Option pricing in jump diffusion models with quadratic spline collocation

Christina C. Christara and Nat Chun-Ho Leung
Department of Computer Science
University of Toronto
Toronto, Ontario M5S 3G4, Canada
{ccc,natleung}@cs.utoronto.ca

Abstract

In this paper, we develop a robust numerical method in pricing options, when the underlying asset follows a jump diffusion model. We demonstrate that, with the quadratic spline collocation method, the integral approximation in the pricing PIDE is intuitively simple, and comes down to the evaluation of the probabilistic moments of the jump density. When combined with a Picard iteration scheme, the pricing problem can be solved efficiently. We present the method and the numerical results from pricing European and American options with Merton's and Kou's models.

Key words: quadratic spline, collocation, finite difference, European option, American option, partial integro-differential equation, Merton model, Kou model, calculation of Greeks

1 Introduction

It is well-known that the simple Black-Scholes model cannot explain the implied volatility smile seen in the calibration of the volatility parameter to market prices. Large jumps in asset returns have also been observed, and these discontinuities are arguably not reflected in a pure diffusion model.

Several models have been developed as alternatives to or extensions of the Black-Scholes model. The local volatility approach, proposed in [11], generalizes the Black-Scholes model by considering a deterministic time-dependent and price-dependent volatility function. This model is theoretically attractive, as the local volatility function can be computed when a smooth set of option prices in all strikes and maturities is known. In reality, a numerical optimization method (see for example [8]) is required, as there is only a restricted set of options that trade enough, so that the quoted prices are accurate enough for these methods to work. As was pointed out in [2], the local volatility approach alone does not produce stationary implied volatilities, in contrast with market experience. Stochastic volatility models, such as those in [14] or [15], have been proposed. As multi-factor models, these models are numerically more expensive.

Another alternative to the classical Black-Scholes model is a jump diffusion model, due to [18]. The idea is to model asset returns by the usual Wiener process combined with a compound Poisson process, the latter corresponding to the "jumps". By varying the jump parameters, one can obtain volatility shapes, and control the skewness and kurtosis of the log asset return. A number of studies (for example, [3] and [4]) suggest the addition of jumps to stochastic volatility models.

When the asset follows a jump diffusion model where the jump density is log-normally distributed ([18]), closed-form formulas can be obtained for European options' prices. For other jump size distributions, closed-form solutions may not be guaranteed. Pricing exotic options in general requires numerical methods as well.

The option prices in a jump model satisfy a partial integro-differential equation (PIDE). The integral is performed over the entire grid, and this non-locality poses numerical challenges. Different numerical treatments have been proposed. Explicit methods or explicit treatment of the integral, such as those proposed in [1] or [19], suffer from severe stability and convergence constraints. An FFT-ADI (Fast Fourier Transform - Alternating Direction Implicit) approach is developed in [2], which is unconditionally stable and has second order convergence. But as noted in both [10] and [5], it is not clear how this method can be extended to pricing American options.

An explicit FFT treatment of the integral term together with a fixed-point iteration is suggested in [10], and has second order convergence. The method, while robust and unconditionally stable, suffers from a common problem in Fourier-based methods. The effect of wrap-around errors resulting from non-periodicity of option values can be large. The remedy for this, suggested in [10], is to enlarge the grid to the point that the wrap-around errors will be non-material. Enlarging the grid can be expensive especially in higher-dimensions. For Kou's model of the jump density, a much larger grid than that of Merton's is necessary to avoid large wrap-around errors. It was also reported in [10] that convergence can be slow for Kou's model, when numerical FFT methods are used.

The Fourier family methods also include the Fourier space timestepping method, developed in [16]. These Fourier methods are effective and robust, yet they usually require special attention to the grids. A uniform grid is usually required for FFTs, sometimes leading to a waste of computational resources in regions that are not important. In addition, a log transform is needed to utilize the FFT methods. That means one has to maintain two grids, and interpolation between the two grids is needed to communicate information. When one does timestepping in the original S-space instead of the Fourier space (the latter suggested in [16]), up to two interpolations can be required in each timestep. Furthermore, it is not easy to generalize the Fourier methods for the pricing of various exotic options.

There are other numerical approaches in the literature. Clever change of variable techniques have been proposed in [5], which computes the integral in Merton's model by solving the heat equation along an artificial dimension, and for Kou's model enables fast valuation of the integral term in linear time. While these are attractive, the transformations are specific to the aforementioned two models (and their related families).

In this paper, we develop a quadratic spline collocation method in the pricing problem under a finite activity jump diffusion model. The evaluation of the integral term is reduced to the computation of the probabilistic moments of the jump density, and works best for jump densities that have analytically tractable partial moments. For a fixed non-adaptive (uniform or non-uniform) grid, our method requires only a pre-computation of certain integral matrices. To the authors' knowledge, no prior work has been done on spline collocation methods in jump diffusion models.

2 PDE/PIDE formulation

2.1 Model

Let $S(t)$ be the price of an asset. In the jump diffusion framework, $S(t)$ is assumed to follow the dynamics

$$\frac{dS}{S} = \mu dt + \sigma dW(t) + d(\sum_{i=1}^{N(t)} (J_i - 1)), \tag{1}$$

where μ and σ are drift and volatility parameters respectively, W_t is a Wiener process, $N(t)$ is Poisson with rate λ, and J_i is the size of the i-th jump, for $i = 1, \ldots, N(t)$. We assume that the J_i's are non-negative i.i.d. with some distribution g.

It is clear that some modelling assumptions on g are needed. In this paper, we focus primarily on two models that are commonly considered studied in the literature, namely Merton's and Kou's models, with the density g defined, respectively, by

$$\text{(Merton)} \quad g(x) = \frac{\exp(-\frac{(\log(x)^2-\mu)}{2\gamma^2})}{\sqrt{2\pi}\gamma x} \tag{2}$$

where $\gamma > 0, \mu \in \mathcal{R}$, and

$$\text{(Kou)} \quad g(x) = \begin{cases} p\eta_1 \frac{\exp(-\eta_1 \log(x))}{x} & \text{for } x \geq 1, \\ (1-p)\eta_2 \frac{\exp(\eta_2 \log(x))}{x} & \text{for } x < 1. \end{cases} \tag{3}$$

where $\eta_1 > 1$, $\eta_2 > 0$, and $0 < p < 1$. The specification (2), originally considered in [18], is the first of its kind, and models a jump size density by a lognormal distribution (equivalently, it models $\log(J)$ by a normal distribution). On the other hand, the assumption (3), considered in [17], models $\log(J)$ by a double exponential distribution.

We remark that calibration of the jump parameters is out of the scope of this study.

Let $V(S,\tau)$ be the value of a contingent claim that depends on S and backward time $\tau = T - t$, where T is the expiration time of the contract. Based on the stochastic differential equation (SDE) (1), it can be shown (see [18]) that V satisfies the PIDE

$$\frac{\partial V}{\partial \tau} = \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + (r - \lambda\kappa) S \frac{\partial V}{\partial S} - (r+\lambda)V + \lambda \int_0^\infty V(S\eta) g(\eta) d\eta \tag{4}$$

with $V(\tau = 0) = F(S(\tau = 0))$ being the terminal payoff, and $\kappa = \mathcal{E}(J-1)$, where $\mathcal{E}$ is the expectation operator over the random variable J.

For simplicity, we take r (the risk-free interest rate) and σ to be constant for the rest of the paper. However, r and σ can be non-constant, e.g. $r = r(t)$ for a projected deterministic risk-free curve or $\sigma = \sigma(S,t)$ for a local volatility model. If r and σ are variable, provided they are deterministic, the arguments in this paper will go through with only trivial modifications.

2.2 Time discretization: Crank-Nicolson and Rannacher

A common numerical timestepping approach for PDEs and PIDEs is the θ-method. For notational convenience, denote

$$\mathcal{L}V \equiv \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + (r - \lambda\kappa) S \frac{\partial V}{\partial S} - (r+\lambda)V + \lambda \int_0^\infty V(S\eta) g(\eta) d\eta.$$

Then the time-dependent PIDE (4) can be written as

$$\frac{\partial V}{\partial \tau} - \mathcal{L}V = 0.$$

Discretizing along the time direction, at the j-th timestep, we have

$$\frac{V^{j+1} - V^j}{\Delta\tau} = \theta \mathcal{L}V^{j+1} + (1-\theta)\mathcal{L}V^j. \tag{5}$$

When $\theta = 0.5$, the scheme is known as Crank-Nicolson (CN), and formally has second order local truncation error in $\Delta\tau$, assuming V satisfies certain smoothness assumptions. The choice $\theta = 1$ is known as fully implicit timestepping. This scheme has favourable stability properties, but it is only first order consistent in $\Delta\tau$. In this work, unless otherwise specified, we use the CN-Rannacher timestepping technique, which uses CN after a few fully implicit timesteppings initially. It should be noted that the CN timestepping alone might not lead to second order convergence, when applied to the Black-Scholes equation (even without the presence of the integral term), due to the discontinuity of the derivative in the initial conditions. However, as shown in [13], the CN-Rannacher timestepping technique results in second order convergence; see also [12]. It is also worth noting, that the integrand of the integral term in (4) may also exhibit discontinuities, but the integral term itself does not.

2.3 Space discretization: quadratic spline collocation

In this work, we explore the use of quadratic spline collocation (QSC) in the space discretization of the PIDE problem (4). This method approximates the solution of a PDE or PIDE using quadratic splines. The method of quadratic spline collocation in parabolic PDEs has been studied in [7].
Define

$$\psi(x) \equiv \frac{1}{2} \begin{cases} x^2 & \text{if } 0 \le x \le 1, \\ -3 + 6x - 2x^2 & \text{if } 1 \le x \le 2, \\ 9 - 6x + x^2 & \text{if } 2 \le x \le 3, \\ 0 & \text{elsewhere.} \end{cases}$$

On a uniform mesh partition of a generic space interval $[a, b]$, with n subintervals and step size $h = \frac{b-a}{n}$, the space of quadratic splines is spanned by the functions $\phi_i(x) = \psi(\frac{x-a}{h} - i + 2)$, where $i = 0, \ldots, n+1$.

We then model the approximation to the solution $V(\tau, x)$ of the PIDE (4) by

$$V_\Delta(\tau, x) = \sum_{i=0}^{n+1} \mathbf{c}_i(\tau)\phi_i(x), \tag{6}$$

for some unknown "degrees of freedom" $\mathbf{c}_i(\tau)$, $i = 0, \ldots, n+1$. The zeroth, first and second derivatives of V_Δ with respect to space variable x can be easily computed given the functional form (6). Note that the second derivative of V_Δ is discontinuous at the nodes. For uniformity in notation, we set $\partial^2 V_\Delta(\cdot, x_0)/\partial x^2 = \partial^2 V_\Delta(\cdot, x_n)/\partial x^2 = 0$.

In the standard QSC method, one defines a set of prescribed collocation points m_i, $i = 0, \ldots, n+1$. If the mesh is given by $[x_{i-1}, x_i]$, for $i = 1, \ldots, n$, and is uniform, then a common choice is $m_i = \frac{x_{i-1}+x_i}{2}$, $i = 1, \ldots, n$, and $m_0 = x_0$, $m_{n+1} = x_n$, the latter two points used to incorporate the spatial boundary conditions. At the j-th timestep, given $\mathbf{c}_i^{j-1}, i = 0, \ldots, n+1$, (equivalently V_Δ^{j-1}), we find $\mathbf{c}_i^j, i = 0, \ldots, n+1$, such that (5) holds at the prescribed collocation points m_i for $i = 1, \ldots, n$, and we impose context-specific Dirichlet boundary conditions at m_0 and m_{n+1}. This results in a linear system with $n+2$ equations, and $n+2$ unknowns (the degrees of freedom).

The values of the quadratic spline V_Δ at the collocation points and the degrees of freedom are related by the *quadratic spline interpolation matrix* T_0. More specifically, if we let $\vec{V}_\Delta(\tau)$ be a vector whose i-th entry is $V_\Delta(\tau, m_{i-1})$, $i = 1, \ldots, n+2$, then

$$\vec{V}_\Delta(\tau) = T_0 \vec{\mathbf{c}}(\tau).$$

Similarly, for the first and second derivatives of V_Δ with respect to x, if we let $\vec{V}_\Delta^{(i)}(\tau)$, $i = 1, 2$, be vectors whose i-th entry, $i = 1, \ldots, n+2$, is $\partial V_\Delta(\tau, m_{i-1})/\partial x$ and $\partial^2 V_\Delta(\tau, m_{i-1})/\partial x^2$, respectively, then

$$\vec{V}_\Delta^{(i)}(\tau) = T_i \vec{\mathbf{c}}(\tau),$$

for some precomputed matrices T_i. We emphasize that T_i, $i = 0, 1, 2$, are tridiagonal matrices. (The details of the T_i matrices are given in Appendix A.)

The matrix arising from the QSC discretization of the spatial derivative part of (4) is given by a matrix T_L, of which the i-th row, $i = 2, \ldots, n+1$, is given by

$$T_L(i,:) = \frac{1}{2}\sigma^2 m_{i-1}^2 T_2(i,:) + (r - \lambda\kappa) m_{i-1} T_1(i,:) - (r+\lambda) T_0(i,:). \tag{7}$$

The first and last rows of T_L are not defined in (7) as they are not needed when forming the *final collocation matrix* that needs to be solved at each timestep. As will be seen, the i-th row, $i = 2, \ldots, n+1$, of the final collocation matrix to be solved at each timestep is

$$T_0(i,:) - \theta(\Delta\tau) T_L(i,:),$$

while the first and last rows are set to incorporate the Dirichlet boundary conditions, thus they are equal to the first and last rows of T_0, respectively. The detailed formulas can be found in [7].

In the case where $\lambda = 0$ (i.e. no jumps), the framework is already discussed in [6]. The present work is to study the collocation approach when the integral term is introduced.

2.4 Treatment of the integral

Given the discretization

$$V_\Delta(\tau, x) = \sum_{i=0}^{n+1} \mathbf{c}_i(\tau)\phi_i(x),$$

one wants to evaluate the quantity

$$\int_0^\infty V_\Delta(\tau, S\eta) g(\eta) d\eta$$

at $S = m_i$, where $i = 1, \ldots, n$ (interior collocation points), so that collocation of the PIDE (4) can be done. To evaluate the integral term, we consider the integral inside and outside the spatial grid separately.

2.4.1 Integral inside the computational domain

To evaluate the integral term inside the generic computational domain $[a, b]$, observe that, in $[a, b]$, the approximation V_Δ is a piecewise polynomial at any τ. On each of the subintervals $[x_{i-1}, x_i]$, we have that $V_\Delta(\tau, x) = a_i x^2 + b_i x + c_i$, for some a_i, b_i, c_i. While a_i, b_i, c_i depend on τ, for simplicity of notation, we omit this dependence. Therefore,

$$x_{i-1} \le S\eta \le x_i \Leftrightarrow \frac{x_{i-1}}{S} \le \eta \le \frac{x_i}{S}.$$

Hence on the interval $[\frac{x_{i-1}}{S}, \frac{x_i}{S}]$, we can write $V_\Delta(\tau, S\eta) = a_i(S\eta)^2 + b_i(S\eta) + c_i$.

It is then clear that the integral is locally computed through partial moments of the jump size density

$$\int_u^v \eta^j g(\eta) d\eta,$$

for $j = 0, 1, 2$, with $u = \frac{x_{i-1}}{S}$ and $v = \frac{x_i}{S}$. If the jump size distribution has sufficient analytical tractability, as in models (2) and (3), then the partial moments can be evaluated *analytically*.

As an example, one can evaluate this quantity for model (2) as follows:

$$\begin{aligned}
\int_u^v \eta^j g(\eta) d\eta &= \int_u^v \eta^j \frac{\exp(-\frac{(\log(\eta)-\mu)^2}{2\gamma^2})}{\sqrt{2\pi}\gamma\eta} d\eta \\
&= \frac{1}{\sqrt{2\pi}\gamma} \int_{\log(u)}^{\log(v)} \exp(jy) \exp(-\frac{(y-\mu)^2}{2\gamma^2}) dy \quad (\text{where } y = \log\eta) \\
&= \frac{1}{\sqrt{2\pi}\gamma} \int_{\log(u)}^{\log(v)} \exp(-\frac{(y-\mu)^2 - 2\gamma^2 jy}{2\gamma^2}) dy \\
&= \frac{1}{\sqrt{2\pi}\gamma} \int_{\log(u)}^{\log(v)} \exp(-\frac{(y-\alpha_j y+\mu^2)}{2\gamma^2}) dy \quad (\text{where } \alpha_j = 2\mu + 2\gamma^2 j) \\
&= \frac{1}{\sqrt{2\pi}\gamma} \int_{\log(u)}^{\log(v)} \exp(-\frac{(y-\frac{\alpha_j}{2})^2 + \mu^2 - \frac{\alpha_j^2}{4}}{2\gamma^2}) dy \\
&= \exp(-\frac{\mu^2 - \frac{\alpha_j^2}{4}}{2\gamma^2}) [\mathcal{N}(\log(v), \frac{\alpha_j}{2}, \gamma) - \mathcal{N}(\log(u), \frac{\alpha_j}{2}, \gamma)].
\end{aligned}$$

The integral inside the computational domain can be evaluated as

$$\begin{aligned}
\int_0^{\frac{x_n}{S}} V_\Delta(\tau, S\eta) g(\eta) d\eta &= \sum_{i=1}^n \int_{\frac{x_{i-1}}{S}}^{\frac{x_i}{S}} (a_i(S\eta)^2 + b_i(S\eta) + c_i) g(\eta) d\eta \\
&= \sum_{i=1}^n [a_i \exp(-\frac{\mu^2 - \frac{\alpha_2^2}{4}}{2\gamma^2}) (\mathcal{N}(\log(\frac{x_i}{S}), \frac{\alpha_2}{2}, \gamma) - \mathcal{N}(\log(\frac{x_{i-1}}{S}), \frac{\alpha_2}{2}, \gamma)) \\
&\quad + b_i \exp(-\frac{\mu^2 - \frac{\alpha_1^2}{4}}{2\gamma^2}) (\mathcal{N}(\log(\frac{x_i}{S}), \frac{\alpha_1}{2}, \gamma) - \mathcal{N}(\log(\frac{x_{i-1}}{S}), \frac{\alpha_1}{2}, \gamma)) \\
&\quad + c_i (\mathcal{N}(\log(\frac{x_i}{S}), \mu, \gamma) - \mathcal{N}(\log(\frac{x_{i-1}}{S}), \mu, \gamma))] \\
&= \sum_{i=1}^n [a_i u_i + b_i v_i + c_i w_i],
\end{aligned}$$

for some coefficients u_i, v_i, w_i, that depend only on the jump density and the grid. Once a grid is determined, these can be evaluated prior to the main computation. The coefficients a_i, b_i, c_i are linear in the degrees of freedom of a quadratic spline, so the integral can be evaluated under an affine transformation. Denote the linear part by a $n \times (n+2)$ matrix $\hat{R}$, so that

$$\left[\int_0^{\frac{x_n}{S}} V_\Delta(\tau, S\eta) g(\eta) d\eta \right]_{S=m_1}^{m_n} = \hat{R}\vec{\mathbf{c}},$$

where V_Δ admits the basis representation given by $\vec{\mathbf{c}}$. For the details of the $\hat{R}$ definition, see Appendix A.

The computation above can also be applied to Kou's density model, and indeed any jump size model that has analytical partial moments. In the case that these cannot be evaluated analytically, one can numerically approximate the partial moments to sufficient accuracy in the pre-computation stage.

We now extend the $\hat{R}$ matrix to a $(n+2) \times (n+2)$ matrix R by adding identically zero first and last rows. The integral matrix R has properties resulting from its origin as a probabilistic expectation. To describe the properties, we adopt the notation $\vec{\mathbf{c}} \geq 0$ to denote $\mathbf{c}_i \geq 0, \forall i$.

PROPOSITION 1. *The matrix R satisfies $R\vec{\mathbf{c}} \geq 0$ for all $\vec{\mathbf{c}} \geq 0$, and $||R||_\infty \leq 1$.*

Proof. For any vector of degrees of freedom $\vec{\mathbf{c}}$, denote by $V_{\mathbf{c}}$ the quadratic spline given by $\vec{\mathbf{c}}$. If $\vec{\mathbf{c}} \geq 0$, then, from the non-negativity of the basis functions, we have $V_{\mathbf{c}} \geq 0$ in the domain that it is defined, and hence

$$\int_0^{\frac{x_n}{m_i}} V_{\mathbf{c}}(\tau, m_i\eta)g(\eta)d\eta \geq 0.$$

For the ∞-norm bound of R, we first show the inequality $||V_{\mathbf{c}}||_\infty \leq ||\vec{\mathbf{c}}||_\infty$, for any vector $\vec{\mathbf{c}}$ of degrees of freedom. A proof of this inequality is as follows. Without loss of generality, let $x \in [0, 1]$ and we have

$$\begin{aligned}
||\mathbf{c}_0\phi_0(x) + \mathbf{c}_1\phi_1(x) + \mathbf{c}_2\phi_2(x)||_\infty &\leq |\mathbf{c}_0|\phi_0(x) + |\mathbf{c}_1|\phi_1(x) + |\mathbf{c}_2|\phi_2(x) \\
&\leq \frac{1}{2}(|\mathbf{c}_0|(1-x)^2 + |\mathbf{c}_1|(-2x^2+2x+1) + |\mathbf{c}_2|x^2) \\
&\leq \frac{\max\{|\mathbf{c}_0|, |\mathbf{c}_1|, |\mathbf{c}_2|\}}{2}((1-x)^2 + (-2x^2+2x+1) + x^2) \\
&= \max\{|\mathbf{c}_0|, |\mathbf{c}_1|, |\mathbf{c}_2|\}.
\end{aligned}$$

Now, again for any vector $\vec{\mathbf{c}}$ of degrees of freedom, we have

$$\begin{aligned}
||R \cdot \vec{\mathbf{c}}||_\infty &= \max_{i=0,\ldots,n+1} |\int_0^{\frac{x_n}{m_i}} V_{\mathbf{c}}(m_i\eta)g(\eta)d\eta| \\
&\leq \max_{i=0,\ldots,n+1} \int_0^{\frac{x_n}{m_i}} |V_{\mathbf{c}}(m_i\eta)g(\eta)|d\eta \\
&\leq \max_{i=0,\ldots,n+1} \int_0^{\frac{x_n}{m_i}} ||\vec{\mathbf{c}}||_\infty g(\eta)d\eta \\
&= \max_{i=0,\ldots,n+1} ||\vec{\mathbf{c}}||_\infty \int_0^{\frac{x_n}{m_i}} g(\eta)d\eta \\
&\leq ||\vec{\mathbf{c}}||_\infty.
\end{aligned}$$

So we have $||R||_\infty \leq 1$. □

2.4.2 Boundary condition: an extended one

The above computation did not consider the part of the integral outside the grid: $\int_{\frac{x_n}{S}}^{\infty} V_\Delta(S\eta)g(\eta)d\eta$. The way to evaluate or approximate this quantity depends on the type of option being priced.
For a simple put option, if the grid is large enough, we may just take

$$V_\Delta(\tau, x) = 0 \text{ for } x > x_n.$$

For a simple call option with strike K with risk-free rate r, we can take

$$V_\Delta(\tau, x) = x - K\exp(-r\tau) \text{ for } x > x_n.$$

For an up-and out barrier option with barrier B, we can (obviously) take

$$V_\Delta(\tau, x) = 0 \text{ for } x > x_n = B.$$

Given these analytical approximations past the grid, the integral can be evaluated on $[x_n, \infty]$, using the same ideas as in Section 2.4.1. Thus, the vector $\mathbf{b}$ that arises from the extended boundary conditions, and which possibly depends on τ, has components

$$\mathbf{b}_1 = 0,\ \mathbf{b}_{i+1} = \int_{\frac{x_n}{m_i}}^{\infty} V(\tau, m_i\eta)g(\eta)d\eta,\ i = 1, \ldots, n,\ \mathbf{b}_{n+2} = 0.$$

Concerning the accuracy of this approximation of the semi-infinite integral, and taking into account that g is a density function, therefore, $\int_{-\infty}^{\infty} g(\eta)d\eta = 1$, we have that

$$\begin{aligned} |\int_{\frac{x_n}{S}}^{\infty} V_\Delta(\tau, S\eta)g(\eta)d\eta - \int_{\frac{x_n}{S}}^{\infty} V(\tau, S\eta)g(\eta)d\eta| &\leq \int_{\frac{x_n}{S}}^{\infty} |V_\Delta(\tau, S\eta) - V(\tau, S\eta)|g(\eta)d\eta \\ &\leq ||(V_\Delta(\tau, x) - V(\tau, x))\mathbf{1}_{x\geq x_n}||_\infty \int_{\frac{x_n}{S}}^{\infty} g(\eta)d\eta \\ &\leq ||(V_\Delta(\tau, x) - V(\tau, x))\mathbf{1}_{x\geq x_n}||_\infty, \end{aligned}$$

where $\mathbf{1}_A$ is the set indicator function of A, i.e. the function that has value 1 at all points of A and value 0 elsewhere.

The interpretation of the above bound is that the error of this approximation to the extended part of the integral is *at most* as bad as the error of the user-specified assumption, or belief, on the asymptotic behaviour of the option. The above argument depends on the (reasonable) assumption that the quantity

$$||(V_\Delta(\tau, x) - V(\tau, x))\mathbf{1}_{x\geq x_n}||_\infty$$

is bounded.

In practice, if we are concerned only about practical values of the option around the strike, then the error of this part of the integral can be made quickly small enough by extending the grid. Assume that $g(x) \leq C\exp(-\rho x)$ asymptotically for some ρ (which is true for both jump densities in (2) and (3)). If we extend the grid so that $\frac{x_n}{S}$ is large for S near the strike, then

$$\begin{aligned} ||(V_\Delta(\tau, x) - V(\tau, x))\mathbf{1}_{x\geq x_n}||_\infty \int_{\frac{x_n}{S}}^{\infty} g(\eta)d\eta &\leq ||(V_\Delta(\tau, x) - V(\tau, x))\mathbf{1}_{x\geq x_n}||_\infty \int_{\frac{x_n}{S}}^{\infty} C\exp(-\rho\eta)d\eta \\ &= \frac{C}{\rho}||(V_\Delta(\tau, x) - V(\tau, x))\mathbf{1}_{x\geq x_n}||_\infty \exp(-\rho\frac{x_n}{S}). \end{aligned}$$

The last two terms decrease (in fact rapidly for the exponential one) as x_n increases, the first being finite. The error decreases exponentially as we extend the grid. So this approximation of the extended part of the integral would not practically affect the accuracy solution, if we specify a reasonable extended boundary condition.

2.5 Dense system

The integral matrix R defined in Section 2.4.1 is a dense one, due to the global nature of the integral at each collocation point. The CN-Rannacher discretization in time, as described in Section 2.2, then involves solving a dense system at each timestep, which is inefficient. To avoid the direct solution of the dense linear system, a fixed-point iteration can be used, as proposed in [10]. We describe the scheme below in the context of QSC.

Recall that the θ-discretization scheme is, in general,

$$\frac{V^{j+1} - V^j}{\Delta\tau} = \theta\mathcal{L}V^{j+1} + (1-\theta)\mathcal{L}V^j,$$

and that, in the context of QSC, the unknowns are the degrees of freedom, $\vec{\mathbf{c}}$.

The θ-discretization of the PIDE, when combined with (7), can be written as

$$\begin{aligned} & (T_0 - \theta(\Delta\tau)T_L - \theta(\Delta\tau)\lambda R)\vec{\mathbf{c}}^{j+1} - \theta(\Delta\tau)\lambda\mathbf{b}^{j+1} \\ = \; & (T_0 + (1-\theta)(\Delta\tau)T_L + (1-\theta)(\Delta\tau)\lambda R)\vec{\mathbf{c}}^{j} + (1-\theta)(\Delta\tau)\lambda\mathbf{b}^{j}. \end{aligned} \tag{8}$$

The boundary terms, $\mathbf{b}^j$ and $\mathbf{b}^{j+1}$, are from our assumptions of the extended boundary behaviour, while the first and last equations of (8) are modified to be

$$T_0(1,:)\vec{\mathbf{c}}^{j+1} = d_1^{j+1}, \quad T_0(n+2,:)\vec{\mathbf{c}}^{j+1} = d_2^{j+1}, \tag{9}$$

where d_1^{j+1} and d_2^{j+1} arise from the corresponding Dirichlet conditions prevailing at time $(j+1)(\Delta\tau)$.

The presence of the dense matrix R in (8) makes the resulting system a dense one. The fixed-point iteration involves solving the system with the dense part "lagging" one step behind. For notational convenience, denote

$$\mathbf{u}^{j+1} = \theta(\Delta\tau)\lambda\mathbf{b}^{j+1} + (T_0 + (1-\theta)(\Delta\tau)T_L + (1-\theta)(\Delta\tau)\lambda R)\vec{\mathbf{c}}^{j} + (1-\theta)(\Delta\tau)\lambda\mathbf{b}^{j},$$

and

$$A = T_0 - \theta(\Delta\tau)T_L, \quad D = \theta(\Delta\tau)\lambda R,$$

where we recall that the first and last rows of T_L and D are zero. Modifying the first and last components of $\mathbf{u}^{j+1}$ to $\mathbf{u}_1^{j+1} = d_1^{j+1}$ and $\mathbf{u}_{n+2}^{j+1} = d_2^{j+1}$, the θ-discretization of the PIDE as given in (8), results in solving the linear system

$$(A - D)\vec{\mathbf{c}}^{j+1} = \mathbf{u}^{j+1} \tag{10}$$

which is dense, due to the presense of D. Denoting by $\vec{\mathbf{c}}_k^j$ the result of the k-th fixed-point iteration at the j-th timestep. the fixed-point iteration in the $(j+1)$-st timestep is as follows:

Set $\vec{\mathbf{c}}_0^{j+1} = \vec{\mathbf{c}}^j$

for $k = 1, \ldots,$ until convergence **do**

Solve

$$A\vec{\mathbf{c}}_k^{j+1} = D\vec{\mathbf{c}}_{k-1}^{j+1} + \mathbf{u}^{j+1}. \tag{11}$$

end for

Set $\vec{\mathbf{c}}^{j+1} = \vec{\mathbf{c}}_k^{j+1}$

Note that the matrix $A = T_0 - \theta(\Delta\tau)T_L$ is tridiagonal, and is easier to solve than the full system. Recall also that the first and last equations of the system to solve are given in (9).

To show that the fixed-point iteration scheme converges, we consider a mathematically equivalent version of the system (10), which simplifies the mathematical derivations. As explained in Appendix B, any $(n+2)\times(n+2)$ system arising from QSC with Dirichlet conditions can by transformed to an equivalent one of size $n \times n$ that has the Dirichlet conditions incorporated. Let

$$(A' - D')\vec{\mathbf{q}}^{j+1} = \mathbf{v}^{j+1} \tag{12}$$

be the system arising after the transformation of (10). The fixed-point iteration corresponding to the transformed system (12) requires the solution of

$$A'\vec{\mathbf{q}}_k^{j+1} = D'\vec{\mathbf{q}}_{k-1}^{j+1} + \mathbf{v}^{j+1}. \tag{13}$$

Denote $\vec{e}_k = \vec{\mathbf{q}}_k^{j+1} - \vec{\mathbf{q}}^{j+1}$, where $\vec{\mathbf{q}}^{j+1}$ is the true solution to (12). It is easy to see that

$$A'\vec{e}_k = D'\vec{e}_{k-1}.$$

It is shown in Appendix C, that, for sufficiently small spatial step size h, and if the left boundary point satisfies $x_0 > 0$, we have

$$||A'^{-1}||_\infty \leq C$$

for some C independent of the mesh size. We have also numerically studied $||A'^{-1}||_\infty$ and verified that it is bounded independently of n, even when $x_0 = 0$. Therefore, using the fact that $||D'||_\infty = O(\Delta\tau)||R||_\infty$ and Proposition 1, we have that

$$||\vec{e}_{k+1}^{j+1}||_\infty \leq C\theta(\Delta\tau)\lambda||\vec{e}^{j}||_\infty. \tag{14}$$

Thus, for sufficiently small $\Delta\tau$, the fixed-point iteration converges.

In practice, with a tolerance of 10^{-9}, and parameters in the range of those in the experiments, the fixed-point iterations converge in around 3 iterations.

2.6 Matrix approximation

Solving an $n \times n$ tridiagonal system takes $O(n)$ time, but a dense matrix-vector multiplication costs $O(n^2)$. Asymptotically, then, the matrix-vector multiplication is more costly than solving the tridiagonal system. In this section, we propose a simple algorithm, which does not alter the asymptotic cost of the multiplication, but could practically save up to half of the work. For the rest of this subsection, we assume that we are working with a rapidly decaying density, such as Merton's (2) and Kou's (3) models. For simplicity, we also assume that the jump is on the down-side. With minor changes, we can treat jumps on the up-side.

Our aim is to find the smallest $i(S)$ such that

$$|\frac{\int_{x_{i(S)}/S}^{x_n/S} V(\tau, S\eta)g(\eta)d\eta}{\max(V(0,S),1)}| \leq C, \tag{15}$$

for some user chosen constant C. The assumption that g is rapidly decaying means that, when S is small, a big region of the integral does not have to be computed, because g has insignificant contribution there.

For a put option or a call option, a no-arbitrage bound can always be obtained. For a put option,

$$|\int_{x_{i(S)}/S}^{x_n/S} V(\tau, S\eta)g(\eta)d\eta| \leq K\int_{x_{i(S)}/S}^{x_n/S} g(\eta)d\eta,$$

where K is the strike. For a call option,

$$|\int_{x_{i(S)}/S}^{x_n/S} V(\tau, S\eta)g(\eta)d\eta| \leq \int_{x_{i(S)}/S}^{x_n/S} S\eta g(\eta)d\eta.$$

Using the above no-arbitrage bounds and certain quantities that are computed through the QSC method, we can, for any $S = m_i, i = 1, \ldots, n$, compute $i(S)$, so that (15) is approximately satisfied. More specifically, through the QSC method, the quantities $\int_{x_j/S}^{x_n/S} g(\eta)d\eta$ and $\int_{x_j/S}^{x_n/S} S\eta g(\eta)d\eta$, for $S = m_i, i = 1, \ldots, n$, and $j = 0, \ldots, n-1$, have been computed. Then, we can compute (an approximation to) $i(S)$ by checking which $i(S)$ satisfies

$$K|\frac{\int_{x_{i(S)}/S}^{x_n/S} g(\eta)d\eta}{\max(V(0,S),1)}| \leq C \tag{16}$$

for a put option and

$$|\frac{\int_{x_{i(S)}/S}^{x_n/S} S\eta g(\eta)d\eta}{\max(V(0,S),1)}| \leq C \tag{17}$$

for a call option.

As in Section 2.4.1, let R be the dense integral matrix. We define the approximation $\tilde{R}$ of R by

$$\tilde{R}_{i,j} = 0 \text{ for } j \geq i_{m_{i-1}}, \quad \tilde{R}_{i,j} = R_{i,j} \text{ else,}$$

where $i_{m_{i-1}} = i(m_{i-1})$ was computed as in (16) or (17), and m_{i-1} is the corresponding collocation point for the ith row.

Suppose we have computed Rz for the dense matrix R and a vector z. Then, for a point z_1 near z, we can compute an approximation y_1 to Rz_1 by

$$y_1 = Rz + \tilde{R}(z_1 - z).$$

From this construction, we have that

$$|y_1 - Rz| \leq |z_1 - z|,$$

so that our approximation error is never greater than the distance between z and z_1. From y_1, we could approximate Rz_2 for a point z_2 near z_1 by

$$y_2 = y_1 + \tilde{R}(z_2 - z_1).$$

In the fixed-point iterations, where typically the maximum change of vectors after 1 or 2 iterations quickly goes down to the order of 10^{-8}, this matrix approximation effectively saves a lot of work that does not materially change the results. The effect of the matrix approximation will be illustrated in Section 3.

If the jump is on the up-side, we find the largest $i(S)$ such that

$$|\frac{\int_0^{x_{i(S)}/S} V(\tau, S\eta)g(\eta)d\eta}{\max(V(0,S),1)}| \leq C, \tag{18}$$

and proceed accordingly.

2.7 American options

The penalty method, developed in [12], can be applied to QSC in valuing American options under jump diffusion models. The application of the penalty method to QSC without jump diffusion was described in [6]. With jump diffusion, the general procedure is similar to the one described in [9]. More specifically, with $\vec{\mathbf{c}}^*$ being the vector of quadratic spline degrees of freedom corresponding to the payoff, and $P(\vec{\mathbf{c}}^j)$ the QSC penalty matrix (scaled version of T_0) defined as in [6], the following iteration takes place at the $(j+1)$st timestep:

Set $\vec{\mathbf{c}}_0^{j+1} = \vec{\mathbf{c}}^j$
Set $P_0^{j+1} = P(\vec{\mathbf{c}}_0^{j+1})$
for $k = 1, \ldots,$ until convergence **do**
  Solve

$$(A + P_{k-1}^{j+1})\vec{\mathbf{c}}_k^{j+1} = D\vec{\mathbf{c}}_{k-1}^{j+1} + \mathbf{u}^{j+1} + P_{k-1}^{j+1}\vec{\mathbf{c}}^*. \tag{19}$$

  Set $P_k^{j+1} = P(\vec{\mathbf{c}}_k^{j+1})$
end for
Set $\vec{\mathbf{c}}^{j+1} = \vec{\mathbf{c}}_k^{j+1}$

3 Numerical experiments

In this section, we present some numerical results. For the Merton model (2), we choose the parameters in Table 1. Note that, for the Merton model, we have $\kappa = \exp(\mu + \gamma^2) - 1$. For Kou's model (3), we choose the parameters in Table 2. For both cases, these parameters are the same as those used in [10].

In the numerical experiments, the CN-Rannacher timestepping scheme was used to discretize the equation along the time direction. Adaptive timestepping, as outlined in [10], with the value of $dnorm$ set to 0.125 for the coarsest grid and divided by 2 for each doubling of n, is used to determine the step size in each timestep. This means that the number, n_t, of timesteps taken is not predetermined. Therefore, there are variations in n_t when different methods are used for the same problem. The space grid is taken to be fixed (but non-uniform) at each timestep, as explained below.

In the QSC experiments, we obtained the non-uniform grid by mapping the uniform grid with the indicated number of points into a non-uniform grid via the mapping defined in [6]. Note that, according to the mapping in [6], a midpoint is placed at the strike. The matrix approximations are obtained with $C = 1$ in (16) or (17). In the finite difference experiments, we implemented the approach outlined in [10]. Essentially the same grid is used in the finite difference experiments for a fair comparison, but the FD grid has to be enlarged or refined to reduce wrap-around errors. Details can be found in [10].

In the following tables, when analytic formulas exist, the term "error" means the signed difference of the computed value minus the exact value at the specified point, while "grid error" and "midpoint error" mean the maximum in absolute value error across the grid and midpoints, respectively. Then, we also define

$$\alpha \equiv \log(|\frac{\text{error}_{n_2}}{\text{error}_{n_1}}|)/\log(\frac{n_1}{n_2}),$$

where $n_2 \approx 2n_1$, as our numerically estimated order of convergence in a non-uniform grid.

We first present the case with an at-the-money European put option, under the jump diffusion model of Merton. We present results from QSC as well as standard finite differences (FDs) as spatial discretization methods.

Table 3 shows the numerical results using QSC (without matrix approximation) for an at-the-money non-dividend-paying put option under the jump diffusion process. Using the same "base" grid (before the added grid points for the FFT wrap-around), the numerical results using the FD method are presented in Table 4. The last column in Tables 3 and 4 shows the average number of fixed-point iterations per timestep. We notice that this decreases, as the mesh and time step sizes decrease. For QSC, as we have shown in (14), the bound for the reduction factor of the error at each fixed-point iteration is proportional to $\Delta\tau$, so we expect a decrease in the average number of fixed-point iterations. Similar results have been shown for finite differences. We also give the price for the same type of option without jumps, using an analytic formula for the solution of the standard Black-Scholes equation. As expected, since $\mu < 0$, the put price without jumps is lower than the one with jumps.

Using the same parameters, we also study the effect of using the matrix approximation. The results without matrix approximation for a case of an out-of-the-money European put option are presented in Table 5, while the respective one with matrix approximation are presented in Table 6. It is observed that the numerical errors and the order of convergence are only minimally affected by the matrix approximation. The fixed-point iteration convergence is not affected at all. In most of the computation, we can work with a reduced matrix instead of a fully dense one.

The robustness of the numerical scheme extends to the computation of the option Greeks, as illustrated in Table 7 for a European call. Second order convergence can be observed in the computation of deltas and gammas. A corresponding analysis for the finite difference method is shown in Table 8. Both methods

perform satisfactorily for the computations of deltas and gammas.

The QSC method applied to Kou's model (3) also gives an almost second order convergence of option values. In this test, we verify this fact for both grid points and midpoints, and show the results in Table 9, where the maximum in absolute value errors at the respective sets of points are displayed. The discontinuity does not interfere with convergence because of the analytic tractability of the moment valuations. We remark that the general numerical treatments as outlined in [10] cannot guarantee second order convergence. Taking into account the last column of Table 9, we also remark, that the convergence of fixed-point iteration for Kou's model is similar to the one for Merton's model.

With matrix approximation, the results do not in general work as fine as those in the Merton's model. This is expected, as the jump size distribution has fatter tails for Kou's model, and so $g(\eta)$ does not decrease as rapidly. More entires are needed to keep the accuracy. Although not shown here, the convergence can get bizarre towards extreme regions (near $S = 0$). The areas near the strike are not affected, as illustrated in Tables 10 and 11. More specifically, the errors with and without approximation agree in about two significant digits, therefore, they are comparable.

Finally, in Tables 12 and 13, we report numerical results from valuing an American option under the jump diffusion models of Merton and Kou, respectively. In these cases, we do not have an exact solution, so the "change" as compared to the half-size grid solution is reported in Tables 12 and 13 instead of the true error. We also report a reference price for Merton's case from the literature. The last columns of each of Tables 12 and 13 indicate that the convergence of fixed-point iteration for American options is similar to the one for European options.

4 Summary and future directions

We developed a QSC spatial discretization method for the PIDE arising in modelling option prices under jump diffusion with various jump size densities. The method treats the integral term of the PIDE in a simple way, based on the piecewise polynomial representation of the approximate solution and on analytical formulae for calculating the zero-th, first and second partial moments of the density function. The integral term discretization gives rise to a dense matrix, the inversion of which is avoided by a fixed-point iteration.

Compared to standard finite differences, the QSC method avoids the transformations to/from the Fourier space and the associated problems with the so-called wrap-around error. The QSC error is, in general, smaller than the respective finite difference error, while both methods are of second order. Second order convergence is observed for the deltas and gammas of the approximation, and, again, QSC exhibits smaller error than finite difference. Though avoiding the Fourier transformation makes the QSC method asymptotically less efficient than finite difference, the method can still be attractive to practitioners due to its simplicity, higher accuracy and ease in calculating the Greeks. Furthermore, the QSC method can still be easily extended to price American and other non-vanilla options.

The technique for treating the integral term of the PIDE presented in this paper is applicable to any piecewise polynomial finite element method. The QSC (or other finite element method) can be combined with adaptive mesh methods, to make the method more efficient for a given number of spatial points. The version of QSC that exhibits fourth order of convergence on certain sets of points can also be incorporated to further improve the performance of the method.

A QSC matrices

In this section, we provide the implementation details of the QSC method for a uniform grid. Suppose we work on the computational domain $[a, b]$. Let $h = (b-a)/n$ be the spatial step size, and $x_i = a+ih, i = 0, \ldots, n$, be the grid points. The collocation points are $m_0 = x_0 = a, m_i = \frac{x_{i-1}+x_i}{2}, i = 1, \ldots, n$, and $m_{n+1} = x_n = b$. There are $n+2$ collocation points, and so the collocation system involves $n+2$ equations.

The QSC matrices for the zeroth, first, and second derivatives are:

$$T_0 = \frac{1}{8}\begin{bmatrix} 4 & 4 & & & & \\ 1 & 6 & 1 & & & \\ & 1 & 6 & 1 & & \\ & & \cdot & \cdot & \cdot & \\ & & & 1 & 6 & 1 \\ & & & & 4 & 4 \end{bmatrix} \tag{20}$$

$$T_1 = \frac{1}{2h}\begin{bmatrix} -2 & 2 & & & & \\ -1 & 0 & 1 & & & \\ & -1 & 0 & 1 & & \\ & & \cdot & \cdot & \cdot & \\ & & & -1 & 0 & 1 \\ & & & & -2 & 2 \end{bmatrix} \tag{21}$$

$$T_2 = \frac{1}{h^2}\begin{bmatrix} 0 & 0 & & & & \\ 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \cdot & \cdot & \cdot & \\ & & & 1 & -2 & 1 \\ & & & & 0 & 0 \end{bmatrix}. \tag{22}$$

Now suppose we have a jump density $g(x)$. Define the matrices U, V and W, of size $n \times n$, by

$$U_{i,j} = m_i^2 \int_{\frac{x_{j-1}}{m_i}}^{\frac{x_j}{m_i}} x^2 g(x)dx, \quad i, j = 1, \ldots, n.$$

$$V_{i,j} = m_i \int_{\frac{x_{j-1}}{m_i}}^{\frac{x_j}{m_i}} x g(x)dx, \quad i, j = 1, \ldots, n,$$

$$W_{i,j} = \int_{\frac{x_{j-1}}{m_i}}^{\frac{x_j}{m_i}} g(x)dx, \quad i, j = 1, \ldots, n.$$

Define $n \times (n+2)$ matrices A, B, and C that "extract" the piecewise quadratic polynomial terms as follows. For $i = 1, ..., n$,

$$A_{i,i} = \frac{1}{2h^2}, \quad A_{i,i+1} = -\frac{1}{h^2}, \quad A_{i,i+2} = \frac{1}{2h^2},$$

$$B_{i,i} = -\frac{i}{h}, \quad B_{i,i+1} = \frac{2i-1}{h}, \quad B_{i,i+2} = -\frac{i-1}{h},$$

$$C_{i,i} = \frac{i^2}{2}, \quad C_{i,i+1} = -\frac{2(i-1)^2 + 2(i-1) - 1}{2}, \quad C_{i,i+2} = \frac{(i-1)^2}{2},$$

and zero otherwise.

Mathematically, A, B, C take a vector of degrees of freedom, which is of size $n+2$, and map it to the second, first and zeroth, respectively, polynomial coefficient in each subinterval $[x_{l-1}, x_l]$, $l = 1, \ldots, n$. For example, if $\vec{\mathbf{c}}$ is the vector of degrees of freedom of V_Δ, $A\mathbf{c} = [a_1, a_2, \ldots, a_n]^T$, where $a_i, i = 1, \ldots, n$, are the quadratic coefficients in the $[x_{i-1}, x_i]$ branch of V_Δ, as in Section 2.4.1.

The dense matrix $\hat{R}$ is then given as follows. First, define

$$\hat{R} = U \times A + V \times B + W \times C.$$

Now $\hat{R}$ is of size $n \times (n+2)$. We extend $\hat{R}$ to an $(n+2) \times (n+2)$ matrix R, by adding identically zero first and last rows.

B Transformation of system

In this section, we explain how the system (10) is transformed to (12).

Consider a generic $(n+2) \times (n+2)$ linear system $B\mathbf{x} = \mathbf{r}$ arising from the QSC discretization of a PIDE with Dirichlet boundary conditions. The QSC discretization of the Dirichlet boundary conditions implies that the first and last equations of the system are

$$\frac{1}{2}\mathbf{x}_1 + \frac{1}{2}\mathbf{x}_2 = \mathbf{r}_1, \quad \frac{1}{2}\mathbf{x}_{n+1} + \frac{1}{2}\mathbf{x}_{n+2} = \mathbf{r}_{n+2},$$

which result in

$$\mathbf{x}_1 = 2\mathbf{r}_1 - \mathbf{x}_2, \quad \mathbf{x}_{n+2} = 2\mathbf{r}_{n+2} - \mathbf{x}_{n+1}.$$

Substituting $\mathbf{x}_1 = 2\mathbf{r}_1 - \mathbf{x}_2$ and $\mathbf{x}_{n+2} = 2\mathbf{r}_{n+2} - \mathbf{x}_{n+1}$ in all other equations results in a transformed $n \times n$ linear system $B'\mathbf{x}' = \mathbf{r}'$, with

$$\begin{aligned} B'_{i,1} = B_{i+1,2} - B_{i+1,1}, \quad B'_{i,n} = B_{i+1,n+1} - B_{i+1,n+2}, \\ \mathbf{r}'_i = \mathbf{r}_{i+1} - 2B_{i+1,1}\mathbf{r}_1 - 2B_{i+1,n+2}\mathbf{r}_{n+2}, \quad \mathbf{x}'_i = \mathbf{x}_{i+1}, \end{aligned}$$

for $i = 1, \ldots, n$, while, for the rest of the entries of B' we have $B'_{i,j} = B_{i+1,j+1}$. Clearly, if $B'\mathbf{x}' = \mathbf{r}'$ is solved, the values $\mathbf{x}_i, i = 2, \ldots, n+1$, have been computed, and then the values of $\mathbf{x}_1$ and $\mathbf{x}_{n+2}$ can be easily recovered from $\mathbf{x}_1 = 2\mathbf{r}_1 - \mathbf{x}_2$ and $\mathbf{x}_{n+2} = 2\mathbf{r}_{n+2} - \mathbf{x}_{n+1}$. Thus, solving $B'\mathbf{x}' = \mathbf{r}'$ is mathematically equivalent to solving $B\mathbf{x} = \mathbf{r}$. Note also that $||B'||_\infty = O(||B||_\infty)$.

C Boundedness of inverse matrix

In this section, we give the proof of the boundedness of $||A'^{-1}||_\infty$. Let $A = T_0 - \theta(\Delta\tau)T_L$, and recall that T_L corresponds to the differential part of $\mathcal{L}$, and has zero first and last rows. Suppose our spatial computational domain is $[a, b]$ with $a > 0$. Also assume that $r - \lambda\kappa > 0$. It is easy to see that the diagonal entries of T_L are negative, as they are of the form

$$(T_L)_{i,i} = -\frac{1}{h^2}\sigma^2 m_{i-1}^2 - \frac{6(r+\lambda)}{8}, i = 2, \ldots, n+1.$$

For h sufficiently small, the off-diagonal entries of T_L are non-negative, as they are of the form

$$(T_L)_{i,i\pm1} = \frac{1}{2h^2}\sigma^2 m_{i-1}^2 \mp \frac{1}{2h}(r - \lambda\kappa)m_{i-1} - \frac{(r+\lambda)}{8}, i = 2, \ldots, n+1,$$

and the $O(h^{-2})$ term dominates.

Now consider the transformation of the matrix A and its constituent matrices T_L and T_0 to A', T'_L and T'_0, respectively, as explained in Appendix B. The negativity of the diagonal entries and the non-negativity of the off-diagonal entries of T'_L are clearly preserved.

For notational convenience, define the operator $\mathcal{K}$ as follows. Let M be a matrix. Define $\mathcal{K}M$ to be the column vector such that the i-th element of $\mathcal{K}M$ is $|M_{i,i}| - |M_{i,i+1}| - |M_{i,i-1}|$. If M is tridiagonal, then, clearly, diagonal dominance of M is equivalent to $\mathcal{K}M > 0$.

We study the sign of the elements of $\mathcal{K}A'$. Let i be a given interior index, $i = 2, \ldots, n+1$. In any case, it is clear that $A'_{i,i} > 0$. For the off-diagonal entries of A', there are three cases:

- $A'_{i,i+1} > 0$ and $A'_{i,i-1} > 0$. In this case, the i-th entry of $\mathcal{K}A'$ is

$$A'_{i,i} - A'_{i,i-1} - A'_{i,i+1} = \frac{1}{2} - \theta(\Delta\tau)(T_L{}'_{i,i} - T_L{}'_{i,i-1} - T_L{}'_{i,i+1}).$$

  However, since $T_L{}'_{i,i} < 0$ and $T_L{}'_{i,i\pm1} > 0$, the term $T_L{}'_{i,i} - T_L{}'_{i,i-1} - T_L{}'_{i,i+1}$ is negative. So $(\mathcal{K}A')_i > \frac{1}{2}$.

- $A'_{i,i+1} < 0$ and $A'_{i,i-1} < 0$. In this case, the i-th entry of $\mathcal{K}A'$ is

$$A'_{i,i} + A'_{i,i-1} + A'_{i,i+1} = 1 - \theta(\Delta\tau)(T_L{}'_{i,i} + T_L{}'_{i,i-1} + T_L{}'_{i,i+1}).$$

  It is easy to see that

$$T_L{}'_{i,i} + T_L{}'_{i,i-1} + T_L{}'_{i,i-1} = -(r+\lambda).$$

  So $(\mathcal{K}A')_i = 1 + \theta(\Delta\tau)(r+\lambda) > 1$.

- $A'_{i,i+1} < 0$ and $A'_{i,i-1} > 0$. This is the only possible different-sign case for off-diagonal entries, because, if $r - \lambda\kappa > 0$, we have

$$T'_{Li,i+1} > T'_{Li,i-1}.$$

  In this case, the i-th entry of $\mathcal{K}A'$ is

$$\begin{aligned} A'_{i,i} - A'_{i,i-1} + A'_{i,i+1} &= \frac{3}{4} - \theta(\Delta\tau)(T_L{}'_{i,i} - T_L{}'_{i,i-1} + T_L{}'_{i,i+1}) \\ &> \frac{3}{4}, \end{aligned}$$

  where the term $T_L{}'_{i,i} - T_L{}'_{i,i-1} + T_L{}'_{i,i+1}$ is negative because it is of the form

$$\begin{aligned} T_L{}'_{i,i} - T_L{}'_{i,i-1} + T_L{}'_{i,i+1} &= -\frac{1}{h^2}\sigma^2 m_{i-1}^2 - \frac{6(r+\lambda)}{8} \\ &\quad -\frac{1}{2h^2}\sigma^2 m_{i-1}^2 + \frac{(r+\lambda)}{8} + \frac{1}{2h}(r-\lambda\kappa)m_{i-1} \\ &\quad +\frac{1}{2h^2}\sigma^2 m_{i-1}^2 - \frac{(r+\lambda)}{8} + \frac{1}{2h}(r-\lambda\kappa)m_{i-1} \\ &= -\frac{1}{h^2}\sigma^2 m_{i-1}^2 - \frac{6(r+\lambda)}{8} + \frac{1}{h}(r-\lambda\kappa)m_{i-1} \end{aligned}$$

  and the $O(h^{-2})$ term dominates for h sufficiently small.

Thus, $(\mathcal{K}A')_i > \frac{1}{2}$, for any interior index i.

As for the boundary cases ($i = 1$ and $i = n$), the proof follows the same lines, with only two entries being involved. Thus, $\mathcal{K}A' > \frac{1}{2}$, from which we can bound $||(A')^{-1}||_\infty$, independently of n.

It is worth noting that the proof technique for the boundedness of $||(A')^{-1}||_\infty$ cannot be used to prove the boundedness of $||A^{-1}||_\infty$, because the first and last rows of A are not strictly diagonally dominant. However, we verified numerically that $||A^{-1}||_\infty$ is bounded independently of the mesh size.

Acknowledgement

This research was supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada.

References

[1] K. I. Amin, Jump diffusion option valuation in discrete time, J. Finance 48 (5) (1993) 1833–63.

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

[3] G. Bakshi, Z. Chen, An alternative valuation model for contingent claims, J. Financial Econ. 44 (1997) 123–165.

[4] D. S. Bates, Jumps and stochastic volatility: Exchange rate processes implicit in Deutsche Mark options, Rev. Financial Stud. 9 (1) (1996) 69–107.

[5] P. Carr, A. Mayo, On the numerical evaluation of option prices in jump diffusion processes, European J. Finance 13 (4) (2007) 353–372.

[6] C. Christara, D. M. Dang, Adaptive and high-order methods for valuing American options, J. Comput. Finance 14 (4) (2011) 73–113.

[7] C. C. Christara, T. Chen, D. M. Dang, Quadratic spline collocation for one-dimensional linear parabolic partial differential equations, Numer. Algorithms 53 (4) (2010) 511–553.

[8] T. F. Coleman, Y. Li, A. Verma, Reconstructing the unknown local volatility function, J. Comput. Finance 2 (3) (1999) 77–102.

[9] Y. d'Halluin, P. A. Forsyth, G. Labahn, A penalty method for American options with jump diffusion processes, Numer. Math. 97 (2) (2004) 321–352.

[10] 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) 87–112.

[11] B. Dupire, Pricing with a smile, Risk 7 (1) (1994) 18–20.

[12] P. A. Forsyth, K. R. Vetzal, Quadratic convergence for valuing American options using a penalty method, SIAM J. Sci. Comput. 23 (6) (2002) 2095–2122.

[13] M. B. Giles, R. Carter, Convergence analysis of Crank-Nicolson and Rannacher time-marching, J. Comput. Finance 9 (4) (2006) 89–112.

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

[15] J. Hull, A. White, The pricing of options on assets with stochastic volatilities, J. Finance 42 (1987) 281–300.

[16] K. R. Jackson, S. Jaimungal, V. Surkov, S. F. Building, Fourier space time-stepping for option pricing with Lévy models, J. Comput. Finance 12 (2) (2008) 1–29.

[17] S. G. Kou, A jump-diffusion model for option pricing, Management Sci. 48 (8) (2002) 1086–1101.

[18] R. C. Merton, Option pricing when underlying stock returns are discontinuous, J. Financial Econ. 3 (1–2) (1976) 125–144.

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

volatility σ	0.15
expiry T	0.25
interest rate r	0.05
strike K	100
mean arrival rate λ	0.1
mean of jump size μ	-0.9
standard deviation of jump size γ	0.45

Table 1: Basic parameters for Merton's model.

volatility σ	0.15
expiry T	0.25
interest rate r	0.05
strike K	100
mean arrival rate λ	0.1
η_1	3.0465
η_2	3.0775
p	0.3445

Table 2: Basic parameters for Kou's model.

n	n_t	Value	Error	α	Aver.no.fix.iter.
64	26	3.150453	1.4274e-03		3.88
128	50	3.149463	4.3756e-04	1.9846	3.68
256	96	3.149144	1.1885e-04	2.0116	3.53
512	188	3.149056	3.0860e-05	2.0085	3.21
1024	373	3.149034	8.5562e-06	1.8797	2.98

Table 3: QSC without matrix approximation for an at-the-money European put, Merton parameters in Table 1. For comparison, the value without jumps is 2.39284975.

n	n_t	Value	Error	α	Aver.no.fix.iter
77	19	3.145145	-3.8759e-03		3.47
154	37	3.148064	-9.6161e-04	1.7446	3.19
308	74	3.148789	-2.3654e-04	1.8915	2.96
615	149	3.148967	-5.8751e-05	1.9488	2.97
1229	300	3.149011	-1.4629e-05	1.9757	2.89

Table 4: Finite Difference for an at-the-money European put, Merton parameters in Table 1.

n	n_t	Value	Error	α	Aver.no.fix.iter.
64	26	1.904609	-1.3401e-03		3.88
128	50	1.905562	-3.8645e-04	2.0871	3.68
256	96	1.905850	-9.8883e-05	2.1037	3.53
512	188	1.905923	-2.6193e-05	1.9788	3.21
1024	373	1.905943	-6.2586e-06	2.0977	2.98

Table 5: QSC without matrix approximation for European put, $S = 105$, Merton parameters in Table 1. For comparison, the value without jumps is 0.89018298.

n	n_t	Value	Error	α	Ratio of entries kept	Aver.no.fix.iter.
64	26	1.904574	-1.3749e-03		0.55587	3.88
128	50	1.905544	-4.0505e-04	1.7632	0.54868	3.68
256	96	1.905841	-1.0867e-04	1.8981	0.54494	3.53
512	188	1.905918	-3.1001e-05	1.8096	0.54311	3.20
1024	373	1.905941	-8.6987e-06	1.8334	0.54219	2.98

Table 6: QSC with matrix approximation for European put, $S = 105$, Merton parameters in Table 1.

n	n_t	Delta error	α_{Delta}	Gamma error	α_{Gamma}	Ratio of entries kept
64	31	-4.3706e-04		1.2603e-04		0.60393
128	60	-1.3173e-04	1.7302	3.8214e-05	1.7215	0.56869
256	117	-3.6887e-05	1.8364	1.0376e-05	1.8809	0.55130
512	231	-1.0172e-05	1.8586	2.5841e-06	2.0055	0.54290
1024	458	-2.8915e-06	1.8146	6.0489e-07	2.0949	0.53878

Table 7: QSC with matrix approximation for an at-the-money European call, Merton parameters in Table 1.

n	n_t	Delta error	α_{Delta}	Gamma error	α_{Gamma}
77	22	-1.1508e-03		-3.2605e-05	
154	45	-3.4375e-04	1.7432	-2.6823e-05	0.2816
308	91	-9.2714e-05	1.8905	-9.3338e-06	1.5229
615	184	-2.4005e-05	1.9494	-2.6920e-06	1.7937
1229	371	-6.1031e-06	1.9757	-7.1935e-07	1.9035

Table 8: Finite difference for an at-the-money European call, Merton parameters in Table 1.

n	n_t	Grid error	α_{grid}	Midpoint error	α_{mid}	Aver.no.fix.iter.
64	24	2.6599e-03		2.6057e-03		3.71
128	45	7.9732e-04	1.7381	7.9066e-04	1.7206	3.60
256	87	2.1641e-04	1.8814	2.1622e-04	1.8705	3.41
512	171	5.6084e-05	1.9481	5.6056e-05	1.9476	3.09
1024	338	1.4999e-05	1.9027	1.4999e-05	1.9020	2.97

Table 9: QSC without matrix approximation for a European put, Kou parameters in Table 2. Maximum errors across grid points and midpoints shown.

		Error for		Error for		Error for	
n	n_t	$S = 95$	α_{95}	$S = 100$	α_{100}	$S = 105$	α_{105}
64	24	1.9637e-03		2.3318e-03		-1.8033e-04	
128	45	5.8531e-04	1.7463	7.0491e-04	1.7259	-4.4851e-05	2.0075
256	87	1.6305e-04	1.8439	1.9177e-04	1.8781	-6.2841e-06	2.8354
512	171	4.2028e-05	1.9559	4.9646e-05	1.9496	-2.2785e-06	1.4636
1024	338	1.1469e-05	1.8736	1.3278e-05	1.9027	-2.3481e-07	3.2785

Table 10: QSC without matrix approximation for a European put, Kou parameters in Table 2.

		Error for		Error for		Error for		Ratio of
n	n_t	$S = 95$	α_{95}	$S = 100$	α_{100}	$S = 105$	α_{105}	entries kept
64	24	1.9502e-03		2.3284e-03		-1.8153e-04		0.73580
128	45	5.7803e-04	1.7544	7.0339e-04	1.7270	-4.5232e-05	2.0048	0.72127
256	87	1.5963e-04	1.8564	1.9113e-04	1.8798	-6.3509e-06	2.8323	0.71456
512	171	4.0309e-05	1.9856	4.9366e-05	1.9526	-2.2717e-06	1.4832	0.71146
1024	338	1.0630e-05	1.9230	1.3153e-05	1.9082	-2.1758e-07	3.3841	0.71005

Table 11: QSC with matrix approximation for a European put, Kou parameters in Table 2.

n	n_t	Value	Absolute change	α	Ratio of entries kept	Aver.no.fix.iter.
64	27	3.2366028			0.59186	3.70
128	51	3.2399223	3.3195e-03		0.56316	3.68
256	98	3.2408754	9.5304e-04	1.8004	0.55167	3.42
512	191	3.2411502	2.7483e-04	1.7940	0.54638	3.09
1024	378	3.2412255	7.5242e-05	1.8689	0.54381	2.99

Table 12: QSC with matrix approximation for an at-the-money American put, Merton parameters in Table 1. Reference price = 3.241251 from [10] using a finite difference method with grid size 4065. For comparison, the value without jumps is 2.50462405, using the method in [6] with grid size 320.

n	n_t	Value	Absolute change	α	Ratio of entries kept	Aver.no.fix.iter.
64	25	2.8043801			0.73580	3.88
128	47	2.8068893	2.5092e-03		0.72127	3.74
256	89	2.8075644	6.7509e-04	1.8941	0.71456	3.49
512	174	2.8077951	2.3074e-04	1.5488	0.71146	3.12
1024	344	2.8078571	6.1974e-05	1.8965	0.71005	2.99

Table 13: QSC with matrix approximation for an at-the-money American put, Kou parameters in Table 2.