THE EVALUATION OF AMERICAN OPTION PRICES UNDER STOCHASTIC VOLATILITY AND JUMP-DIFFUSION DYNAMICS USING THE METHOD OF LINES

CARL CHIARELLA*
School of Finance and Economics, University of Technology
Sydney, PO Box 123, Broadway, NSW 2007, Australia
carl.chiarella@uts.edu.au

BODA KANG
School of Finance and Economics
University of Technology, Sydney
boda.kang@uts.edu.au

GUNTER H. MEYER
School of Mathematics
Georgia Institute of Technology, Atlanta
meyer@math.gatech.edu

ANDREW ZIOGAS
Integral Energy, Australia
Andrew.Ziogas@integral.com.au

Received 20 March 2008
Accepted 23 September 2008

This paper considers the problem of numerically evaluating American option prices when the dynamics of the underlying are driven by both stochastic volatility following the square root process of Heston [18], and by a Poisson jump process of the type originally introduced by Merton [25]. We develop a method of lines algorithm to evaluate the price as well as the delta and gamma of the option, thereby extending the method developed by Meyer [26] for the case of jump-diffusion dynamics. The accuracy of the method is tested against two numerical methods that directly solve the integro-partial differential pricing equation. The first is an extension to the jump-diffusion situation of the componentwise splitting method of Ikonen and Toivanen [21]. The second method is a Crank-Nicolson scheme that is solved using projected successive over relaxation and which is taken as the benchmark for the price. The relative efficiency of these methods for computing the American call option price, delta, gamma and free boundary is analysed. If one seeks an algorithm that gives not only the price but also the delta and gamma to the same level

*Corresponding author.
of accuracy for a given computational effort then the method of lines seems to perform best amongst the methods considered.

Keywords: American options; stochastic volatility; jump-diffusion processes; Volterra integral equations; free boundary problem; method of lines.

1. Introduction

Derivative securities are commonly written on underlying assets with return dynamics that are not sufficiently well described by the geometric Brownian motion process proposed by Black and Scholes [6] and Merton [24]. There have been numerous efforts to develop alternative asset return models that are capable of capturing the leptokurtic features found in financial market data, and subsequently use these models to develop option prices that accurately reflect the volatility smiles and skews found in market traded options. There are two classical ways of developing option pricing models that are capable of generating such behaviour; the first is to add jumps into the price process for the underlying asset, as originally proposed by Merton [25]; the second is to allow the volatility to evolve stochastically, for instance according to the square-root process introduced by Heston [18].

While both alternative models have proven valuable in capturing the leptokurtosis found in realised market returns, Cont and Tankov [12] indicate that a model combining both jump-diffusion and stochastic volatility features can lead to even better results. Such a model is proposed by Bates [5], combining the features of the models by Merton [25] and Heston [18]. A similar model is considered by Scott [30], generalised to allow for stochastic interest rates. Scott explores the pricing of European options under these dynamics, but he does not consider American options.

Apart from the work of Yan and Hanson [35] and Hanson and Yan [17], there seems to have been very little research on American option pricing under both stochastic volatility and jump-diffusion dynamics, despite the fact that many traded options contain early exercise features. In this paper we consider the problem of numerically evaluating American options under the combined stochastic volatility and jump-diffusion model of Bates [5].

There exists a considerable amount of work on the numerical evaluation of American options under either jump-diffusion or stochastic volatility. For the jump-diffusion case, several numerical methods have been proposed. There are numerous examples of finite difference methods, including Carr and Hirsa [8], d’Halluin et al. [13], Briani et al. [7] and Andersen and Andreasen [4]. Other methods include numerical integration [10], finite elements [23], tree methods [3] and the method of lines [26].

Numerical methods for American option pricing under stochastic volatility are less frequently encountered in the current literature. Tzavalis and Wang [33] use a numerical integration approach, facilitated by the assumption that the early exercise surface is an exponential-linear function of the volatility. Adolfsson et al. [2] have also implemented a version of this method. Ikonen and Toivanen [20, 21] use a finite
difference approach featuring operator splitting, and there are several examples featuring multi-grid finite difference methods, including Clarke and Parrott [11], Oosterlee [28] and Reisinger and Wittum [29]. The method of lines has also been applied to this problem by Adolfsson et al. [2].

To date there does not seem to be a great deal of literature on numerical methods for dealing with American options under both stochastic volatility and jump-diffusion dynamics. In this paper we extend the method of lines solution proposed by Meyer [26] for American option pricing under jump-diffusion dynamics to allow for stochastic volatility. One of the strengths of this method is that the option price, delta, gamma and free boundary are all computed as part of the solution process. As a benchmark for the method of lines, we consider two finite difference schemes. The first is a standard two-dimensional Crank-Nicholson implicit scheme solved using projected successive over-relaxation (PSOR) techniques, with appropriate adjustments to deal with the integral over the jumps term. We use this algorithm with a large order of discretisation as the “true” solution for the option price. The second method we consider is a generalisation of the componentwise splitting algorithm of Ikonen and Toivanen [21] to include jumps. We use these finite difference methods to verify the accuracy of the method of lines algorithm, and then explore the impact that stochastic volatility and jump-diffusion has on the early exercise boundary for the American call.

The remainder of the paper is structured as follows. Section 2 outlines the free boundary problem that arises from pricing an American call option under stochastic volatility and jump-diffusion dynamics. Section 3 outlines in detail the method of lines for an American call option under stochastic volatility and jump-diffusion. Section 4 summarises our implementation of the componentwise splitting algorithm, which basically extends Ikonen and Toivanen [21] to incorporate the integral over the jumps term. Under the assumption that jump sizes are log-normally distributed, we implement the method of lines and provide some numerical examples in Sec. 5. We benchmark the accuracy of the method of lines against a Crank-Nicholson finite difference scheme and the componentwise splitting approach. In Sec. 6, we explore the impact of both jump-diffusion and stochastic volatility on the early exercise boundary and American call prices. Concluding remarks are presented in Sec. 7.

2. Problem Statement: The Merton-Heston Model

Let $C(S, v, \tau)$ be the price of an American call option written on a stock of price $S$ with time to expiry $\tau$ and strike price $K$. For the underlying dynamics, we assume that the stochastic differential equation (SDE) for $S$ is given by the jump-diffusion process proposed by Merton [25], in conjunction with the square root volatility process by Heston [18]. Thus the dynamics for $S$ under the so-called historical

\[^1\]Note that $\tau = T - t$, where $T$ is the maturity date of the option and $t$ is time.
measure $Q$ are governed by the SDE system
\begin{align}
dS &= (\mu - \lambda k)Sdt + \sqrt{v}SdZ_1 + (Y - 1)SdN, \tag{2.1} \\
dv &= \kappa_v(\theta - v)dt + \sigma \sqrt{v}dZ_2. \tag{2.2}
\end{align}

In (2.1), $\mu$ is the instantaneous stock return per unit time, $v$ is the instantaneous squared stock volatility per unit time, and $Z_1$ is a standard Wiener process under $Q$. Furthermore, we define the Poisson jump arrival process $N$ by
\[ dN = \begin{cases} 
1, & \text{with probability } \lambda dt, \\
0, & \text{with probability } (1 - \lambda dt),
\end{cases} \]
and set
\[ k = \mathbb{E}_Q[(Y - 1)] = \int_0^\infty (Y - 1)G(Y)dY, \]
where $G(Y)$ is the continuous probability density function for the multiplicative jump sizes, $Y$, generated by the measure $Q$. In (2.2), $\theta$ is the long-run mean for $v$, $\kappa_v$ is the rate of mean reversion, $\sigma$ is the instantaneous variance of $v$ per unit time, and $Z_2$ is a standard Wiener process under $Q$ correlated with $Z_1$ such that $\mathbb{E}[dZ_1dZ_2] = \rho dt$. Note that $dN, Y, dZ_1$ and $dZ_2$ are otherwise uncorrelated.

Let $r$ be the risk-free rate of interest, and assume that the stock pays a continuously compounded dividend yield at rate $q$. Here we assume that $r$ and $q$ are both constant, although the results which follow can be readily generalised to facilitate the case where $r$ and $q$ are deterministic functions of time. We assume, following Heston [18], that the market price of volatility risk is proportional to $\sqrt{v}$, and is of the form $\lambda_v \sqrt{v}$, where $\lambda_v$ is a constant. Using standard hedging arguments and an application of Itô’s lemma for jump-diffusion processes, it can be shown$^2$ that $C$ satisfies the integro-partial differential equation (IPDE)
\begin{align}
\frac{\partial C}{\partial \tau} &= \frac{vS^2}{2} \frac{\partial^2 C}{\partial S^2} + \rho \sigma vS \frac{\partial^2 C}{\partial S \partial v} + \frac{\sigma^2 v}{2} \frac{\partial^2 C}{\partial v^2} \\
&\quad + \left( r - q - \lambda \int_0^\infty (1 - \lambda_J(Y))(Y - 1)G(Y)dY \right) S \frac{\partial C}{\partial S} \\
&\quad + (\kappa_v \theta - (\kappa_v + \lambda_v)v) \frac{\partial C}{\partial v} \\
&\quad - rC + \lambda \int_0^\infty (1 - \lambda_J(Y))[C(SY, v, \tau) - C(S, v, \tau)]G(Y)dY, \tag{2.3}
\end{align}
in the region $0 \leq \tau \leq T$, $0 < S \leq b(v, \tau)$, and $0 \leq v < \infty$, where $b(v, \tau)$ denotes the early exercise boundary at time to maturity $\tau$ and variance level $v$, and $\lambda_J(Y)$ denotes the market price of risk associated with a jump in the value of the stock.

$^2$See for example Cheang et al. [9]. This paper also discusses the role the assumptions concerning the market prices of volatility risk and jump risk play in choosing a risk-neutral pricing measure $Q^*$. 
with magnitude \( Y \) (that is a jump from \( S \) to \( SY \)). The initial condition for (2.3) is the American call payoff function, given by

\[
C(S, v, 0) \equiv c(S, v) = \max(S - K, 0).
\]

The boundary condition at \( v = 0 \) is less obvious. Some authors (such as [11, 21]) prefer to use the exercise payoff condition \( C(S, 0, \tau) = \max(S - K, 0) \). Alternatively we note that by setting \( v = 0 \) in (2.3) we find that \( C(S, 0, \tau) \) is given by the solution of the IPDE

\[
\frac{\partial C(S, 0, \tau)}{\partial \tau} = \left( r - q - \lambda \int_0^\infty (1 - \lambda f(Y))(Y - 1)G(Y)dY \right) S \frac{\partial C(S, 0, \tau)}{\partial S} \\
+ \kappa v \theta \frac{\partial C(S, 0, \tau)}{\partial v} - rC(S, 0, \tau) + \lambda \int_0^\infty (1 - \lambda f(Y))[C(SY, 0, \tau) - C(S, 0, \tau)]G(Y)dY.
\]

(2.5)

In this paper, we have preferred to, in a sense, let the model itself yield the boundary condition by extrapolating the solution for \( v \approx 0 \) to \( v = 0 \) as detailed in Sec. 3. This procedure is justified by the fact that the boundary condition at \( v = 0 \) is an outflow condition as is demonstrated in the Appendix. Also we have found that it yields the same solution as solving (2.5) but is computationally more efficient to implement.

In the limit as \( v \to \infty \) the option price becomes insensitive to the variance and so we have the condition

\[
\lim_{v \to \infty} \frac{\partial C}{\partial v}(S, v, \tau) = 0.
\]

(2.6)

In the asset domain, the boundary conditions are

\[
C(0, v, \tau) = 0,
\]

(2.7)

\[
C(b(v, \tau), v, \tau) = b(v, \tau) - K.
\]

(2.8)

Condition (2.7) is the trivial condition that the option is worthless when the stock price falls to zero. Condition (2.8) is the value-matching condition, and follows because the option value is equal to the payoff at the free boundary. Finally, to avoid arbitrage opportunities, we require two additional conditions along the early exercise surface, namely

\[
\lim_{S \to b(v, \tau)} \frac{\partial C}{\partial S} = 1, \quad \lim_{S \to b(v, \tau)} \frac{\partial C}{\partial v} = 0.
\]

(2.9)

The boundary conditions (2.9) are referred to in the literature as the smooth-pasting conditions, and these follow by assuming that option holders will select their exercise strategy so as to maximise the value of the American call option. Mathematically, this is equivalent to ensuring that \( \partial C/\partial S \) and \( \partial C/\partial v \) will be continuous for all values of \( S \). Figure 1 demonstrates the payoff, price profile and early exercise boundary for the American call under consideration.
The IPDE (2.3) may be written in the form
\[
\frac{\partial C}{\partial \tau} = \frac{vS^2}{2} \frac{\partial^2 C}{\partial S^2} + \rho \sigma vS \frac{\partial^2 C}{\partial S \partial v} + \frac{\sigma^2 v}{2} \frac{\partial^2 C}{\partial v^2} + (r - q - \lambda^* k^*)S \frac{\partial C}{\partial S} + \lambda^* \int_0^\infty C(SY, v, \tau)G^*(Y) dY,
\]
(2.10)
where
\[
\lambda^* E_{\mathbb{Q}^*}[F(Y)] = \lambda^* \int_0^\infty F(Y)G^*(Y) dY \equiv \lambda \int_0^\infty F(Y)(1 - \lambda^* Y)G(Y) dY,
\]
(2.11)
and
\[
k^* \equiv E_{\mathbb{Q}^*}[(Y - 1)],
\]
(2.12)
with \(\alpha \equiv \kappa_v \theta\) and \(\beta \equiv \kappa_v + \lambda_v\), and we note that
\[
\int_0^\infty C(S, v, \tau)G^*(Y) dY = C(S, v, \tau),
\]
where \(G^*\) is a density function under the risk-neutral pricing measure \(\mathbb{Q}^*\). The relevant boundary conditions are (2.4)–(2.9), and the domain for the problem is \(0 \leq \tau \leq T, 0 \leq S \leq b(v, \tau)\) and \(0 \leq v < \infty\).

Option pricing under stochastic volatility and jump-diffusion dynamics of course involves market incompleteness since both the volatility risk and the jump risk are not priced in the market. When the market is incomplete option pricing formulae are not unique. In (2.10) the non-uniqueness is reflected in the market price of volatility risk parameter \(\lambda_v\) (embedded in the parameter \(\beta\)) and the market price of jump risk \(\lambda_J(Y)\) that is embedded in the parameter \(\lambda^*\) and jump-size distribution under the adjusted measure \(\mathbb{Q}^*\) [see Eq. (2.11)]. There is a large literature on how these parameters may be chosen, for instance by minimising the variance of hedging cost or some entropy measure. Here we shall simply assume that the parameters
λ, λ∗ as well as the parameters of the distribution G∗ have somehow been obtained either by the methods referred to or simply by calibration to market data.

Our task is to develop a numerical scheme to solve the IPDE (2.10) subject to the boundary conditions (2.4)–(2.9). The main numerical challenges arise from, (i) the two spatial dimensions (S and v) so that we are seeking an early exercise surface, and (ii) the integral term over the jump-size distribution which will involve the unknown option price over a whole set of values of S.

3. Numerical Solution Using the Method of Lines

A wide range of finite difference and finite element methods in the literature are used to price American options under stochastic volatility or jump-diffusion. Our task is to adapt some of these methods to the situation of combined stochastic volatility/jump diffusion dynamics. One method that seems interesting to so adapt is the method of lines, which has been applied to American options with constant volatility by Meyer and van der Hoek [27], with extensions to the jump-diffusion setting by Meyer [26], and to the stochastic volatility case by Adolfsson et al. [2].

The method of lines has several advantages when dealing with American options; the main one perhaps being that the price, free boundary, delta and gamma are all found as part of the computation. The method discretises the IPDE in an intuitive manner, and is readily adapted to be second order accurate in time. In this section we adapt the implementations of Meyer [26] to also incorporate stochastic volatility.

The key idea behind the method of lines is to replace an IPDE with an equivalent system of one-dimensional integro-differential equations (IDEs), whose solution is more readily obtained using numerical techniques. When volatility is constant, the system of IDEs is developed by discretising the time derivative. For the IPDE (2.10), we must also discretise the derivative terms involving the variance, v, and provide a means of dealing with the integral term. We begin by setting \(v_m = m\Delta v\), where \(m = 0, 1, 2, \ldots, M\). Typically we will set the maximum variance to be \(v_M = 100\%\).

Furthermore, we discretise the time to expiry according to \(\tau_n = n\Delta \tau\), where \(\tau_N = T\). We denote the option price along the variance line \(v_m\) and time line \(\tau_n\) by \(C(S, v_m, \tau_n) \equiv C^m_n(S)\), and set

\[
V(S, v_m, \tau_n) = \frac{\partial C(S, v_m, \tau_n)}{\partial S} \equiv V^m_n(S),
\]

which is of course the option delta at the grid point.

We now select finite difference approximations for the derivative terms with respect to v. For the second order term, at the grid point \((S, v_m, \tau_n)\) we use the standard central difference scheme

\[
\frac{\partial^2 C}{\partial v^2} = \frac{C^m_{n+1} - 2C^m_n + C^m_{n-1}}{(\Delta v)^2},
\]
and for the cross-derivative term at the grid point \((S, v, \tau_n)\) we use the central difference approximation
\[
\frac{\partial^2 C}{\partial S \partial v} = \frac{V_{m+1}^n - V_{m-1}^n}{2\Delta v}.
\] (3.3)

Since the coefficients of the second order derivative terms go to zero as \(v \to 0\), we use an upwinding finite difference scheme (see [16, Chapter 8]) for the first order derivative term, such that, at the grid point \((S, v, \tau_n)\) we have
\[
\frac{\partial C}{\partial v} = \begin{cases} 
\frac{C_{m+1}^n - C_m^n}{\Delta v} & \text{if } v \leq \frac{\alpha}{\beta}, \\
\frac{C_m^n - C_{m-1}^n}{\Delta v} & \text{if } v > \frac{\alpha}{\beta}.
\end{cases}
\] (3.4)

Since the second order derivative terms both vanish as \(v \to 0\), upwinding helps to stabilise the finite difference scheme with respect to \(v\).

The integral term in (2.10) at each grid point is estimated using numerical integration. We assume that the jump sizes are lognormally distributed, with
\[
G^*(Y) = \frac{1}{Y \delta \sqrt{2\pi}} \exp \left\{ -\frac{\left[ \ln Y - \left( \gamma - \frac{\delta^2}{2} \right) \right]^2}{2\delta^2} \right\}.
\] (3.5)

Lognormal jump sizes were considered by Merton [25] and remain a popular choice for jump-size distributions. However the method developed here could allow a wide range of choices for \(G^*(Y)\), for instance the double exponential distribution of Kou [22]. Making the change of variable \(X = [\ln Y - (\gamma - \delta^2/2)]/\sqrt{2}\delta\), the integral term in (2.10) becomes
\[
I(S, v, \tau) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-X^2} C(S \exp\{(\gamma - \delta^2/2 + \sqrt{2}\delta X^H\}, v, \tau) dX.
\] (3.6)

We approximate (3.6) at the grid point \((v_m, \tau_n)\) with the discretisation obtained by applying the Hermite Gauss-quadrature scheme, namely
\[
I_m^H = \frac{1}{\sqrt{2\pi}} \sum_{j=0}^{J} w_j^H C_m^n(S \exp\{(\gamma - \delta^2/2 + \sqrt{2}\delta X_j^H)\}),
\] (3.7)

where \(w_j^H\) and \(X_j^H\) are the weights and abscissas for the Hermite Gauss-quadrature scheme with \(J\) integration points. Computations of the weights and abscissas for this scheme are given by Abramowitz and Stegun [1]. We interpolate for the required non grid point values of \(C_m^n\) using cubic splines fitted in \(S\) along the line at the grid point \((v_m, \tau_n)\).

Next we must select a discretisation for the time derivative. Initially we use a standard backward difference scheme, given at the grid point \((S, v, \tau)\) by
\[
\frac{\partial C}{\partial \tau} = \frac{C_m^n - C_{m-1}^n}{\Delta \tau}.
\] (3.8)
This approximation is only first order accurate with respect to time. For the case of the standard American put option, Meyer and van der Hoek [27] demonstrate that the accuracy of the method of lines increases considerably by using a second order approximation for the time derivative, specifically

\[
\frac{\partial C}{\partial \tau} = \frac{3}{2} \frac{C^n_m - C^{n-1}_m}{\Delta \tau} - \frac{1}{2} \frac{C^{n-1}_m - C^{n-2}_m}{\Delta \tau}.
\] (3.9)

Thus we initiate the method of lines solution by using (3.8) for the first several time steps, and then switching to (3.9) for all subsequent time steps.

Applying (3.2)–(3.9) to the IPDE (2.10), we must now solve a system of second order IDEs at each time step and variance grid point. For the first few time steps, the IDE at the grid point \( v = v_m \) and \( \tau = \tau_n \) is

\[
\frac{v_m S^2}{2} \frac{d^2 C^n_m}{dS^2} + \rho \sigma v_m S \frac{V^n_{m+1} - V^n_{m-1}}{2 \Delta S} + \frac{\sigma^2 v_m}{2} \frac{C^n_{m+1} - 2C^n_m + C^n_{m-1}}{(\Delta v)^2} + \frac{\alpha - \beta v}{2} \frac{C^n_{m+1} - C^n_{m-1}}{\Delta v} + \frac{\alpha - \beta v}{2} \frac{C^n_{m+1} - 2C^n_m + C^n_{m-1}}{\Delta v} + (r - q - \lambda^* k^*)S \frac{dC^n_m}{dS} = \frac{(r + \lambda^*)C^n_m}{\Delta \tau} + \lambda^* P^n_m - \frac{C^n_m - C^{n-1}_m}{\Delta \tau} = 0,
\] (3.10)

and for all subsequent time steps the IDE is

\[
\frac{v_m S^2}{2} \frac{d^2 C^n_m}{dS^2} + \rho \sigma v_m S \frac{V^n_{m+1} - V^n_{m-1}}{2 \Delta S} + \frac{\sigma^2 v_m}{2} \frac{C^n_{m+1} - 2C^n_m + C^n_{m-1}}{(\Delta v)^2} + \frac{\alpha - \beta v}{2} \frac{C^n_{m+1} - C^n_{m-1}}{\Delta v} + \frac{\alpha - \beta v}{2} \frac{C^n_{m+1} - 2C^n_m + C^n_{m-1}}{\Delta v} + (r - q - \lambda^* k^*)S \frac{dC^n_m}{dS} = \frac{(r + \lambda^*)C^n_m}{\Delta \tau} + \lambda^* P^n_m - \frac{C^n_m - C^{n-1}_m}{\Delta \tau} = 0.
\] (3.11)

We require two boundary conditions in the \( v \) direction, one at \( v_0 \) and the other at \( v_M \). For large values of \( v \), the rate of change of the option price with respect to \( v \) diminishes. So for sufficiently large values of \( v \), one can treat this rate of change as zero without any impact on the accuracy of the solution at other values of \( v \).

Thus we set \( \partial C/\partial v = 0 \) along the variance boundary \( v = v_M \). When \( v \) is zero, we fit a quadratic polynomial through the option prices at \( v_1 \), \( v_2 \) and \( v_3 \), and then use this to extrapolate an approximation of the price at \( v_0 \). This provides us with a satisfactory estimate of the price along \( v_0 \) for the purpose of generating a stable solution for small values of \( v \).

After taking the boundary conditions into consideration, at each time step \( n \) we must solve a system of \( M - 1 \) second order IDEs along the variance lines. This is done using a two stage iterative scheme. First, we treat the IDEs as ODEs by using \( C^{n-1}_m \) as an initial approximation for \( C^n_m \) in the integral term \( P^n_m \). We then solve
the ODEs for increasing values of $v$, using the latest available estimates for $C_{m+1}^n$, $C_{m-1}^n$, $V_{m+1}^n$ and $V_{m-1}^n$. The initial estimates for $C_m^n$ and $V_m^n$ are simply $C_{m-1}^n$ and $V_{m-1}^n$. Otherwise we use the latest estimates for $C_m^n$ and $V_m^n$ found during the current iteration through the variance lines. We iterate until the price profile converges to a desired level of accuracy. Second, once the price has converged, we update the estimate of the integral term $I_m^n$ using the current price profile estimate, and repeat the process until convergence is obtained for both levels of iteration. We then proceed to the next time step.

The generic first order form for (3.10) and (3.11) is

$$\frac{dC_m^n}{dS} = V_m^n, \quad (3.12)$$

$$\frac{dV_m^n}{dS} = A_m(S)C_m^n + B_m(S)V_m^n + P_m^n(S), \quad (3.13)$$

where $P_m^n(S)$ is a function of $C_{m+1}^n$, $C_{m-1}^n$, $V_{m+1}^n$, $V_{m-1}^n$, $C_m^{n-1}$, $C_m^{n-2}$ and $I_m^n$. We solve (3.12)–(3.13) using the Riccati transform, full details of which are provided by Meyer and van der Hoek [27].\(^3\) Note that we are only able to apply the Riccati transform to the system (3.12)–(3.13) provided that both equations are treated as ODEs. This is made possible by approximating $I_m^n$ using the values $C_{m-1}^n$ of the previous time step, as stated earlier, and then using an iterative technique in which the integral term is updated until the price converges, as in Meyer [26]. For later use we note that the right hand sides of Eqs. (3.12) and (3.13) respectively yield the delta and gamma at each point of the grid.

The Riccati transformation is given by

$$C_m^n(S) = R_m(S)V_m^n(S) + W_m^n(S), \quad (3.14)$$

where $R$ and $W$ are solutions to the initial value problems

$$\frac{dR_m}{dS} = 1 - B_m(S)R_m(S) - A_m(S)(R_m(S))^2, \quad R_m(0) = 0, \quad (3.15)$$

$$\frac{dW_m^n}{dS} = -A_m(S)R_m(S)W_m^n - R_m(S)P_m^n(S), \quad W_m^n(0) = 0, \quad (3.16)$$

and $V$ is the solution to

$$\frac{dV_m^n}{dS} = A_m(S)(R_m(S)V_m^n(S) + W_m^n(S)) + B_m(S)V_m^n + P_m^n(S), \quad V_m^n(b_m^n) = 1, \quad (3.17)$$

where we denote the free boundary at grid point $(v_m, \tau_m)$ by $b(v_m, \tau_m) = b_m^n$.\(^4\) Since $R_m$ is independent of $\tau$, we begin by solving (3.15) and storing the solution. Next we solve (3.16) for increasing values of $S$, ranging from $0 < S < S_{\text{max}}$, where we select $S_{\text{max}}$ sufficiently large such that $S_{\text{max}} > b_m^n$ will be guaranteed. We then step

\(^3\)The Riccati transform basically replaces a given differential system [here (3.12) and (3.13)] with an equivalent set of uncoupled equations of lower dimension [here (3.15), (3.16) and (3.17)].

\(^4\)All ODEs have been solved by use of the implicit trapezoidal rule (see [31]).
forward in $S$ using the generated values of $R_m$ and $W_m^n$ until we encounter the value $S^*$ such that

\[ S^* - K = R_m(S^*) + W_m^n(S^*), \tag{3.18} \]

and thus $S^*$ is the value of the free boundary at grid point $(v_m, \tau_n)^6$. Once $b_m^n$ has been determined we then solve (3.17) starting at $S = b_m^n$ and sweeping back to $S = 0$. Finally we use the calculated values of $R_m, W_m^n$ and $V_m^n$ in (3.14) to determine the option price at each grid point along the variance lines at time to maturity $\tau_n$.

In Fig. 2 we illustrate one sweep through the grid points on the $v - \tau$ plane. In Fig. 3 we show the stencil for the typical grid point in Fig. 2; this essentially shows the grid point values of $C$ that enter the right-hand side of (3.13). Figure 4 then illustrates the solution of (3.18) along a line in the $S$ direction from a typical grid point in the $v - \tau$ plane.

4. Numerical Solution Using the Componentwise Splitting Method

Ikonen and Toivanen [21] have proposed a non-standard finite difference method that involves discretisation of the spot price on nonuniform grids in order to solve the American option pricing problems under stochastic volatility. We employ the

---

5 We test Eq. (3.18) at each grid point and find the grid points between which $S - K - R_m(S) - W_m^n(S)$ changes sign. We then use Newton’s method to search for the root of $S^*$ of the cubic interpolant through the four nearest points around of this point.

6 We remind the reader that at $S^*$ the first of the free boundary conditions (2.9) becomes $V_m^n(S^*) = 1$. 

---
Fig. 3. Stencil for the typical grid point of Fig. 2. The stencil for \( C_m^n \) depends on \((C_{m-1}^n, C_m^n, C_{m+1}^n, C_{m-2}^n)\).

Fig. 4. Solving for the free boundary point along a \((v_m, \tau_n)\) line.

The same method and with the only difference being that we extend it to handle the integral over jumps term. Here, we give the basic details of the componentwise splitting method in order to make clear how the jump term is incorporated. We refer the reader to Ikonen and Toivanen [21] (henceforth IT) for full details.

For a reference grid point \((S, v)\), we have neighbouring grid points at \((S - h_l, v, \tau)\) and at \((S + h_r, v, \tau)\), where \(h_l\) and \(h_r\) are the local left and right grid step sizes for the spot price, respectively. The choice of these step sizes is discussed later. In the volatility direction we employ a uniform grid with the step size \(h\) and, hence, neighboring grid points are at \((S, v - h, \tau)\) and at \((S, v + h, \tau)\).

By use of the Taylor expansions at \((S - h_l, v, \tau)\) and \((S + h_r, v, \tau)\), the first-order and second-order derivatives in the \(S\)-direction are given by the finite difference
approximations
\[
\frac{\partial C}{\partial S} \approx \frac{1}{h_l + h_r} \left( \frac{h_l}{h_r} C(S + h_r, v, \tau) - \left( \frac{h_l}{h_r} - \frac{1}{h_l} \right) C - \frac{h_r}{h_l} C(S - h_l, v, \tau) \right),
\]
(4.1)
and
\[
\frac{\partial^2 C}{\partial S^2} \approx \frac{2}{h_l + h_r} \left( \frac{1}{h_l} C(S - h_l, v, \tau) - \left( \frac{1}{h_l} + \frac{1}{h_r} \right) C + \frac{1}{h_r} C(S + h_r, v, \tau) \right).
\]
(4.2)

The first-order and second-order derivatives in the \( v \)-direction are approximated in the usual way with a central difference.

IT discuss the second-order cross derivative terms when the correlation is positive, however we wish to handle both the positive and negative correlation cases. Hence, here we detail the negative correlation case and refer the reader to IT for the positive correlation case. By considering Taylor expansions at \((S + h_r, v - h, \tau)\) and \((S - h_l, v + h, \tau)\) IT obtain for the second-order cross derivative terms the approximations
\[
\frac{\partial^2 C}{\partial S \partial v} \approx \frac{1}{h_r h} \left( -C(S + h_r, v - h, \tau) + C + h_r \frac{\partial C}{\partial S} \right.
\]
\[
- h \frac{\partial C}{\partial v} + \frac{1}{2} h_r^2 \frac{\partial^2 C}{\partial S^2} + \frac{1}{2} h_r^2 \frac{\partial^2 C}{\partial v^2} \left. \right) ,
\]
(4.3)
and
\[
\frac{\partial^2 C}{\partial S \partial v} \approx \frac{1}{h_l h} \left( C(S - h_l, v + h, \tau) + C - h_l \frac{\partial C}{\partial S} \right.
\]
\[
+ h_l \frac{\partial C}{\partial v} + \frac{1}{2} h_l^2 \frac{\partial^2 C}{\partial S^2} + \frac{1}{2} h_l^2 \frac{\partial^2 C}{\partial v^2} \left. \right) ,
\]
(4.4)

which involve derivatives only in the \( S \)- and \( v \)-directions. In order to obtain a discretisation with the desired so called \( M \)-Matrix properties (that is, positive diagonal elements and negative sub-diagonal and super-diagonal elements) IT suggest the convex combination
\[
\frac{\partial^2 C}{\partial S \partial v} \approx \frac{w}{h_l h} \left( -C(S - h_l, v + h, \tau) + C - h_l \frac{\partial C}{\partial S} + h_l \frac{\partial C}{\partial v} + \frac{1}{2} h_l^2 \frac{\partial^2 C}{\partial S^2} + \frac{1}{2} h_l^2 \frac{\partial^2 C}{\partial v^2} \right)
\]
\[
+ \frac{1 - w}{h_r h} \left( -C(S + h_r, v - h, \tau) + C + h_r \frac{\partial C}{\partial S} \right.
\]
\[
- h_r \frac{\partial C}{\partial v} + \frac{1}{2} h_r^2 \frac{\partial^2 C}{\partial S^2} + \frac{1}{2} h_r^2 \frac{\partial^2 C}{\partial v^2} \left. \right) \) \)
(4.5)
of (4.3) and (4.4) for the second-order cross derivative. The weighting parameter \( w \) has a value between zero and one. For most of the grid points we use the weight \( w = 0.5 \). On the boundaries\(^7\) \([0, S_L] \times \{v_M\} \) where the boundary condition (2.6) is posed

\(^7\)Note that \( L \) and \( M \) are the number of \( S \) steps and the number of \( v \) steps respectively whilst \( N \) is the number of \( \tau \) steps.
we use $w = 0$ to avoid crossing the boundary in the $v$ direction. On the boundaries \( \{ S_L \} \times [0, v_M] \) where the boundary condition (2.8) is posed we use the weight $w = 1$ to avoid crossing the boundary in the $S$ direction. For small $S$ the weight is set to zero to obtain the $M$-Matrix properties. We refer the reader to IT for a discussion analogous to the above for the positive correlation case.

Using the approximation (4.5) for the cross-derivative in the partial differential equation (2.10), we obtain a form which contains only the partial derivatives in the $S$ direction and in the $v$ direction. This intermediate form reads

\[
\frac{\partial C}{\partial \tau} - \left[ \frac{1}{2} v S^2 + w \rho \sigma v S \frac{h_l}{2h} + (1-w) \rho \sigma v S \frac{h_r}{2h} \right] \frac{\partial^2 C}{\partial S^2} \\
- \left[ \frac{1}{2} \sigma^2 v + w \rho \sigma v S \frac{h_l}{2h} + (1-w) \rho \sigma v S \frac{h_r}{2h} \right] \frac{\partial^2 C}{\partial v^2} \\
- \left[ rS - w \rho \sigma v S \frac{1}{h_l} + (1-w) \rho \sigma v S \frac{1}{h_r} \right] \frac{\partial C}{\partial S} \\
- \left[ \kappa (\theta - v) - w \rho \sigma v S \frac{1}{h_l} + (1-w) \rho \sigma v S \frac{1}{h_r} \right] \frac{\partial C}{\partial v} \\
- \left[ -r + w \rho \sigma v S \frac{1}{h_l h} + (1-w) \rho \sigma v S \frac{1}{h_r h} \right] C \\
+ w \rho \sigma v S \frac{1}{h_l h} C(S - h_l, v + h, \tau) + (1-w) \rho \sigma v S \frac{1}{h_r h} C(S + h_r, v - h, \tau) \\
- I(S, v, \tau) = 0. \tag{4.6}
\]

where the last item $I(S, v, \tau)$ is the jump term from (3.6) which is approximated as in (3.7) in Sec. 3.

A seven point finite difference stencil around the typical $(S, v, \tau)$ point, which is displayed in Fig. 5, is obtained by using the finite difference approximations introduced in the previous part of this section for the spatial derivatives appearing in (4.6). We will discretise and solve (4.6) without jumps first and then we will add the jump term back later. The spatial discretisation leads to a semi-discrete equation which has the matrix representation

\[
\frac{\partial C}{\partial \tau} + \mathbf{A} C = 0 \tag{4.7}
\]

where $\mathbf{A}$ is a block tridiagonal $(L+1)(M+1) \times (L+1)(M+1)$ matrix and $C$ is a vector of length $(L+1)(M+1)$.

Next, we implement the Crank-Nicolson method to discretise the semi-discrete problem (4.7) as

\[
\left( I + \frac{1}{2} \Delta \tau \mathbf{A} \right) C^{(k+1)} = \left( I - \frac{1}{2} \Delta \tau \mathbf{A} \right) C^{(k)} \quad k = 0, \ldots, N - 1, \tag{4.8}
\]

where $N$ is the number of time steps and $I$ is the identity matrix.
After the discretisation of the underlying IPDE with two spatial variables an approximate price of an American option can be obtained by solving a sequence of linear complementarity problems (LCPs) (see [34])

\[
\begin{aligned}
& BC^{(k+1)} \geq DC^{(k)}, \quad C^{(k+1)} \geq c, \\
& (BC^{(k+1)} - DC^{(k)})^T (C^{(k+1)} - c) = 0,
\end{aligned}
\]  

\hspace{1cm} (4.9)

for \( k = 0, \ldots, N - 1 \). The matrices \( B \) and \( D \) in (4.9) are defined by (4.8) for the Crank-Nicolson method. The initial value \( C^{(0)} \) is given by the discrete form \( c \) of the payoff function \( c \) of the option, so that the \( i \)th element of \( C^{(0)} \) is given by

\[
C^{(0)}_i = \max(S_i - K, 0).
\]  

\hspace{1cm} (4.10)

In order to solve the sequence (4.9) of LCPs more efficiently, we implement the componentwise splitting method for LCPs based on the decomposition of the matrix \( A \) in (4.7) according to

\[
A = A_S + A_{Sv} + A_v.
\]  

\hspace{1cm} (4.11)

The matrices \( A_S, A_{Sv}, A_v \) contain the couplings of the finite difference stencil in the \( S \)-direction, in the \( Sv \)-direction, and in the \( v \)-direction, respectively. We refer the reader to IT for the full details of this decomposition.

We follow IT in using a second-order accurate splitting method by performing a Strang symmetrisation (see [32]) for the splitting method which uses the Crank-Nicolson method. We choose to perform first a half time step with \( A_S \) and then with \( A_v \), a full time step with \( A_{Sv} \), and finally a half time step with \( A_v \) and then
with \( A_S \). The notations used are
\[
B_{S/2} = I + \frac{1}{4} \Delta \tau A_S, \quad B_{v/2} = I + \frac{1}{4} \Delta \tau A_v,
\]
\[
B_{Sv} = I + \frac{1}{2} \Delta \tau A_{Sv}, \quad D_{Sv} = I - \frac{1}{2} \Delta \tau A_{Sv},
\]
\[
D_{S/2} = I - \frac{1}{4} \Delta \tau A_S, \quad D_{v/2} = I - \frac{1}{4} \Delta \tau A_v,
\]
in terms of which, the original LCP (4.9) is approximated by following five LCPs in successive directions:
\[
B_x C^{(j+\frac{1}{5})} \geq D_x C^{(j)}, \quad C^{(j+\frac{1}{5})} \geq c, \quad (4.13)
\]
\[
(B_x C^{(j+\frac{1}{5})} - D_x C^{(j)})^T (C^{(j+\frac{1}{5})} - c) = 0, \quad (4.14)
\]
where \( j \) and \( x \) run successively over the values
\[
j = \left( k, k + \frac{1}{5}, k + \frac{2}{5}, k + \frac{3}{5}, k + \frac{4}{5} \right),
\]
and
\[
x = \frac{S}{2}, \frac{v}{2}, Sv, \frac{v}{2}, \frac{S}{2},
\]
while \( k \) runs over the values \( k = 0, 1, 2, \ldots, N - 1 \).

We will omit the details of the implementations\(^8\) of the above system but again refer the reader to IT.

To be consistent with the Strang symmetrisation given by (4.13) and (4.14), the evaluation of the integral term should also be decomposed and distributed amongst the five partial time steps. In fact, we add the integral term \( I(S, v, \tau) \) to the explicit terms\(^9\) of (4.13) separately and it is evaluated in a similar way as it is in Sec. 3 by using a cubic spline interpolation on the stock prices from the previous time step. Through iterations at each time step the integral jump term converges. More precisely the integral jump term is handled in the following way: first, we evaluate \( I(S, v, \tau) \) at payoff from the previous time step or the result from the latest iteration; then we add 1/6 of \( I(S, v, \tau) \) to the \( S \) direction which is the explicit term of LCPs (4.13) for \( j = k \) and \( x = S/2 \); we add another 1/6 of \( I(S, v, \tau) \) to the \( v \) direction which is the explicit term of LCPs (4.13) for \( j = k + 1/5 \) and \( x = v/2 \); next we add 1/3 of \( I(S, v, \tau) \) to the \( Sv \) direction which is the explicit term of LCPs (4.13) for \( j = k + 2/5 \) and \( x = Sv \); and we add 1/6 of \( I(S, v, \tau) \) to the \( v \) direction which is the explicit term of LCPs (4.13) for \( j = k + 3/5 \) and \( x = v/2 \); finally we add 1/6 of \( I(S, v, \tau) \) to the \( S \) direction which is the explicit term of LCPs (4.13) for \( j = k + 4/5 \)

\(^8\)In our implementation the boundary condition at \( v = 0 \) is handled in the same way as those described in Sec. 3 which is different from IT, as has previously been discussed in Sec. 2.

\(^9\)By the explicit term we mean the term on the right hand side of each of the inequalities (4.13), which are at the most recent time iteration step.
and \( x = S/2 \). Now we have added a full time step integral to the original LCP (4.9). Next, we evaluate the integral term with cubic spline interpolation based on the new price and calculate the average difference of the successive iterations in the grid. We continue to add the integral terms to LCPs and to calculate the new price based on the new integral if the difference is larger than some tolerance, otherwise we will proceed to the next time step.

5. Numerical Results

To demonstrate the performance of the method of lines algorithm outlined in Sec. 3 we implement the method for a given set of parameter values, chosen in order to best illustrate the impact that stochastic volatility and jump-diffusion may have on the early exercise boundary for an American call option. The parameter values used are listed in Table 1.

We consider the case where \( r < q \), and a time to maturity of 6 months, as this best demonstrates the changes that arise in the free boundary when jumps are introduced. The value of \( \sigma \) is chosen intentionally large in order to emphasize the impact of stochastic volatility on the free boundary. We assume that jump sizes are log-normally distributed about a mean value of \( Y = 1 \) so that in a sense the jumps up and down average out. This allows us to focus on the impact that the Wiener correlation, \( \rho \), has on the free boundary. In addition, the small value of \( \delta \) has been chosen so as to avoid further increases to the overall variance of \( S \).

When implementing the method of lines we take the following case as an example to show its convergence pattern. We use \( N = 50 \) time-steps and \( M = 100 \) variance lines, with maximum variance \( v_M = 100\% \). We take a non-uniform grid in \( S \), splitting the domain into three intervals. Given that the strike price is \( K = 100 \), the maximum value for \( S \) is set to 400, with a total of 1138 grid points (denoted by \( S_{pts} \)), distributed between the three intervals such that there are 40 points for \( 0.5 \leq S \leq 1 \), there are 198 points for \( 1 \leq S \leq 100 \), and finally 900 grid points for \( 100 \leq S \leq 400 \).

For the Hermite Gauss-quadrature scheme in (3.7) we use \( J = 50 \) abscissa points. All iterative calculations utilise the stopping condition that the maximum over all \( S \) of \( |C_m^n(S)^{(i)} - C_m^n(S)^{(i-1)}| \) is less than \( 10^{-8} \), where the subscript \( i \) denotes the solution at the \( i \)th iteration. We note that for the parameter values given in Table 1, Table 1. Parameter values used for the American call option. The stochastic volatility (SV) parameters correspond to the Heston model. The jump-diffusion (JD) parameters correspond to the Merton model with log-normal jump sizes.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>SV parameter</th>
<th>Value</th>
<th>JD parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( T )</td>
<td>0.50</td>
<td>( \theta )</td>
<td>0.04</td>
<td>( \lambda^* )</td>
<td>5.00</td>
</tr>
<tr>
<td>( r )</td>
<td>0.03</td>
<td>( \kappa_v )</td>
<td>2.00</td>
<td>( \gamma )</td>
<td>0.00</td>
</tr>
<tr>
<td>( q )</td>
<td>0.05</td>
<td>( \sigma )</td>
<td>0.40</td>
<td>( \delta )</td>
<td>0.10</td>
</tr>
<tr>
<td>( K )</td>
<td>100</td>
<td>( \lambda_v )</td>
<td>0.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \rho )</td>
<td></td>
<td></td>
<td></td>
<td>( \pm 0.50 )</td>
<td></td>
</tr>
</tbody>
</table>
the solution along the variance lines typically converges for less than 85 iterations, and convergence with respect to updating the integral term (3.6) generally needs no more than 6 iterations. Furthermore, the number of iterations along the variance lines reduce by more than a quarter after the first time the integral term is updated and continues to reduce by more than a half each time the integral term is updated. A typical sequence is provided in Table 2. It is also of interest to observe how the total number of variance iterations changes as the number of variance lines changes; this effect is shown in Table 3. It seems that the number of variance iterations increases faster than the number of lines in the $v$ direction. This deterioration in convergence is similar to that observed in a standard Gauss-Seidel scheme, and is probably due to the fact that as in Gauss-Seidel iteration we are using the latest updates for the values as we step through the $(v, \tau)$ grid. It is also worth noting that of all the components within the iterative scheme, computing the integral term (3.7) is the most computationally intensive, since we must perform $J$ extrapolations of $C(S, v, \tau)$ with respect to $S$ at every point in the $S-v$ grid. Thus we are required to fit a total of $M$ cubic splines at each iteration with respect to the integral term, and each spline is fitted using every grid point in $S$.

A sample early exercise surface is provided in Fig. 6, generated using the method of lines for the case where $\rho = -0.5$. The value of the free boundary at expiry is independent of $v$. The free surface, $b(v, \tau)$, is an increasing function of $v$, and along a given value of $v$ we observe an early exercise boundary of the form typically found for American call options. It is worth noting that the free surface generated by

<table>
<thead>
<tr>
<th>Integral iteration</th>
<th>Variance iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st</td>
<td>82</td>
</tr>
<tr>
<td>2nd</td>
<td>61</td>
</tr>
<tr>
<td>3rd</td>
<td>39</td>
</tr>
<tr>
<td>4th</td>
<td>18</td>
</tr>
<tr>
<td>5th</td>
<td>3</td>
</tr>
<tr>
<td>6th</td>
<td>1</td>
</tr>
<tr>
<td>Total SV iterations</td>
<td>204</td>
</tr>
</tbody>
</table>

Table 2. Sample convergence pattern for the method of lines iterative procedures. Parameter values are as given in Table 1.

<table>
<thead>
<tr>
<th>No. of points in the $v$ direction</th>
<th>No. of total variance iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>41</td>
</tr>
<tr>
<td>50</td>
<td>75</td>
</tr>
<tr>
<td>60</td>
<td>96</td>
</tr>
<tr>
<td>90</td>
<td>174</td>
</tr>
<tr>
<td>100</td>
<td>204</td>
</tr>
<tr>
<td>120</td>
<td>274</td>
</tr>
</tbody>
</table>

Table 3. The changes of the number of required “volatility iterations” with respect to the changes of the number of points in the $v$ direction.
the method of lines is smooth, even when jumps are present, a feature not often displayed in the free boundary estimates generated using finite difference methods, such as Ikonen and Toivanen [20].

We contrast the relative pricing accuracy of both the method of lines (MOL) and component wise splitting (CS) methods. Using the parameter values in Table 1, and setting the spot variance to \( v = 0.04 \), we generate American call option prices for \( S \) values of 80, 90, 100, 110 and 120 using the three numerical methods discussed above; the method of lines as outlined in Sec. 3, the componentwise splitting approach which is detailed in Sec. 4, and a Crank-Nicolson scheme where the system of difference equations is solved using PSOR. The integral term is approximated in the same manner as for the MOL and CS, and we iterate at each time step, updating the integral term until the price profile converges.\(^{10}\) We use \( J = 50 \) abscissa points for the integral term in all methods. We experimented with each of the above methods with an increase in the number of abscissa points from 50 to 100, but this led to a significant increase in the run time with no significant change in the option prices.

For both finite difference methods, we find that it is more efficient to update the integral term external to solving the system of difference equations\(^{11}\). Note that

\(^{10}\)Specifically, the system of difference equations is solved using PSOR for a given estimate of the integral term. The system of difference equations is then solved again using an updated estimate for the integral term. This is repeated until the price profile converges.

\(^{11}\)For the componentwise splitting method, the five tridiagonal systems are solved sequentially for a given estimate of the integral term. The integral term is then updated, and the tridiagonal system solved again, repeating the procedure until the price profile converges.
while we do not prove convergence for these iterative schemes with respect to value
of the integral term, convergence is always observed in practice for the parameter
values under consideration. The source code for all methods was implemented using
NAG Fortran with the IMSL library running on the UTS, Faculty of Business F&E
HPC Linux Cluster which consists of 8 nodes running Red Hat Enterprise Linux 4.0
(64bit) with 2 × 3GHz 4MB Cache Xeon 5160 (dual core) Processors, 8GB 667MHz
DDR2-RAM.

In the following, we treat the solution for the price from PSOR with 1000 time
steps, 3000 variance steps and 6000 share price steps as the “true” solution for
the purpose of comparing the efficiency of both the MOL and CS methods. We
also provide runtimes for the different methods, and compute the root
mean square relative differences “RMSRDs” for each method in relation to the “true”
solution from PSOR. This allows us to make some further observations about the
relative performance of the other two methods. We find that increasing the number
of time steps for the MOL has little impact on the prices, while the price accuracy
improves more when the grid size in the variance and share price directions is refined.

\[ \text{RMSRD} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} \left( \frac{C(S_i) - \hat{C}(S_i)}{C(S_i)} \right)^2} \]

where \( S_i = 80 + 10 \cdot (i - 1) \), \( \hat{C}(S) \) is the estimate of the price, and \( C(S) \) is the true price. It is important to use RMSRD to measure the
ersors from price, delta and gamma together since they have quite different numerical scales.
Table 5. American call prices computed using method of lines (MOL), componentwise splitting (CS) and Crank-Nicolson with PSOR (PSOR). Parameter values are given in Table 1, with $\rho = -0.50$ and $v = 0.04$. The first number in brackets for the CS method indicates the ratio between the grid step sizes at $S_{\text{max}}$ and $K$ imposed on the the non-uniform grid in $S$. See Sec. 4.

<table>
<thead>
<tr>
<th>Method ($N, M, S_{\text{pts}}$)</th>
<th>$S$</th>
<th>RMSRD (%)</th>
<th>Runtime (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOL (50,100,1138)</td>
<td>80</td>
<td>1.1369</td>
<td>0.0578</td>
</tr>
<tr>
<td>MOL (200,100,1138)</td>
<td>90</td>
<td>3.3512</td>
<td>21.7156</td>
</tr>
<tr>
<td>MOL (200,250,2995)</td>
<td>100</td>
<td>7.5922</td>
<td>21.7168</td>
</tr>
<tr>
<td>CS (2.5) (200,100,294)</td>
<td>110</td>
<td>13.8786</td>
<td>21.7191</td>
</tr>
<tr>
<td>CS (2.5) (300,100,294)</td>
<td>120</td>
<td>21.7168</td>
<td>0.0404</td>
</tr>
<tr>
<td>CS (2.5) (300,200,549)</td>
<td>110</td>
<td>13.8798</td>
<td>21.6384</td>
</tr>
<tr>
<td>CS (2.5) (1000,1000,2764)</td>
<td>120</td>
<td>21.7162</td>
<td>0.0995</td>
</tr>
<tr>
<td>PSOR (200,200,300)</td>
<td>80</td>
<td>1.1233</td>
<td>0.0404</td>
</tr>
<tr>
<td>PSOR (500,500,1000)</td>
<td>90</td>
<td>3.3199</td>
<td>21.6384</td>
</tr>
<tr>
<td>PSOR (1000,3000,6000)</td>
<td>100</td>
<td>7.5440</td>
<td>21.6834</td>
</tr>
<tr>
<td></td>
<td>110</td>
<td>13.8309</td>
<td>21.6834</td>
</tr>
<tr>
<td></td>
<td>120</td>
<td>21.6834</td>
<td>0.7588</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>13.8734</td>
<td>21.7120</td>
</tr>
<tr>
<td></td>
<td>110</td>
<td>21.7162</td>
<td>0.0404</td>
</tr>
<tr>
<td></td>
<td>120</td>
<td>21.7120</td>
<td>0.0404</td>
</tr>
</tbody>
</table>

Clearly, the CS method runs faster when the grid is small compared with the MOL; with RMSRD as high as 0.0404% within 98 seconds it is almost 10 times faster than the MOL and produces results even more accurate than the MOL. However the accuracy deteriorates when the number of time steps is increased somewhat. It is also observed that the CS method produces prices with some oscillations especially in the negative correlation case.

If one wishes to achieve a given level of accuracy, we can see from the tables that the MOL seems the best method as it produces a higher accuracy with a RMSRD of 0.0177% within just half the time of CS, which produces results with a RMSRD of 0.0654%. Thus we are confident in asserting that the method of lines is very competitive for evaluating American options under stochastic volatility and jump-diffusion.

To see the overall efficiency of the three methods, we plot in Figs. 7–10 below the comparisons of the average accuracy of the American call price, delta and gamma with the MOL, the CS method and PSOR. Note that these figures are plotted up to the point on the horizontal axis where the “exact” solution curves start to become artificially steep due to the fact that these are the solutions taken to be the “exact” ones. The RMSRD on the vertical axis in each of Figs. 7 to 9 is calculated for the corresponding set of values, call prices, deltas or gammas, at the underlying prices $S = 80, 90, 100, 110$ and 120 and with both correlations $\rho = \pm 0.5$. In Fig. 10 we display the average of the sum of the above three RMSRDs. The runtime on the horizontal axis in all graphs is the computer time taken to produce all three quantities, namely the call prices, deltas and gammas.

A number of comments based on the figures and the calculations are warranted. First, it is clear from Fig. 10 that the method of lines (MOL) performs best in calculating, to similar accuracy, the call prices, deltas and gammas. For instance, it costs MOL around 1,000 seconds to achieve an overall relative accuracy of $10^{-3}$
Fig. 7. Runtime efficiency of American call prices with MOL, CS and PSOR. We take the price from PSOR with a large grid consisting of 1,000 time steps, 3,000 variance steps and 6,000 share price steps as the true solution. The root mean-square relative differences (RMSRDs) for each method in relation to this true solution correspond to the cases with share prices ranging from 80 to 120 and $\rho = \pm 0.5$.

Fig. 8. Runtime efficiency of American call deltas with MOL, CS and PSOR. We take the delta from MOL with a large grid consisting of 500 time steps, 1,000 variance steps and 11,380 share price steps as the true delta. The root mean-square relative differences (RMSRDs) for each method in relation to this true solution correspond to the cases with share prices ranging from 80 to 120 and $\rho = \pm 0.5$. 
Fig. 9. Runtime efficiency of gammas with MOL, CS and PSOR. We take the American call gammas from MOL with a large grid consisting of 500 time steps, 1,000 variance steps and 11,380 share price steps as the true gamma. The root mean-square relative differences (RMSRDs) for each method in relation to this true solution correspond to the cases with share prices ranging from 80 to 120 and $\rho = \pm 0.5$.

Fig. 10. Overall runtime efficiency with MOL, CS and PSOR. We take the solution from PSOR with a grid size of 1,000 time steps, 3,000 variance steps and 6,000 share price steps as a true solution for the price. We take the delta and gamma from the MOL with a large grid consisting of 500 time steps, 1,000 variance steps and 11,380 share price steps as the true delta and the true gamma. The root mean-square relative differences (RMSRDs) for each method in relation to this true solution correspond to the cases with share prices ranging from 80 to 120 and $\rho = \pm 0.5$. 
while it costs CS around 1,000,000 seconds to achieve the same overall accuracy and it takes PSOR even longer. Second, the results from the calculations (see Figs. 8 and 9) show that the American call deltas and gammas seem to have a faster convergence rate than the American call prices with the MOL. This is natural because for MOL, within each iteration, the call prices are obtained after working out the deltas and gammas to the same degree of accuracy. We also see from the calculations that the value of the deltas and gammas do not change up to 5 decimals when refining the grid size which is also evidence indicating that the deltas and gammas with the MOL are much closer to the “true” deltas and gammas than those for both CS and PSOR.

6. Impact of Stochastic Volatility and Jumps on the Free Surface and Prices

In this section, we explore the impact of stochastic volatility and jump-diffusion on the early exercise boundary and the price profile of an American call option. We consider four nested models for the underlying asset price: (i) the geometric Brownian motion (GBM) model of Black and Scholes [6] and Merton [24]; (ii) the jump-diffusion (JD) model of Merton [25]; (iii) the stochastic volatility (SV) model of Heston [18]; and (iv) the combined stochastic volatility and jump-diffusion (SVJD) model of Bates [5], given by (2.1)–(2.2). Here we aim to observe the impact that stochastic volatility and jump-diffusion have on the shape of the early exercise boundary, where the variance of $S$ is consistent for all four models.

Setting the spot variance to $v = 0.04$ (corresponding to a volatility — standard deviation — of 20%) in the SVJD model, we determine the time-averaged variance $s^2$ for ln $S$ over the life of the option by using the characteristic function for the marginal density of $x = \ln S$ given in Cheang et al. [9]. By requiring that $s^2$ be equal for all the models, we then determine the necessary parameter values for each model to ensure that they all have consistent variance over the time period of interest. A summary of the parameters for each model is presented in Table 6. For the jump-diffusion model, we use $\lambda^*$, $\gamma$ and $\delta$ as given in Table 1. For the Heston model, we vary the spot variance, $v_{SV}$, and long-run variance, $\theta_{SV}$, in order to match the variance. We determine suitable values for $v_{SV}$ and $\theta_{SV}$ by assuming that $v_{SV} = \theta_{SV}$, and then rounding $v_{SV}$ up to the nearest whole percent.

In Figs. 11 and 12 we plot the early exercise boundary for the American call option for the parameter values given in Tables 1 and 6. By matching the variance of $S$ over the life of the option for all four models (GBM, JD, SV, SVJD), we can discuss the impact of stochastic volatility and jump-diffusion on the American call option.

---

13 Note that $\theta_{SV}$ is the value of $\theta$ required to give the global variance in Table 5 when only SV dynamics are present.

14 This choice is made to avoid interpolating the solution with respect to $v$. Note that there are many combinations of $v$, $\theta$, $\kappa$, $\sigma$ and $\lambda_v$ in the SV model that can be used to match the variance of $S$ to that generated by the SVJD model.
Table 6. Parameters used to match the time-averaged variance for the GBM, JD, SV and SVJD models for a 6-month option. The global volatilities, \( s \), are 29.7860% for \( \rho = 0.50 \), and 30.2760% for \( \rho = -0.50 \). The value of \( v \) in the SVJD model is 4%.

<table>
<thead>
<tr>
<th>Model</th>
<th>Parameter</th>
<th>Value: ( \rho = 0.50 )</th>
<th>Value: ( \rho = -0.50 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>GBM</td>
<td>( v_{GBM} )</td>
<td>8.8721%</td>
<td>9.1664%</td>
</tr>
<tr>
<td>JD</td>
<td>( v_{JD} )</td>
<td>3.8596%</td>
<td>4.1539%</td>
</tr>
<tr>
<td>SV</td>
<td>( \theta_{SV} )</td>
<td>9.0000%</td>
<td>8.5250%</td>
</tr>
<tr>
<td></td>
<td>( v_{SV} )</td>
<td>10%</td>
<td>9%</td>
</tr>
</tbody>
</table>

Firstly, we note that the addition of jumps has a significant impact on the early exercise boundary. Near expiry the boundary is higher than for cases without jumps, and far from expiry the boundary is lower than the no-jump case. This holds true when adding jumps to both the GBM and SV models. In financial terms, near expiry the impact of a jump in \( S \) is likely to directly impact on the expiry value of the option, and thus the option holder is more cautious about exercising the call.

Fig. 11. Exploring the effect of jump-diffusion and stochastic volatility on the early exercise boundary for an American call option. The correlation is \( \rho = 0.50 \); all other parameter values are as listed in Tables 1 and 6.
Comparing the Early Exercise Boundary for various Models; $\rho = 0.5$

Fig. 12. Exploring the effect of jump-diffusion and stochastic volatility on the early exercise boundary for an American call option. The correlation is $\rho = -0.50$; all other parameter values are as listed in Tables 1 and 6.

for fear of incurring the rebalancing cost from downward jumps. For longer time periods, jumps in $S$ can be offset both by long-term diffusion and additional jumps in $S$ in the opposite direction. Thus the holder of the option is willing to exercise for smaller values of $S$.

For positive correlation, the addition of stochastic volatility consistently reduces the value of the early exercise boundary, relative to the nested model with constant volatility. The reverse is true when the correlation is negative. From Heston [18] we know that, in the European case, call options under stochastic volatility are worth less in-the-money than in the GBM case for positive correlation, and worth more in the same region when the correlation is negative. The free boundary behavior observed in Figs. 11 and 12 indicates that these option price changes persist in the case of American call options. The correlation coefficient affects the skewness of the density for $S$, and this is reflected in the option price as $\rho$ changes between positive and negative values.

We also note that the kurtosis of the density for $S$ is determined by the value of $\sigma$ (the “vol of the vol”), as demonstrated by Heston [18]. As $\sigma$ increases, the impact of stochastic volatility on the free boundary becomes more pronounced. As $\sigma$ tends towards zero we observe that the SV boundary converges to the GBM boundary,
with the same behaviour observed for SVJD and JD. We do not demonstrate this feature here, but note again that the large value of $\sigma$ has been chosen to demonstrate more clearly the impact of stochastic volatility on the early exercise boundary in Figs. 11 and 12.

Finally, we are interested in demonstrating the impact of stochastic volatility and jumps on the American call price, relative to the pure-diffusion case. In Fig. 13 we plot the price differences between the stochastic volatility (SV), stochastic volatility with jump diffusion (SVJD) and pure-diffusion American call prices when the correlation is negative (the case usually observed in practice). All other parameter values are the same as those used to generate the free boundaries in Figs. 11 and 12. Positive (negative) differences indicate that the SV or SVJD price is greater than (less than) the pure-diffusion price. When the call is at-the-money or out-of-the-money, the SV price is much lower than the pure-diffusion price. After adding the jumps the SVJD price is still lower than the pure-diffusion price but a bit higher than the SV price. The situation changes when the call goes in-the-money where the SV price starts to be higher than the pure-diffusion price but the SVJD price is lower than the SV price, while it is still higher than the pure-diffusion price.

In order to see how the parameter $\sigma$ affects the American call price under either stochastic volatility or stochastic volatility with jumps, we plot the price differences

Fig. 13. The effect of jump-diffusion and stochastic volatility on American call option prices. The correlation is $\rho = -0.50$; all other parameter values are as listed in Tables 1 and 6.
with different values of “volatility of the volatility”, $\sigma$, in Fig. 14 for $\rho = -0.5$. Here we see that the price profiles attain larger differences for the SV model, no doubt because the SV component is only partially contributing to the overall volatility in the SVJD case. A similar effect has been observed for $\rho = 0$ and $\rho = 0.5$.

In all of the above situations, both the SV and SVJD price will “converge” to the pure-diffusion price when the call is either very deep out-of-the-money or very deep in-the-money.

7. Conclusion

This paper studies the numerical evaluation of American call options under stochastic volatility and jump-diffusion. We propose a numerical method for solving the integro-partial differential equation that determines the option price by extending the method of lines algorithm to the case of stochastic volatility and jump-diffusion. Our approach generalises the method of lines solution of Meyer [26] to incorporate stochastic volatility. The method involves an approximation of the integral term using a Hermite Gauss-quadrature scheme, combined with cubic spline interpolation of the option price. The method iterates at each time step by updating the integral term until convergence is obtained. Although we do not provide any proof for the convergence of the iterations in this algorithm, we certainly observe convergence for all reported parameter values.
The accuracy of the method of lines (MOL) is assessed by comparing the American call prices with those obtained via two alternative schemes based on finite difference methods. The first of these is the standard Crank-Nicholson implicit scheme solved using projected successive over-relaxation (PSOR), and the second is based on the componentwise splitting finite difference scheme of Ikonen and Toivanen [21]. For both methods the integral term is approximated in the same manner as for the method of lines, explicitly estimating the integral and then iterating by updating the integral term until convergence of the price is obtained.

We have compared the three methods by using efficiency plots that show the run time needed to achieve a certain level of (relative) accuracy. We have used solutions from PSOR with a very high discretisation as the “true” solution for the price and a very high discretisation of the MOL for the “true” delta and gamma. We have done the comparisons for the calculations of prices, deltas and gammas separately and then jointly (that is the sum of the relative errors of all three at a certain level of accuracy). We find with all calculations that the MOL generally has the best performance, especially when one wants all three quantities to be within a given level of relative accuracy. We believe this is a consequence of the fact that the MOL calculates both the delta and gamma in the process of calculating the price, all to the same degree of accuracy.

Using the method of lines, we explore how the early exercise boundary of an American call option changes with respect to both jump-diffusion and stochastic volatility. Matching the variance over the life of the option, we compare the free boundaries generated by classical geometric Brownian motion, Merton’s [25] jump-diffusion, Heston’s [18] stochastic volatility, and the combined stochastic volatility and jump-diffusion model of Bates [5]. In all cases we assume that the jump sizes are log-normally distributed. We find that jumps have a pronounced effect on the free boundary, increasing its value close to expiry, but reducing it for longer time horizons. The addition of stochastic volatility increases the value of the boundary when correlation is negative, and decreases the value when correlation is positive. We have also examined how the price profiles change as we add stochastic volatility, then stochastic volatility plus jumps to the standard geometric Brownian motion situation. We focused in particular on the impact of the correlation between the stock price and volatility noise processes as well as on the volatility of the volatility.

A number of issues remain to be investigated in future research. First, the method proposed here does not need to be restricted to the Heston stochastic volatility process and log-normal jump size distributions, for instance it should be possible to apply the method to treat the double-exponential jump size distribution proposed by Kou [22]. Second, it would also be useful to extend the MOL (and also the CS method) to consider the case when the jumps are distributed according to more general processes such as Levy processes, since the pricing equation is still an integro-partial differential equation similar to (2.3) as can be seen in Cont and Tankov [12].
Appendix: The Boundary Condition at \( v = 0 \)

There seems to be no theory for degenerate IPDEs, but since the IPDE (2.10) will be time discretised and solved iteratively as a sequence of elliptic problems we shall ignore the integral term in our discussion of the boundary condition at \( v = 0 \). The coefficient matrix, that we shall denote \( A(S, v) \), associated with the second order derivatives of Eq. (2.10) is given by

\[
A(S, v) = \begin{pmatrix}
\frac{vs^2}{2} & \frac{\rho \sigma v S}{2} \\
\frac{\rho \sigma v S}{2} & \frac{\sigma^2 v}{2}
\end{pmatrix}.
\]

This matrix is positive semi-definite and singular for \( S = 0, v = 0 \) and \( |\rho| = 1 \).

The degeneracy on the boundary can be handled with the theory in Houston et al. [19]. The Fichera function associated with the boundary for this problem is given by

\[
b(S, v) = \left[ \left( r - q - \lambda^* k^* \right) S - \left( v S + \frac{\rho \sigma S}{2} \right) \right] n_1 + \left[ (\alpha - \beta v) - \left( \frac{\rho \sigma v}{2} + \frac{\sigma^2}{2} \right) \right] n_2
\]

where \( n = (n_1, n_2) = (1, 0) \) on \( S = 0 \) and \( n = (0, 1) \) on \( v = 0 \).

It follows that \( b(S, 0) = \alpha - \frac{\rho \sigma v}{2} \). If \( b(S, 0) \geq 0 \) then the boundary \( v = 0 \) is known as an outflow boundary where no boundary condition needs to be specified. If \( b(S, 0) < 0 \) then \( C(S, 0) \) must be specified.

The dependence of the boundary condition on the magnitude of \( \sigma \) does not make financial sense and is a direct consequence of the Heston model. For competing stochastic volatility models of the form

\[
dv = k_v(\theta - v)dt + \sigma v^{1+\epsilon}dz, \quad \epsilon > 0,
\]

the corresponding Fichera function will turn out to be

\[
b(S, 0) = \alpha > 0,
\]

so that \( v = 0 \) will always be an outflow boundary. An analogous problem arises in d’Halluin et al. [15] when pricing bonds with a CIR interest rate model. An \( \epsilon \)-modification of the interest rate model produced an outflow boundary, but small \( \epsilon \) had negligible influence on the numerical bond prices.

We also observe that \( b(0, v) = 0 \), so that \( S = 0 \) is also an outflow boundary. However, our numerical method will solve (2.10) for \( S \geq S_0 > 0 \) where \( A(S_0, v) \) is non-singular so that imposition of the boundary condition (2.7) is permitted.

The outflow boundary at \( v = 0 \) requires that the numerical scheme be a consistent approximation of the differential equation as \( v \to 0 \). This is readily shown.

Let \( L_1C_1 \) denote the left side of equation (3.10) or (3.11) for \( v_1 = \Delta v \). It requires \( C_0 \) and \( v_0 \). Now let \( C_{S0}^{E}(S) \) and \( C_{S0}^E(S) \) denote the quadratic extrapolant of \( C(S, v) \) and \( C_S(S, v) \) through the values at \( v_1, v_2 \) and \( v_3 \). It follows from
polynomial approximation that

$$|C(E|S) - C(S, 0)| \leq K \Delta v^3,$$  \hspace{1cm} (A.1)

and

$$|C(E|S) - C(S, 0)| \leq K \Delta v^3.$$  \hspace{1cm} (A.2)

Therefore

$$\lim_{\Delta v \to 0} \frac{C(E|S) - 2C(S, v_1) + C(S, v_2)}{\Delta v^2} = C_{vv}(S, v_1),$$

and similarly that

$$\lim_{\Delta v \to 0} \frac{C(S, v_1) - C(E|S)}{2\Delta v} = C_{Sv}(S, v_1).$$

This implies that

$$\lim_{v_1 \to 0} L_1 C(S, v_1) - LC(S, v_1) = 0$$

where $L$ is the IPD operator defined by Eq. (2.10). Hence our scheme is a consistent approximation of the IPDE as $v \to 0$. Note that $LC(S, 0)$ has hyperbolic character.

References


