boundary is at $X = 400$. The grid generation parameters in (10), (11), and (12) are $\alpha = 4$ and $\gamma = 0.45$.

Using a fine time-space grid we computed the reference prices for corresponding put options which are given in Table 3 along with the reference prices for other models and American options. After using the parity (52) we obtain call option prices that coincide with ones given in [15].

Table 2 reports the errors at 90, 100, and 110 as well as the ratios of consecutive discretization errors in the l_2-norm. Also, we give the number of iterations and CPU time in milliseconds. We have used the same stopping criterion and PC as in Section 5.1. Based on Table 2 again the discretization is second-order accurate. While the FFT based approach in [15] cannot maintain second-order convergence due to the jump in the density.

5.3 European put option with local volatility function

Here we price European put options with the same parameters as in (51) except that the local volatility is given by the function

$$\sigma(t,x) = 0.15 + 0.15(0.5 + 2t)\frac{(x/100 - 1.2)^2}{(x/100)^2 + 1.44}. \tag{53}$$

Table 3: The reference prices for put options.

type and volatility	m	n	v at 90	v at 100	v at 110
European, $\sigma = 0.15$ in [15]			9.430457	2.731259	0.552363
European, $\sigma = 0.15$	2050	20480	9.430457	2.731259	0.552363
European, σ in (53)	2050	20480	9.456392	2.767587	0.561144
American, $\sigma = 0.15$	45945	40960	10.005071	2.807879	0.561876
American, σ in (53)	45945	40960	10.010192	2.843178	0.570667

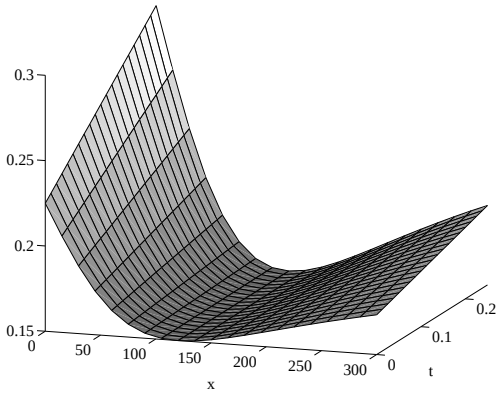

Figure 2: The local volatility function $\sigma(t, x)$.

The volatility surface is shown in Figure 2. We use the same grids as in Section 5.2 and the same stopping criterion for (38) as in Section 5.1. Table 4 reports the same values as Table 2 but with the local volatility function. The errors are computed using the reference prices in Table 3. The error behaves the same way as with the call option together with the constant volatility $\sigma = 0.15$. Furthermore, the number of iterations and the CPU times are similar in both cases. There is very slight increases in CPU times due to some additional computations caused by the use of the local volatility function.

Table 4: The pricing errors, the ratio of l_2 errors, the total number of iterations, and the CPU time in milliseconds for the European put options with the local volatility function.

m	n	error at 90	error at 100	error at 110	rate	iter.	time
6	40	7.2×10^{-3}	-2.4×10^{-2}	-6.7×10^{-3}		19	0.2
10	80	1.0×10^{-3}	-4.2×10^{-3}	-1.8×10^{-3}	5.4	27	0.4
18	160	7.0×10^{-4}	-8.8×10^{-4}	-1.7×10^{-4}	4.2	43	1.2
34	320	-1.2×10^{-4}	-2.0×10^{-4}	-2.1×10^{-4}	3.6	68	3.8
66	640	-3.8×10^{-6}	-4.7×10^{-5}	-3.8×10^{-5}	5.2	132	13.9

Table 5: The pricing errors, the ratio of l_2 errors, the total number of iterations, and the CPU time in milliseconds for the American put options with $\sigma = 0.15$.

method	m	n	error at 90	error at 100	error at 110	rate	iter.	time
split	10	40	-5.1×10^{-3}	-7.7×10^{-2}	-1.1×10^{-2}		31	0.2
penalty	10	40	-5.1×10^{-3}	-7.8×10^{-2}	-1.1×10^{-2}		33	0.2
split	21	80	-5.1×10^{-3}	-1.9×10^{-2}	-1.0×10^{-3}	4.0	62	0.6
penalty	21	80	-4.7×10^{-3}	-2.0×10^{-2}	-1.1×10^{-3}	3.9	61	0.8
split	47	160	4.1×10^{-3}	-4.7×10^{-3}	2.2×10^{-4}	3.2	114	2.2
penalty	47	160	3.0×10^{-3}	-5.1×10^{-3}	1.8×10^{-4}	3.5	121	2.9
split	109	320	9.6×10^{-4}	-1.2×10^{-3}	-8.0×10^{-5}	4.1	218	8.6
penalty	109	320	3.5×10^{-4}	-1.3×10^{-3}	-1.0×10^{-4}	4.3	231	11.0
split	256	640	3.6×10^{-4}	-2.8×10^{-4}	8.7×10^{-6}	3.3	512	38.7
penalty	256	640	1.4×10^{-4}	-3.5×10^{-4}	-4.1×10^{-7}	3.7	513	48.3

5.4 American put option

In the following example, we price American put options defined by the model parameters in (51). We use grids which are slightly less refined in the space with $\gamma = 0.4$ in (12) while the time refinement defined by $\alpha = 4$ in (10) and (11) is the same as before. The approximations of the prices of American options are less accurate due to the early exercise constraint. In order to improve the accuracy we multiply the number of time steps m roughly by $2^{1.25} \approx 2.38$ when the number of spatial steps n is doubled. An alternative approach would be to use a selector for the size of the time steps as has been done in [14] and [18], for example.

Here we use a more strict stopping criterion for the iterations (38) and (50) given by $\|\boldsymbol{r}^{l+1}\|_2 < 10^{-11}\|\boldsymbol{b}^{(k)}\|_2$, where $\boldsymbol{r}^{l+1}$ is the residual vector. With the penalty method we use the penalty parameter $\epsilon = 10^{-4}$.

Table 5 reports the results for the operator splitting method and the penalty method. The accuracies of both methods are fairly similar. Neither of them can quite maintain the second-order accuracy. Also the convergences of the iterations (38) for the operator splitting method and (50) for the penalty method are similar. Due to this the CPU times are also similar with the penalty method being slightly slower due to some additional updates caused by the changing coefficient matrix.

5.5 American put option with local volatility function

Here we price the same American put options as in the above section when the constant volatility is replaced by the local volatility function (53). The parameters in the methods are the same as before. The results reported in Table 6 are very similar to the ones obtained with $\sigma = 0.15$. There is a slight increase in the CPU times due to some additional computations caused by the use of the local volatility.

Table 6: The pricing errors, the ratio of l_2 errors, the total number of iterations, and the CPU time in milliseconds for the American put options with the local volatility function.

method	m	n	error at 90	error at 100	error at 110	rate	iter.	time
split	10	40	-1.0×10^{-2}	-8.4×10^{-2}	-1.1×10^{-2}		30	0.2
penalty	10	40	-1.0×10^{-2}	-7.7×10^{-2}	-1.1×10^{-2}		33	0.3
split	21	80	-4.3×10^{-3}	-2.1×10^{-2}	-1.2×10^{-3}	4.0	63	0.8
penalty	21	80	-1.8×10^{-3}	-2.0×10^{-2}	-1.1×10^{-3}	4.0	61	0.9
split	47	160	4.0×10^{-3}	-5.2×10^{-3}	1.2×10^{-4}	3.2	115	2.9
penalty	47	160	2.7×10^{-3}	-5.1×10^{-3}	1.8×10^{-4}	3.4	120	3.5
split	109	320	9.9×10^{-4}	-1.3×10^{-3}	-1.3×10^{-4}	4.0	218	11.7
penalty	109	320	3.4×10^{-5}	-1.3×10^{-3}	-1.0×10^{-4}	4.3	226	13.9
split	256	640	2.9×10^{-4}	-3.4×10^{-4}	-8.8×10^{-5}	3.8	512	53.3
penalty	256	640	1.2×10^{-4}	-3.4×10^{-4}	-4.4×10^{-8}	3.6	513	62.8

6 Conclusions

We described methods for the numerical valuation of European and American options under Kou's jump-diffusion model [27]. Following Andersen and Andreasen in [4] the volatility can be also a deterministic function. We considered only put options but the same methods can be used for call options and other options with more general payoff functions. The use of finite difference discretizations on refined grids lead to accurate prices. We described recursion formulas for the integral term which lead to very efficient numerical methods requiring $\mathcal{O}(n)$ operations per time step whereas all earlier approaches have required at least $\mathcal{O}(n \log n)$ operations. Here n denotes the number of spatial grid points. Furthermore, the recursion formulas are easy to implement also for nonuniform grids and they have second-order spatial accuracy. In the same way it is possible derive formulas with higher order accuracy.

For the dense linear systems resulting from pricing European options we used the stationary iterative method described Tavella and Randall in [37] which have been used also by Almendral and Oosterlee in [2], and d'Halluin, Forsyth, and Vetzal in [15]. Based on a theoretical result in [15] and numerical experiments we conclude that this iteration converges very rapidly requiring only a few iterations per time step. For American options we proposed two approaches: the operator splitting method described in [22], and the penalty method described in [14] by d'Halluin, Forsyth, and Labahn. In the numerical experiments the accuracy and computational effort of these both methods for American options were fairly similar. The experiments also demonstrated that European and American option prices with error around 0.1% can be computed in a few milliseconds on a PC. Thus, the developed methods are very efficient.

Acknowledgments

The author is grateful to Dr. Samuli Ikonen for many fruitful discussions on numerical methods for option pricing.

References

[1] Y. ACHDOU AND O. PIRONNEAU, *Computational methods for option pricing*, vol. 30 of Frontiers in Applied Mathematics, SIAM, Philadelphia, PA, 2005.

[2] A. ALMENDRAL AND C. W. OOSTERLEE, *Numerical valuation of options with jumps in the underlying*, Appl. Numer. Math., 53 (2005), pp. 1–18.

[3] ——, *Accurate valuation of European and American under the CGMY process*, SIAM J. Sci. Comput., (2006). To appear.

[4] L. ANDERSEN AND J. ANDREASEN, *Jump-diffusion processes: Volatility smile fitting and numerical methods for option pricing*, Rev. Deriv. Res., 4 (2000), pp. 231–262.

[5] H. T. BANKS, V. A. BOKIL, D. CIORANESCU, N. L. GIBSON, G. GRISO, AND B. MIARA, *Homogenization of periodically varying coefficients in electromagnetic materials*, J. Sci. Comput., 28 (2006), pp. 191–221.

[6] D. S. BATES, *Jumps and stochastic volatility: Exchange rate processes implicit Deutsche mark options*, Review Financial Stud., 9 (1996), pp. 69–107.

[7] F. BLACK AND M. SCHOLES, *The pricing of options and corporate liabilities*, J. Political Economy, 81 (1973), pp. 637–654.

[8] M. BRIANI, C. LA CHIOMA, AND R. NATALINI, *Convergence of numerical schemes for viscosity solutions to integro-differential degenerate parabolic problems arising in financial theory*, Numer. Math., 98 (2004), pp. 607–646.

[9] P. CARR, H. GEMAN, D. B. MADAN, AND M. YOR, *The fine structure of asset returns: an empirical investigation*, J. Business, 75 (2002), pp. 305–332.

[10] F. H. CLARKE, *Optimization and nonsmooth analysis*, John Wiley & Sons Inc., New York, 1983.

[11] N. CLARKE AND K. PARROTT, *Multigrid for American option pricing with stochastic volatility*, Appl. Math. Finance, 6 (1999), pp. 177–195.

[12] R. CONT AND P. TANKOV, *Financial modelling with jump processes*, Chapman & Hall/CRC, Boca Raton, FL, 2004.

[13] R. CONT AND E. VOLTCHKOVA, *A finite difference scheme for option pricing in jump diffusion and exponential Lévy models*, SIAM Numer. Anal., 43 (2005), pp. 1596–1626.

[14] Y. D'HALLUIN, P. A. FORSYTH, AND G. LABAHN, *A penalty method for American options with jump diffusion processes*, Numer. Math., 97 (2004), pp. 321–352.

[15] Y. D'HALLUIN, P. A. FORSYTH, AND K. R. VETZAL, *Robust numerical methods for contingent claims under jump diffusion processes*, IMA J. Numer. Anal., 25 (2005), pp. 87–112.

[16] D. DUFFIE, J. PAN, AND K. SINGLETON, *Transform analysis and asset pricing for affine jump-diffusions*, Econometrica, 68 (2000), pp. 1343–1376.

[17] B. DUPIRE, *Pricing with a smile*, Risk, 7 (1994), pp. 18–20.

[18] P. A. FORSYTH AND K. R. VETZAL, *Quadratic convergence for valuing American options using a penalty method*, SIAM J. Sci. Comput., 23 (2002), pp. 2095–2122.

[19] M. B. GILES AND R. CARTER, *Convergence analysis of Crank-Nicolson and Rannacher time-marching*, J. Comput. Finance, 9 (2006), pp. 89–112.

[20] S. HESTON, *A closed-form solution for options with stochastic volatility with applications to bond and currency options*, Rev. Financial Stud., 6 (1993), pp. 327–343.

[21] M. HINTERMÜLLER, K. ITO, AND K. KUNISCH, *The primal-dual active set strategy as a semismooth Newton method*, SIAM J. Optim., 13 (2003), pp. 865–888.

[22] S. IKONEN AND J. TOIVANEN, *Operator splitting methods for American option pricing*, Appl. Math. Lett., 17 (2004), pp. 809–814.

[23] ——, *Operator splitting methods for pricing American options with stochastic volatility*, Tech. Rep. B11/2004, Department of Mathematical Information Technology, University of Jyväskylä, Jyväskylä, Finland, 2004. Submitted.

[24] ——, *Efficient numerical methods for pricing American options under stochastic volatility*, Tech. Rep. B12/2005, Department of Mathematical Information Technology, University of Jyväskylä, Jyväskylä, Finland, 2005. Submitted.

[25] K. ITO AND K. KUNISCH, *Parabolic variational inequalities: The Lagrange multiplier approach*, J. Math. Pures Appl., 85 (2006), pp. 415–449.

[26] K. ITO AND J. TOIVANEN, *Lagrange multiplier approach with optimized finite difference stencils for pricing American options*, Tech. Rep. B6/2006, Department of Mathematical Information Technology, University of Jyväskylä, Jyväskylä, Finland, 2006. Submitted.

[27] S. G. KOU, *A jump-diffusion model for option pricing*, Management Sci., 48 (2002), pp. 1086–1101.

[28] S. G. KOU AND H. WANG, *Option pricing under a double exponential jump diffusion model*, Management Sci., 50 (2004), pp. 1178–1198.

[29] T. A. MANTEUFFEL AND A. B. WHITE, JR., *The numerical solution of second-order boundary value problems on nonuniform meshes*, Math. Comp., 47 (1986), pp. 511–535.

[30] A.-M. MATACHE, P.-A. NITSCHE, AND C. SCHWAB, *Wavelet Galerkin pricing of American options on Lévy driven assets*, Quant. Finance, 5 (2005), pp. 403–424.

[31] A.-M. MATACHE, C. SCHWAB, AND T. P. WIHLER, *Fast numerical solution of parabolic integrodifferential equations with applications in finance*, SIAM J. Sci. Comput., 27 (2005), pp. 369–393.

[32] A.-M. MATACHE, T. VON PETERSDORFF, AND C. SCHWAB, *Fast deterministic pricing of options on Lévy driven assets*, M2AN Math. Model. Numer. Anal., 38 (2004), pp. 37–71.

[33] R. C. MERTON, *Theory of rational option pricing*, Bell J. Econom. Management Sci., 4 (1973), pp. 141–183.

[34] ——, *Option pricing when underlying stock returns are discontinuous*, J. Financial Econ., 3 (1976), pp. 125–144.

[35] D. M. POOLEY, K. VETZAL, AND P. A. FORSYTH, *Remedies for non-smooth payoffs in option pricing*, J. Comput. Finance, 6 (2003), pp. 25–40.

[36] R. RANNACHER, *Finite element solution of diffusion problems with irregular data*, Numer. Math., 43 (1984), pp. 309–327.

[37] D. TAVELLA AND C. RANDALL, *Pricing financial instruments: The finite difference method*, John Wiley & Sons, Chichester, 2000.

[38] I. R. WANG, J. W. L. WAN, AND P. A. FORSYTH, *Robust numerical valuation of European and American options under the CGMY process*, tech. rep., David R. Cheriton School of Computer Science, University of Waterloo, Waterloo ON, Canada, 2006.

[39] X. L. ZHANG, *Numerical analysis of American option pricing in a jump-diffusion model*, Math. Oper. Res., 22 (1997), pp. 668–690.