A Penalty Method for American Options with Jump Diffusion Processes

Y. d'Halluin[*], P.A. Forsyth[†], and G. Labahn[‡]

March 9, 2003

Abstract

The fair price for an American option where the underlying asset follows a jump diffusion process can be formulated as a partial integral differential linear complementarity problem. We develop an implicit discretization method for pricing such American options. The jump diffusion correlation integral term is computed using an iterative method coupled with an FFT while the American constraint is imposed by using a penalty method. We derive sufficient conditions for global convergence of the discrete penalized equations at each timestep. Finally, we present numerical tests which illustrate such convergence.

Keywords: Jump diffusion, implicit discretization, American option
AMS Classification: 65M12, 65M60, 91B28

Acknowledgment: This work was supported by the Natural Sciences and Engineering Research Council of Canada, RBC Financial Group, and a subcontract with Cornell University, Theory & Simulation Science & Engineering Center, under contract 39221 from TG Information Network Co. Ltd.

1 Introduction

The pricing and hedging of derivative securities, also known as contingent claims, is a subject of much practical importance. One basic type of derivative is an *option*. The owner of a *call* option has the right but not the obligation to purchase an underlying asset (such as a stock) for a specified price (called the exercise price or strike price) on or before a specified expiry date. A *put* option is similar except the owner of such a contract has the right but not the obligation to sell. Options which can be exercised only on the expiry date are called *European*, whereas options which can be exercised any time up to and including the expiry date are classified as *American*.

The standard approach to valuation of derivatives begins with specifying a stochastic process for the underlying asset. Then, a suitably managed (usually dynamic) self-financing portfolio is constructed to minimize the risk for the holder of this portfolio. The initial cost of constructing this portfolio is then the fair value of the contingent claim. The management of the portfolio (hedging)

[*]School of Computer Science, University of Waterloo, Waterloo ON, Canada N2L 3G1 (e-mail: ydhallui@elora.uwaterloo.ca).

[†]School of Computer Science, University of Waterloo, Waterloo ON, Canada N2L 3G1 (e-mail: paforsyt@elora.uwaterloo.ca).

[‡]School of Computer Science, University of Waterloo, Waterloo ON, Canada N2L 3G1 (e-mail:glabahn@scg.uwaterloo.ca)

requires buying and selling various amounts of the underlying security. These hedging strategies require evaluation of the (mathematical) derivatives of the solution for the fair value.

It is common knowledge that the constant volatility Black-Scholes model is not consistent with market prices. In order to match observed market prices for options, traders use a matrix of implied volatilities [28], or generate a volatility surface [5]. However, as discussed in [2], volatility surfaces tend not to be very stable as a function of time. In particular, the surface obtained by matching today's prices, tends to become very flat as one looks out farther in time. This is a significant problem if this surface is used to price and hedge options which are very sensitive to the volatility in the future (forward start options for example).

A richer model which has attracted attention is based on the jump diffusion process, first suggested in [18]. Empirical studies of stock market behavior seem to indicate that geometric Brownian motion should be augmented by a discontinuous jump model using jumps based on a Poisson distribution, in order to reproduce the observed behavior [17]. For example, using a jump diffusion model, Andersen and Andreasen [2] fit S&P 500 option prices, and obtain excellent fits with stable parameters. If the same fitting exercise is attempted without the Poisson jumps, then the parameters are much less stable.

In the case of European options with jump diffusion, Anderson and Andreasen [2] use an operator splitting approach coupled with an FFT for evaluation of the jump integral term. This method is unconditionally stable, second order in time, and does not require solution of a dense matrix at each timestep.

The objective of this article is to develop a robust numerical method for pricing American options with jump diffusion. Most options traded on exchanges have the American early exercises feature, so being able to price these types of options under a jump diffusion model clearly is of practical importance.

Theoretical work on the properties of solutions to American option pricing problems under various assumptions for the jump process is reviewed in [21]. Approximation methods are discussed in [20]. However, most approximation methods require that the volatility surface be a function of time only. Numerical methods for American options with a finite number of jump states are described in [19]. A technique based on binomial lattices, which are essentially explicit finite difference methods, is discussed in [1]. Explicit methods, of course, suffer from timestep limitations due to stability considerations.

Previous work on numerical methods for American options under jump diffusion used an explicit timestepping method for the jump integral term, and used a standard linear complementarity solver to solve the algebraic linear complementarity problem at each timestep [32]. This method is only first order correct in time, and conditionally stable.

We are particularly interested in developing a method which can be used to price complex, path dependent options with American type constraints. Examples of these types of contingent claims include shout options [29], insurance guarantees [30], and convertible bonds [3]. It is also desirable that the method can be easily extended to handle true nonlinear effects, such as transaction costs and uncertain volatility [22].

In this paper we develop an implicit timestepping approach, which has the potential of second order accuracy in the time direction. Although there are various methods which can be used to efficiently solve linear complementarity problems [6] for one dimensional Partial Differential Equations (PDEs), we use a penalty method [12, 33] to enforce the American constraint. As discussed in [12], a penalty method can be easily extended to multifactor models [33], and to nonlinear models such as uncertain volatility and transaction costs [22]. It is a common misconception that penalty methods result in poorly conditioned algebraic problems. This is shown not to be the case in [12].

In this paper, we also give detailed convergence proofs of the convergence of the iteration for solution of the algebraic penalized equations at each timestep. These proofs require that the discretized PDE is an M-matrix. However, we have observed computationally that the method developed here also converges rapidly even if the discretized equations are not M-matrices. This is consistent with the observed convergence of the penalty method for American options under a stochastic volatility model [33].

The remainder of the paper is organized as follows. In the next section we give the mathematical model for options with jump diffusion processes given in terms of a partial integral differential equation. Section 3 gives a discretization for this equation in the case of European options. Section 4 extends the partial integral differential equation for use in the case of American options using the penalty method and shows that the resulting iteration is convergent. Section 5 gives some numerical examples. The paper ends with a conclusion and topics for future research.

2 Mathematical model

In this section we give the mathematical model for options with jump diffusion processes. We do this for both European and American options. Thus let S represent the underlying stock price. Then potential stock paths followed by the stock can be modeled by a stochastic differential equation given by

$$\frac{dS}{S} = \xi dt + \sigma dZ + (\eta - 1)dq \tag{2.1}$$

where

$$\begin{aligned}
\xi &\quad \text{is the drift rate,} \\
dq &\quad \text{is the independent Poisson process,} = \begin{cases} 0 & \text{with probability } 1 - \lambda dt \\ 1 & \text{with probability } \lambda dt, \end{cases} \\
\lambda &\quad \text{is the mean arrival time of the Poisson process,} \\
\eta - 1 &\quad \text{is an impulse function producing a jump from } S \text{ to } S\eta, \\
\sigma &\quad \text{is the volatility,} \\
dZ &\quad \text{is an increment of the standard Gauss-Wiener process.}
\end{aligned}$$

Let $V(S,t)$ be the value of a European contract that depends on the underlying stock price S and time t. The following backward Partial Integral Differential Equation (PIDE) for the value of $V(S,\tau)$ is found to be [2, 18, 28]

$$V_\tau = \frac{\sigma^2 S^2}{2} V_{SS} + (r - \lambda\kappa) S V_S - rV + \left(\lambda \int_0^\infty V(S\eta) g(\eta) d\eta - \lambda V \right), \tag{2.2}$$

where

$$\begin{aligned}
T &\quad \text{is the expiry/maturity date,} \\
r &\quad \text{is the risk free interest rate (assumed to be positive),} \\
\tau &= T - t, \text{where } t \text{ is the current time,} \\
g(\eta) &\quad \text{is the probability density function of the jump amplitude } \eta, \text{ thus} \\
&\quad \text{for all } \eta, g(\eta) \geq 0, \text{ and } \int_0^\infty g(\eta) d\eta = 1.
\end{aligned}$$

As a specific example, consider the probability density function suggested by [18, 26]:

$$g(\eta) = \frac{e^{\left(-\frac{(\log(\eta)-\mu)^2}{2\gamma^2}\right)}}{\sqrt{2\pi}\gamma\eta}. \tag{2.3}$$

If $E[\cdot]$ is the expectation operator,

$$E[\eta] \;=\; \int_0^\infty y g(y)\, dy \tag{2.4}$$

then, $E[\eta] = \exp(\mu + \gamma^2/2)$, which means that the expected relative change in the stock price is given by $\kappa = E[\eta - 1] = \exp(\mu + \gamma^2/2) - 1$.

For brevity, the details of the derivation of equation (2.2) have been omitted (see [2, 18, 28]). Equation (2.2) can be rewritten as

$$V_\tau = \frac{\sigma^2 S^2}{2} V_{SS} + (r - \lambda\kappa) S V_S - (r + \lambda)V + \lambda \int_0^\infty V(S\eta) g(\eta) d\eta. \tag{2.5}$$

Note that if we set $\lambda = 0$ in (2.5), then the classical Black-Scholes partial differential equation for pricing European option contracts is recovered [14, 28].

If we define

$$\mathcal{L}V \equiv V_\tau - \left(\frac{\sigma^2 S^2}{2} V_{SS} + (r - \lambda\kappa) S V_S - (r + \lambda)V + \lambda \int_0^\infty V(S\eta) g(\eta) d\eta \right) \tag{2.6}$$

and if $V^*(S, \tau)$ is the payoff, then the American option pricing problem can be stated as

$$\begin{aligned} \mathcal{L}V &\geq 0 \\ (V - V^*) &\geq 0 \\ (\mathcal{L}V = 0) &\vee (V - V^* = 0) \end{aligned} \tag{2.7}$$

where the notation $(\mathcal{L}V = 0) \vee (V - V^* = 0)$ denotes that either $(\mathcal{L}V = 0)$ or $(V - V^* = 0)$ at each point in the solution domain.

In the case of a put option, the boundary conditions are

$$V(S, \tau) = 0 \quad ; \qquad S \to \infty, \tag{2.8}$$

$$\mathcal{L}V = V_\tau - rV \quad ; \qquad S \to 0. \tag{2.9}$$

and the payoff for a put is

$$V^*(S) = V(S, \tau = 0) = \max(K - S, 0) \tag{2.10}$$

where K is the strike price. Other types of payoffs (e.g. call, digital) can be priced by suitable modifications to the payoff (2.10) and the boundary condition (2.8).

3 Implicit Discretization Methods: European Options

In this section we show how to discretized equation (2.5) for the case of a European option where there is no problem with early exercise. We do this by separately looking at the integral and partial differential equation components.

3.1 Discretization of the Integral Term

Straightforward discretization of equation (2.5) could use standard numerical discretization methods [26, 34, 35] for the differential operator combined with numerical integration methods such as Simpson's rule or Gaussian quadrature [24] for the integral term. However, this straightforward approach is computationally expensive [26]. Instead we transform the integral in equation (2.5) into a correlation integral. This allows efficient Fast Fourier Transform (FFT) methods to be used to evaluate the integral for all values of S.

Let

$$\mathcal{I}(S) = \int_0^\infty V(S\eta)g(\eta)d\eta. \tag{3.1}$$

and set $x = \log(S)$. Then using the change of variable

$$y = \log(\eta), \eta = e^y \quad \text{and} \quad d\eta = e^y dy, \tag{3.2}$$

gives

$$\mathcal{I}(x) = \int_{-\infty}^{\infty} \overline{V}(x+y)f(y)dy \tag{3.3}$$

where $f(y) = g(e^y)e^y$ and $\overline{V}(y,\tau) = V(e^y,\tau)$. Note that $f(y)$ is the probability density of a jump of size $y = \log(\eta)$. For example, with the density function given in (2.3) we would have a probability density given by

$$f(y) \quad = \quad \frac{e^{\left(-\frac{(y-\mu)^2}{2\gamma^2}\right)}}{\sqrt{2\pi}\gamma}. \tag{3.4}$$

Equation (3.3) corresponds to the correlation product of $\overline{V}(y)$ and $f(y)$.

The discrete form of the correlation integral is

$$\mathcal{I}_i \quad = \quad \sum_{j=-\frac{N}{2}+1}^{\frac{N}{2}} \overline{V}_{i+j} f_j \Delta y + O(\Delta y^2), \tag{3.5}$$

where $\mathcal{I}_i = \mathcal{I}(i\Delta x)$, $\overline{V}_j = \overline{V}(j\Delta x)$, $x_j = j\Delta x$, and

$$f_j \quad = \quad \int_{x_j-\frac{\Delta x}{2}}^{x_j+\frac{\Delta x}{2}} f(y)\; dy. \tag{3.6}$$

We have assumed that $\Delta y = \Delta x$, and that $\overline{V}(\log S) = V(S)$. We have also assumed in equation (3.5) that N is selected sufficiently large so that the solution in areas of interest is unaffected by the application of an asymptotic boundary condition for large values of S. In particular, we assume that $\overline{V}_{\frac{N}{2}+j}$, for $j > 0$ can be approximated by an asymptotic boundary condition. In practice, since f_j decays rapidly for $|j| > 0$, this does not cause any difficulty. Note that $\overline{V}_{-\frac{N}{2}+j}$, for $j < 0$, can be interpolated from known values V_k, since these points represent values near $S = 0$.

An important property to note, which will be used in later sections, is that

$$f_j \geq 0 \quad \text{for all } j \text{ and} \quad \sum_{j=-\frac{N}{2}+1}^{\frac{N}{2}} f_j\; \Delta y \leq 1, \tag{3.7}$$

since $f(y)$ is a probability density, and f_j is defined by equation (3.6). The inequality (3.7) arises since we have truncated the infinite integral (3.3).

The discrete form of the correlation integral (3.5) uses an equally spaced grid in $\log S$ coordinates. While this is convenient for FFT evaluation of the correlation integral, this is not particularly convenient for discretizing the PDE. We use an unequally spaced grid in S coordinates for the PDE discretization $[S_0, ..., S_p]$. Let

$$V_i^n \quad = \quad V(S_i, \tau_n) \ . \tag{3.8}$$

Now, $\overline{V}_j$ will not necessarily coincide with any of the discrete values V_k in equation (3.8). Consequently, we linearly interpolate (using linear Lagrange basis functions defined on the S grid) to determine the appropriate values, that is, if

$$S_{p_j} \quad \leq e^{j\Delta x} \leq \quad S_{p_j+1}, \tag{3.9}$$

then

$$\overline{V}_j = \psi_{p_j} V_{p_j} + (1 - \psi_{p_j}) V_{p_j+1} + O((S_{i+1} - S_i)^2), \tag{3.10}$$

where ψ_{p_j} is a non-negative interpolation weight. We are now faced with the problem that the integral I_i is evaluated at a point $S = e^{x_i}$ which does not coincide with a grid point S_k. We handle this by simply linearly interpolating the I_i to get the desired value. If

$$e^{x_{q_k}} \quad \leq S_k \leq \quad e^{x_{q_k+1}}, \tag{3.11}$$

then

$$\mathcal{I}(S_k) \quad = \quad \phi_{q_k} \mathcal{I}_{q_k} + (1 - \phi_{q_k}) \mathcal{I}_{q_k+1} + O((e^{x_{q_k}} - e^{x_{q_k+1}})^2) \ , \tag{3.12}$$

where ϕ_{q_k} is an interpolation weight. Note that

$$0 \leq \phi_i \leq 1 \text{ and } 0 \leq \psi_i \leq 1. \tag{3.13}$$

Putting equations (3.5), (3.10), and (3.12) together gives

$$\mathcal{I}(S_k) \quad = \quad \sum_{j=-\frac{N}{2}+1}^{\frac{N}{2}} \omega_j^k(V) f_j \Delta y, \tag{3.14}$$

where $V = [V_0, V_1, ..., V_p]^T$ and

$$\begin{aligned} \omega_j^k(V) \quad &= \quad \phi_a[\psi_b V_b + (1 - \psi_b) V_{b+1}] + (1 - \phi_a)[\psi_c V_c + (1 - \psi_c) V_{c+1}] \\ &= \quad \phi_a \psi_b V_b + \phi_a (1 - \psi_b) V_{b+1} + (1 - \phi_a) \psi_c V_c + (1 - \phi_a)(1 - \psi_c) V_{c+1} \end{aligned} \tag{3.15}$$

where $a = q_k$, $b = p_{q_k+j}$ and $c = p_{q_k+j+1}$. Note that $\omega_j^k(V)$ is linear in V, and that if $\mathbf{1} = [1, 1, ..., 1]$, then it follows from properties (3.13) that

$$\omega_j^k(\mathbf{1}) \quad = \quad 1 \quad \text{for all } k, j \quad .$$

The discrete sum (3.5) can be conveniently evaluated for all i using an FFT in $O(N \log N)$ flops, where N is the number of nodes in the $\log S$ grid. For details concerning the FFT method used to evaluate the sum (3.5), in particular about the choice of grid which minimizes wrap around effects, we refer the reader to [8]. While there are various methods for carrying out both forward and reverse FFTs for unequally spaced data [10, 27, 23], for our purposes these methods do not appear to be any more efficient than the straightforward interpolation approach described here (cf. [8]) . Essentially, this is because it is only necessary to evaluate the integral (3.5) correct to second order. It is not necessary to obtain highly accurate Fourier coefficients. The interpolation method described here will also allow us to prove convergence properties of the iterative algorithm used to solve the discrete nonlinear algebraic equations.

We also note for the special case of a Gaussian log normal probability density for the jump size, a Fast Gauss Transform [13, 4] could also be used to evaluate the correlation integral in $O(N)$ flops. However, recent work has indicated that non-log normal jump size probability densities may fit market data better than Gaussian log normal densities [16]. It is suggested in [4] that it may be possible to extend the Fast Gauss Transform to handle the density suggested in [16]. However, in this work, we will use an FFT method to compute the correlation integral, since this is a standard approach with readily available software. In any case, if a Fast Gauss Transform is used instead of an FFT, all the convergence results in subsequent sections are unchanged.

3.2 Discretization of the Full PIDE

Equation (2.5) can now be approximated by replacing derivatives by difference approximations. The integral term is approximated using equation (3.14). To avoid algebraic complexity, at this stage, we use a fully implicit method for the usual PDE, and use a weighted timestepping method for the jump integral term. The discrete equations can then be written as

$$\begin{aligned} V_i^{n+1}\left[1+(\alpha_i+\beta_i+r+\lambda)\Delta\tau\right] - \Delta\tau\beta_i V_{i+1}^{n+1} - \Delta\tau\alpha_i V_{i-1}^{n+1} \\ = V_i^n + (1-\theta_J)\Delta\tau\lambda \sum_{j=-\frac{N}{2}+1}^{\frac{N}{2}} \omega_j^i(V^{n+1}) f_j \Delta y + \theta_J \Delta\tau\lambda \sum_{j=-\frac{N}{2}+1}^{\frac{N}{2}} \omega_j^i(V^n) f_j \Delta y, \quad (3.16) \end{aligned}$$

where θ_J is a timeweighting such that $0 \le \theta_J \le 1$ and where α_i, β_i depend on the type of approximations used for the derivatives and second derivatives. Note that discretization (3.16) is only first order accurate in time.

There are a number of different discretizations of the derivative terms leading to various choices for α_i and β_i. Discretizing the first derivative term of equation (2.5) with central differences leads to

$$\begin{aligned} \alpha_{i,central} &= \frac{\sigma_i^2 S_i^2}{(S_i - S_{i-1})(S_{i+1} - S_{i-1})} - \frac{(r-\lambda\kappa)S_i}{S_{i+1} - S_{i-1}} \\ \beta_{i,central} &= \frac{\sigma_i^2 S_i^2}{(S_{i+1} - S_i)(S_{i+1} - S_{i-1})} + \frac{(r-\lambda\kappa)S_i}{S_{i+1} - S_{i-1}}. \end{aligned} \quad (3.17)$$

However if $\alpha_{i,central}$ or $\beta_{i,central}$ is negative, oscillations may appear in the solution. The oscillations can be avoided by using forward or backward differences at the problem nodes, leading to (forward difference)

$$\begin{aligned} \alpha_{i,forward} &= \frac{\sigma_i^2 S_i^2}{(S_i - S_{i-1})(S_{i+1} - S_{i-1})} \\ \beta_{i,forward} &= \frac{\sigma_i^2 S_i^2}{(S_{i+1} - S_i)(S_{i+1} - S_{i-1})} + \frac{(r-\lambda\kappa)S_i}{S_{i+1} - S_i}, \end{aligned} \quad (3.18)$$

or, (backward difference)

$$\alpha_{i,backward} = \frac{\sigma_i^2 S_i^2}{(S_i - S_{i-1})(S_{i+1} - S_{i-1})} - \frac{(r - \lambda\kappa)S_i}{S_{i+1} - S_i}$$
$$\beta_{i,backward} = \frac{\sigma_i^2 S_i^2}{(S_{i+1} - S_i)(S_{i+1} - S_{i-1})}. \tag{3.19}$$

Algorithmically, we decide between a central or forward discretization at each node for equation (3.16) as follows:

$$
\begin{aligned}
&\text{If } [\alpha_{i,central} \geq 0 \text{ and } \beta_{i,central} \geq 0] \text{ then} \\
&\quad \alpha_i = \alpha_{i,central} \\
&\quad \beta_i = \beta_{i,central} \\
&\text{ElseIf } [\beta_{i,forward} \geq 0] \text{ then} \\
&\quad \alpha_i = \alpha_{i,forward} \\
&\quad \beta_i = \beta_{i,forward} \\
&\text{Else} \\
&\quad \alpha_i = \alpha_{i,backward} \\
&\quad \beta_i = \beta_{i,backward} \\
&\text{EndIf}
\end{aligned} \tag{3.20}
$$

Note that the test condition (3.20) guarantees that α_i and β_i are non-negative. For typical values of σ, r and grid spacing, forward differencing is rarely required for single factor options. In practice, since this occurs at only a small number of nodes remote from the region of interest, the limited use of a low order scheme does not result in poor convergence as the mesh is refined. For situations where the low order method causes excessive numerical diffusion, a flux limiter can be used [35, 9]. As we shall see, requiring that all α_i and β_i are non-negative has important theoretical ramifications.

As $S \to 0$, equation (2.2) reduces to

$$V_\tau = -rV, \tag{3.21}$$

which is simply incorporated into the discrete equations (3.16) by setting $\alpha_i, \beta_i, \lambda = 0$ at $S_i = 0$.

In practice we truncate the S grid at some large value $S_p = S_{max}$, where we impose Dirichlet conditions. This is imposed replacing equation (3.16) at $S = S_{max} = S_p$, by

$$V_p^{n+1} = \text{specified}. \tag{3.22}$$

In [8], it is shown that the fully implicit scheme (3.16) is unconditionally stable for any θ_J, $0 \leq \theta_J \leq 1$. Note that this unconditional stability is due to fully implicit treatment of the term λV in equation (2.5). In [32], this term is treated explicitly.

3.3 Crank-Nicolson Discretization

The discretization method used in the previous subsection is only first order correct in the time direction. In order to improve the timestepping error, we can use a Crank-Nicolson method in the following

$$
\begin{aligned}
&\left[1+(\alpha_i+\beta_i+r+\lambda)(1-\theta)\Delta\tau\right]V_i^{n+1}-(1-\theta)\Delta\tau\beta_i V_{i+1}^{n+1}-(1-\theta)\Delta\tau\alpha_i V_{i-1}^{n+1}\\
&=\left[1-(\alpha_i+\beta_i+r+\lambda)\theta\Delta\tau\right]V_i^n+\theta\Delta\tau\alpha_i V_{i-1}^n+\theta\Delta\tau\beta_i V_{i+1}^n\\
&+(1-\theta_J)\lambda\Delta\tau\sum_{j=-\frac{N}{2}+1}^{j=\frac{N}{2}}\omega_j^i(V^{n+1})f_j\Delta y+\theta_J\lambda\Delta\tau\sum_{j=-\frac{N}{2}+1}^{j=\frac{N}{2}}\omega_j^i(V^n)f_j\Delta y. \qquad (3.23)
\end{aligned}
$$

where $\theta_J = \theta = \frac{1}{2}$ for Crank-Nicolson timestepping.

Remark 3.1 (Crank-Nicolson timestepping stability) *In the European case with jumps (i.e. $\lambda \neq 0$) it is shown in [8] that Crank-Nicolson timestepping is unconditionally algebraically stable.*

3.4 Matrix Formulation

For our purposes it is best to formulate our discretization using a more compact notation. To this end we can write equation (3.23) in matrix form as follows. Define matrices A and B such that

$$
[A \cdot V^n]_i = \Delta\tau\alpha_i V_{i-1}^n-(\alpha_i+\beta_i+r+\lambda)\Delta\tau V_i^n+\Delta\tau\beta_i V_{i+1}^n, \qquad (3.24)
$$

$$
[B \cdot V^n]_i = \sum_j b_{ij}V_j^n = \sum_{\ell=-\frac{N}{2}+1}^{\frac{N}{2}}\omega_\ell^i(V^n)f_\ell\Delta y. \qquad (3.25)
$$

We can then write a fully implicit ($\theta = \theta_J = 0$) or Crank Nicolson ($\theta = \theta_J = \frac{1}{2}$) discretization as

$$
[I-(1-\theta)A]V^{n+1} = [I+\theta A]V^n+(1-\theta_J)\lambda\Delta\tau BV^{n+1}+\theta_J\lambda\Delta\tau BV^n. \qquad (3.26)
$$

We remark that the entries of the matrix B have the property

$$
0 \le b_{ij} \le 1 \text{ and } \sum_j b_{ij} \le 1, \qquad (3.27)
$$

a fact that will be important in the error analysis given later.

4 American Options

We can extend equation (3.26) to the American option case by using a penalty method [12]. In this section, we develop an iterative algorithm for solution of the nonlinear discretized equations that result from such a penalty method. We also show that this algorithm is globally convergent.

4.1 The Penalty Method

The basic idea of the penalty method is simple. We replace problem (2.7) by the nonlinear PIDE [11]

$$
V_\tau = \frac{\sigma^2S^2}{2}V_{SS}+(r-\lambda\kappa)SV_S-(r+\lambda)V+\lambda\int_0^\infty V(S\eta)g(\eta)d\eta+\rho\max(V^*-V,0), \qquad (4.1)
$$

where, in the limit as the positive penalty parameter $\rho \to \infty$, the solution satisfies $V \geq V^*$.

As shown in [12], in the case where $\lambda = 0$ (no jumps) the penalty method can be used to obtain an approximate solution to the discretized complementarity problem (2.7) at each timestep. For details regarding the penalty method, we refer the reader to [12].

Let V^* be the vector of payoffs obtained upon exercise, and let the diagonal matrix P be given by

$$P(V^{n+1})_{ii} = \begin{cases} Large & \text{if } V_i^{n+1} < V_i^* \\ 0 & \text{otherwise.} \end{cases} \tag{4.2}$$

Then the matrix form of the discrete equations for the penalized method is given by

$$\begin{aligned} &[I - (1-\theta)A + P(V^{n+1})]V^{n+1} = \\ &\qquad [I + \theta A]V^n + (1-\theta_J)\lambda\Delta\tau B V^{n+1} + \theta_J \lambda \Delta\tau B V^n + \left[P(V^{n+1})\right] V^*. \end{aligned} \tag{4.3}$$

Dirichlet boundary conditions are enforced at $i = imax$ by setting

$$\begin{aligned} A_{ij} &= 0 \ ; \ i = imax \\ P_{ij} &= 0 \ ; \ i = imax \\ b_{ij} &= 0 \ ; \ i = imax \\ V_{imax}^{n+1} &= V_{imax}^n . \end{aligned} \tag{4.4}$$

In order to avoid algebraic complication, we assume that the Dirichlet condition at S_{imax} is independent of time.

Remark 4.1 (Stability of a fully implicit discretization) *It is straightforward to show, via a maximum analysis, that setting $\theta = 0$ in equation* (4.3) *results in an unconditionally stable method for any θ_J, $0 \leq \theta_J \leq 1$.*

4.2 The Matrix Iteration

In order to solve equation (4.3), we use the following iteration scheme (assuming $\theta = \theta_J$)

Iteration

Let $(V^{n+1})^0 = V^n$

Let $\hat{V}^k = (V^{n+1})^k$

Let $\hat{P}^k = P((V^{n+1})^k)$

For $k = 0, 1, 2, \ldots$ until convergence

Solve

$$\begin{aligned} &\left[I - (1-\theta)A + \hat{P}^k\right] \hat{V}^{k+1} \\ &= [I + \theta A]\, V^n + \hat{P}^k V^* \\ &+ (1-\theta)\lambda\Delta\tau B\hat{V}^k + \theta\lambda\Delta\tau B V^n \end{aligned} \tag{4.5}$$

If $\max_i \dfrac{|\hat{V}_i^{k+1} - \hat{V}_i^k|}{\max(1, |\hat{V}_i^{k+1}|)} < tolerance$ then quit

EndFor

Note that the matrix vector multiplies in iteration (4.5) ($B\hat{V}^k$) can be computed in $O(N \log N)$ operations using an FFT. As a result, work for each step of this iteration consists of

- Interpolation of the solution of the original S grid onto an equally spaced $\log S$ grid.
- A forward FFT of the interpolated solution.
- Evaluation of the correlation product in the frequency domain,
- An inverse FFT.
- Interpolation of the correlation product from the $\log S$ grid onto the original S grid.
- A factor and solve of the tridiagonal matrix $\left[I - (1-\theta)A + \hat{P}^k\right]$.

Consequently, the work for each iteration is dominated by the forward and back FFTs.

4.3 Convergence of the Iteration

In this subsection we consider the problem of convergence of the iteration scheme (4.5). Convergence is proved by a number of properties of the intermediate quantities $\hat{V}^k$.

Lemma 4.1 (Bounded iterates) *Suppose that $\alpha_i, \beta_i \geq 0$ for all i in the discretization* (3.23) *and that B has the properties* (3.27)*. Then, for a given timestep, all iterates $\hat{V}^{k+1}$ in scheme* (4.5) *are bounded independent of k.*

Proof. Writing iteration (4.5) in component form gives

$$\begin{aligned}\left[1 + (1-\theta)(\alpha_i + \beta_i + \lambda + r)\Delta\tau + \hat{P}_{ii}^k\right] \hat{V}_i^{k+1} &= c_i + \hat{P}_{ii}^k V_i^* + (1-\theta)\lambda\Delta\tau \sum_j b_{ij} \hat{V}_j^k \\ &\quad + (1-\theta)\Delta\tau \left[\alpha_i \hat{V}_{i-1}^{k+1} + \beta_i \hat{V}_{i+1}^{k+1}\right], \end{aligned} \tag{4.6}$$

where

$$c_i = \left(\left[I + \theta A\right] V^n + \theta\lambda\Delta\tau B V^n\right)_i . \tag{4.7}$$

From the component form (4.6), it follows that ($i < imax$)

$$\begin{aligned}\left[1 + (1-\theta)(\alpha_i + \beta_i + \lambda + r)\Delta\tau + \hat{P}_{ii}^k\right] |\hat{V}_i^{k+1}| &\leq \|c\|_\infty + \hat{P}_{ii}^k \|V^*\|_\infty + (1-\theta)\lambda\Delta\tau \|\hat{V}^k\|_\infty \\ &\quad + (1-\theta)\Delta\tau \left[\alpha_i + \beta_i\right] \|\hat{V}^{k+1}\|_\infty. \end{aligned} \tag{4.8}$$

Let m be an index such that

$$|\hat{V}_m^{k+1}| = \max_i |\hat{V}_i^{k+1}| = \|\hat{V}^{k+1}\|_\infty. \tag{4.9}$$

Note that if $m = imax$ then we have

$$\|\hat{V}^{k+1}\|_\infty = |V_{imax}^n| \leq \|V^n\|_\infty. \tag{4.10}$$

Assume now that $m < imax$. Then from equations (4.8) and (4.9) we obtain

$$\left[1 + (1-\theta)(\lambda + r)\Delta\tau + \hat{P}_{mm}^k\right] \|\hat{V}^{k+1}\|_\infty \leq \|c\|_\infty + \hat{P}_{mm}^k \|V^*\|_\infty + (1-\theta)\lambda\Delta\tau \|\hat{V}^k\|_\infty. \tag{4.11}$$

Equation (4.11) then gives

$$\|\hat{V}^{k+1}\|_\infty \quad \leq \quad \frac{\|c\|_\infty + P^k_{mm}\|V^*\|_\infty}{1+(1-\theta)(\lambda+r)\Delta\tau + P^k_{mm}} + \frac{(1-\theta)\lambda\Delta\tau\|\hat{V}^k\|_\infty}{1+(1-\theta)(\lambda+r)\Delta\tau + P^k_{mm}}. \tag{4.12}$$

Let

$$C_1 = \max(\|c\|_\infty, \|V^*\|_\infty) \text{ and } C_2 = \frac{(1-\theta)\lambda\Delta\tau}{1+(1-\theta)(\lambda+r)\Delta\tau} \tag{4.13}$$

so that

$$\|\hat{V}^{k+1}\|_\infty \quad \leq \quad C_1 + \frac{(1-\theta)\lambda\Delta\tau\|\hat{V}^k\|_\infty}{1+(1-\theta)(\lambda+r)\Delta\tau + P^k_{mm}} \leq C_1 + C_2\|\hat{V}^k\|_\infty. \tag{4.14}$$

Summing over the index k, equation (4.14) gives

$$\|\hat{V}^{k+1}\|_\infty \quad \leq \quad C_1 \sum_{i=0}^{k} C_2^i + C_2^{k+1}\|\hat{V}^0\|_\infty. \tag{4.15}$$

Noting that $\hat{V}^0 = V^n$ and that $C_2 < 1$, equation (4.15) then gives

$$\|\hat{V}^{k+1}\|_\infty \quad \leq \quad \|V^n\|_\infty + \frac{C_1}{1-C_2}, \tag{4.16}$$

where C_1, C_2 are independent of k. From equation (4.10) we see that bound (4.16) is also valid for $m = imax$ and therefore for all m. □

After some manipulation, we can write iteration (4.5) as

$$\left[I - (1-\theta)A + \hat{P}^k\right](\hat{V}^{k+1} - \hat{V}^k) = (\hat{P}^k - \hat{P}^{k-1})(V^* - \hat{V}^k) + (1-\theta)\lambda\Delta\tau B(\hat{V}^k - \hat{V}^{k-1}) \ . \tag{4.17}$$

In order to prove convergence of the scheme (4.5), it will be convenient to determine the sign of $(\hat{P}^k - \hat{P}^{k-1})(V^* - \hat{V}^k)$ in equation (4.17).

Lemma 4.2 (Positive penalty term) *Given the definition of the penalty matrix $\hat{P}^k$ from equation (4.2), and the iteration scheme (4.5), we have that*

$$(\hat{P}^k - \hat{P}^{k-1})(V^* - \hat{V}^k) \quad \geq \quad 0 \quad \textit{for all } k \geq 1 \ . \tag{4.18}$$

Proof. For each index i we have two possible cases. If $\hat{V}^k_i < V^*_i$ for component i then $\hat{P}^k_{ii} = Large$ so that

$$(\hat{P}^k_{ii} - \hat{P}^{k-1}_{ii})(V^* - \hat{V}^k)_i = (Large - \hat{P}^{k-1}_{ii})(V^* - \hat{V}^k)_i \geq 0.$$

On the other hand if $\hat{V}^k_i \geq V^*_i$ then $\hat{P}^k_{ii} = 0$ hence

$$(\hat{P}^k_{ii} - \hat{P}^{k-1}_{ii})(V^* - \hat{V}^k)_i = -\hat{P}^{k-1}_{ii}(V^* - \hat{V}^k)_i \geq 0.$$

Thus for all $k \geq 1$ we always have

$$(\hat{P}^k - \hat{P}^{k-1})(V^* - \hat{V}^k) \geq 0. \tag{4.19}$$

□

Recall that an *M-matrix* has positive diagonals, non-positive off-diagonals, the row sums are non-negative with at least one such sum being positive. Such a matrix has the useful property that all the entries in its inverse are non-negative.

Lemma 4.3 (M-matrices) *Let A, B and $\hat{P}^k$ be given by (3.24), (3.25) and (4.2), respectively. Assume that $\alpha_i \geq 0, \beta_i \geq 0$ in equation (3.23), that B has the properties (3.27) and that we use a Dirichlet boundary condition in (4.4). Then both*

$$[I - (1-\theta)A + P^k] \text{ and } [I - (1-\theta)A + P^k - (1-\theta)\lambda\Delta\tau B] \tag{4.20}$$

are M matrices.

Proof. It follows from equations (3.23), (3.24) and (3.27) that both of the above matrices have positive diagonals, non-positive off-diagonals and with row sum non-negative. Since a Dirichlet condition is imposed at $i = imax$ (4.4), for both matrices there is at least one row which has a strictly positive row sum. □

Recall that the discrete equations can be written as ($\theta = \theta_J$)

$$[I - (1-\theta)A + \hat{P}^{n+1} - (1-\theta)\lambda\Delta\tau B]V^{n+1} = [I + \theta A]V^n + \theta\lambda\Delta\tau BV^n + \hat{P}^{n+1}V^*. \tag{4.21}$$

We can now prove the following result

Theorem 4.1 (Uniqueness of solution) *Under the conditions required for Lemmas 4.2 and 4.3, any solution to equation (4.21) for a given timestep is unique.*

Proof. Suppose that we have two solutions V_1, V_2 to equation (4.21). Let $\hat{P}_1 \equiv P(V_1)$ and $\hat{P}_2 \equiv P(V_2)$ so that

$$[I - (1-\theta)A + \hat{P}_1 - (1-\theta)\lambda\Delta\tau B]V_1 = [I + \theta A]V^n + \theta\lambda\Delta\tau BV^n + \hat{P}_1 V^* \tag{4.22}$$

and

$$[I - (1-\theta)A + \hat{P}_2 - (1-\theta)\lambda\Delta\tau B]V_2 = [I + \theta A]V^n + \theta\lambda\Delta\tau BV^n + \hat{P}_2 V^*. \tag{4.23}$$

Equation (4.22) can be written as

$$[I - (1-\theta)A + \hat{P}_2 - (1-\theta)\lambda\Delta\tau B]V_1 + (\hat{P}_1 - \hat{P}_2)V_1 = [I + \theta A]V^n + \theta\lambda\Delta\tau BV^n + \hat{P}_1 V^* \tag{4.24}$$

which after subtracting (4.23) from (4.24) gives

$$[I - (1-\theta)A + \hat{P}_2 - (1-\theta)\lambda\Delta\tau B](V_1 - V_2) = (\hat{P}_1 - \hat{P}_2)(V^* - V_1)\ . \tag{4.25}$$

Using the same arguments as in the proof of Lemma 4.2 we have that $(\hat{P}_1 - \hat{P}_2)(V^* - V_1) \geq 0$. From Lemma 4.3 it follows that $I - (1-\theta)A + \hat{P}_2 - (1-\theta)\lambda\Delta\tau B$ is an M-matrix and hence $(V_1 - V_2) \geq 0$. Interchanging subscripts, we also have that $(V_2 - V_1) \geq 0$ and hence $V_1 = V_2$. □

Before we prove our main convergence result, we need the following Lemma.

Lemma 4.4 (Norm of an iteration matrix) *Let A, B and $\hat{P}^k$ be given by (3.24), (3.25) and (4.2), respectively. Assume that $\alpha_i \geq 0, \beta_i \geq 0$ in equation (3.23), that B has the properties (3.27) and that we use a Dirichlet boundary condition in (4.4). Then for $\mathcal{Q}^k = [I - (1-\theta)A + P^k]$ we have*

$$\|[\mathcal{Q}^k]^{-1} B\|_\infty \quad \leq \quad \frac{1}{1 + (1-\theta)(r+\lambda)\Delta\tau} \ . \tag{4.26}$$

Proof. Let y, z be vectors, z arbitrary, satisfying $\mathcal{Q}^k y = Bz$. Then in component form we have that $y_{imax} = 0$ and for $i < imax$:

$$[1 + (1-\theta)(\alpha_i + \beta_i + r + \lambda)\Delta\tau + \hat{P}_{ii}^k] y_i = (1-\theta)\alpha_i \Delta\tau y_{i-1} + (1-\theta)\beta_i \Delta\tau y_{i+1} + \sum_j b_{ij} z_j. \tag{4.27}$$

From the properties of $\alpha_i, \beta_i, \hat{P}^k, B$, we then immediately have that

$$\|y\|_\infty \quad \leq \quad \frac{\|z\|_\infty}{1 + (1-\theta)(r+\lambda)\Delta\tau}, \tag{4.28}$$

giving (4.26). □

We are now in a position to prove our main convergence result

Theorem 4.2 (Convergence of iteration (4.5)) *Let A, B and $\hat{P}^k$ be given by (3.24), (3.25) and (4.2), respectively. Assume that $\alpha_i \geq 0, \beta_i \geq 0$ in equation (3.23), that B has the properties (3.27) and that we use a Dirichlet boundary condition in (4.4). Then iteration (4.5) is globally convergent to the unique solution of equation (4.21) for any initial iterate $\hat{V}^0$.*

Proof. Iteration (4.5) can be written as

$$\mathcal{Q}^k(\hat{V}^{k+1} - \hat{V}^k) \quad = \quad (\hat{P}^k - \hat{P}^{k-1})(V^* - \hat{V}^k) + (1-\theta)\lambda\Delta\tau B(\hat{V}^k - \hat{V}^{k-1}), k \geq 1, \tag{4.29}$$

where $\mathcal{Q}^k \equiv I - (1-\theta)A + \hat{P}^k$. For any $k \geq 1$ we can then write

$$(\hat{V}^{k+1} - \hat{V}^k) = U^k + W^k \cdot (\hat{V}^1 - \hat{V}^0), \tag{4.30}$$

with

$$\begin{aligned} U^k \quad &= \quad [\mathcal{Q}^k]^{-1}(\hat{P}^k - \hat{P}^{k-1})[V^* - \hat{V}^k] \\ &\quad + (1-\theta)\lambda\Delta\tau[\mathcal{Q}^k]^{-1} B [\mathcal{Q}^{k-1}]^{-1}(\hat{P}^{k-1} - \hat{P}^{k-2})[V^* - \hat{V}^{k-1}] \\ &\quad + \ \ldots \\ &\quad + [(1-\theta)\lambda\Delta\tau]^{k-1}[\mathcal{Q}^k]^{-1} B [\mathcal{Q}^{k-1}]^{-1} B \ \ldots \ [\mathcal{Q}^1]^{-1} B(\hat{P}^1 - \hat{P}^0)[V^* - \hat{V}^1], \\ W^k \quad &= \quad [(1-\theta)\lambda\Delta\tau]^k [\mathcal{Q}^k]^{-1} B [\mathcal{Q}^{k-1}]^{-1} B \ \ldots \ [\mathcal{Q}^1]^{-1} B. \end{aligned} \tag{4.31}$$

We show that both U^k and W^k tend to zero as k gets large.

Note first that both $U^k \geq 0$ and $W^k \geq 0$. To show this we have that Lemma 4.2 implies $(\hat{P}^k - \hat{P}^{k-1})(V^* - \hat{V}^k) \geq 0$ for all $k \geq 1$ while from Lemma 4.3, we have that $[\mathcal{Q}^k]^{-1} \geq 0$. Since $B \geq 0$ we have that all the components in U^k are non-negative. A similar statement is true for W^k since $\mathcal{Q}^k$ is an M-matrix and since $B \geq 0$.

From Lemma 4.4 and equation (4.31) we have that for each i

$$\|W^i\|_\infty \le \left[\frac{(1-\theta)\lambda\Delta\tau}{1+(1-\theta)(r+\lambda)\Delta\tau}\right]^i, \tag{4.32}$$

and hence

$$\|\sum_{i=1}^{k} W^i\|_\infty \le \sum_{i=1}^{k}\left[\frac{(1-\theta)\lambda\Delta\tau}{1+(1-\theta)(r+\lambda)\Delta\tau}\right]^i \le \left[\frac{(1-\theta)\lambda\Delta\tau}{1+(1-\theta)r\Delta\tau}\right]. \tag{4.33}$$

Thus $\{\sum_{i=1}^{k} W^i\}_{k=1,\ldots}$ is a sequence of non-decreasing terms which are bounded from above. As such the sequence converges. In particular we have that W^k tends to zero as k tends to infinity.

Summing over the index k equation (4.30) gives

$$\hat{V}^{k+1} = \hat{V}^1 + \sum_{i=1}^{k} U^i + \sum_{i=1}^{k} W^i \cdot (\hat{V}^1 - \hat{V}^0). \tag{4.34}$$

From equation (4.32) we have that $(\sum_{i=1}^{k} W^i \cdot (\hat{V}^1 - \hat{V}^0)\ )$ converges to a finite value, furthermore from Lemma 4.1 the left hand side of equation (4.34) is bounded from above. Thus the sequence $\{\sum_{i=1}^{k} U^i\}_{k=1,\ldots}$ is both non-decreasing and bounded from above. Hence this sequence also converges and so U^k approaches 0 as k approaches infinity.

Thus a convergent limit exists, and from Theorem 4.1, this is the unique solution to equation (4.21). □

Remark 4.2 (Monotonicity) *Previous convergence results for penalty methods have typically required that the quantities V^k are monotonic (cf. [12]). From equation (4.30) we see that we do not necessarily have this property if $\hat{V}^1 < \hat{V}^0$. Of course we could ensure monotonicity by forcing $\hat{V}^1 \ge \hat{V}^0$. However the proof of Theorem 4.2 shows that this is not really required. In addition, numerical experiments demonstrate that forcing monotonicity does not improve convergence.*

Remark 4.3 (Speed of convergence) *Typically, $\lambda\Delta\tau \ll 1$. For example, for S&P 500 data, $\lambda \simeq .1$ [2], and a typical timestep is $\Delta\tau < .1$, giving $\lambda\Delta\tau \simeq .01$. For $\lambda\Delta\tau \ll 1$, equation (4.32) becomes*

$$\|W^i\|_\infty \ \simeq \ ((1-\theta)\lambda\Delta\tau)^i,$$

so that the term

$$\sum_{i=1}^{k} W^i \cdot (\hat{V}^1 - \hat{V}^0),$$

in equation (4.34) converges very rapidly. Our experience with the penalty method for American options with Brownian motion [12] (no jumps) indicates that the term

$$\sum_{i=1}^{k} U^i,$$

in equation (4.34) also converges rapidly. This rapid convergence will be confirmed with the numerical examples of the next section.

Remark 4.4 (Non M-matrices) *Our proof of convergence relies on the fact that the discretization of the PDE resulted in an M-matrix. However, we have observed (experimentally) that convergence is still rapid even if the coefficient matrix is not an M-matrix.*

An example where our discretization is not an M-matrix appears naturally as follows. It is often convenient to impose an asymptotic linearity boundary condition [28]

$$V_{SS} \quad = \quad 0 \quad ; \quad S \rightarrow \infty \tag{4.35}$$

This boundary condition is particularly useful in complex path dependent cases where it is difficult to determine the asymptotic form of the solution [30].

Condition (4.35) is enforced by setting $(V_{SS})_i^n = 0$ at $i = imax$, and using a backward difference approximation for V_S. A little thought shows that for $r > 0$, this corresponds to using downwind weighting of the first order term at $i = imax$. This method can be shown to be stable [31], and the matrix solution can be obtained using Gaussian elimination without pivoting as long as the order of elimination is $i = 0, 1, \ldots$. However, in this case the coefficient matrix is no longer an M-matrix. In fact iterative methods for obtaining complementarity solutions may fail due to a small pivot since these methods [7] require repeated elimination steps in the forward ($i = 0, 1, \ldots$) and reverse ($i = imax, imax - 1, \ldots$) directions.

5 Numerical Examples

In this section we give a number of numerical examples which illustrate the performance and convergence of our iteration scheme. The examples are chosen to demonstrate that for practical values of the parameters, the iterative method for solving the discrete nonlinear algebraic equations at each timestep converges rapidly. In fact, the number of iterations required for convergence of European options (with jumps) is on average, almost the same as the corresponding American option. We also verify that quadratic convergence is obtained as the grid and timesteps are refined, for Crank-Nicolson timestepping. In [12] the authors showed experimentally that in order to restore quadratic convergence when pricing American put option, a timestep selector must be used. Consequently, as in [12] we use a timestep selector based on a modified form of that suggested in [15]. Given an initial timestep $\Delta\tau^{n+1}$, then a new timestep is selected so that

$$\Delta\tau^{n+2} = \left(\min_i \left[\frac{\text{dnorm}}{\frac{|V(S_i,\tau^n+\Delta\tau^{n+1})-V(S_i,\tau^n)|}{\max(\text{ D},|V(S_i,\tau^n+\Delta\tau^{n+1})|,|V(S_i,\tau^n)|)}} \right] \right) \Delta\tau^{n+1}, \tag{5.1}$$

where dnorm is a target relative change (during the timestep) specified by the user. The scale D prevents the timestep selector from taking an excessive number of timesteps in regions where the value is small. In general it is set to D = 1.0 for options valued in dollars.

5.1 American Put Option Example

As a first example, we consider the case of an American put option under a jump diffusion process. Table 1 lists the data used for this example. This data is essentially the (rounded) data obtained in [2] by matching option prices on the $S\&P$ 500. As discussed in [2], the magnitude of and frequency of the jumps obtained by calibration with option prices is larger than historical data would suggest, indicating an effect of risk preferences of investors.

T	.25
λ	.10
γ	.45
μ	-.9
σ	.15
r	.05
K	100
B-S Implied Volatility	.1886

Table 1: *Data used in the put option example.*

Nodes	Timesteps	Itns	Value	Change	Ratio
127	40	121	3.2373512		
254	100	239	3.2404239	.0030727	
508	218	507	3.2410657	.0006418	4.8
1016	453	1044	3.2412099	.0001442	4.5
2032	924	2106	3.2412435	.0000336	4.3

Table 2: *Value of an American put, under jump diffusion process, $S = 100$, $t = 0$, data as in Table 1. Itns is the total number of iterations required in algorithm (4.5), for all timesteps. Change is the change from one level of refinement and the next. Ratio is ratio of changes. Crank-Nicolson timestepping is used with the timestep selector defined by* (5.1)*, where dnorm* $= .05$ *and the initial timestep* $dt_{init} = .005$*, on the coarsest grid.*

Following [2], we assume that the jump magnitude density is given by the log normal distribution (3.4). We also compare the option pricing solution obtained by a jump diffusion model to a constant volatility Black-Scholes solution. In order to make a fair comparison between these two approaches, the constant volatility Black-Scholes solution is computed using the implied volatility shown in Table 1. This implied volatility is the constant volatility which reproduces the jump-diffusion price at $S = K = 100$ for a European call (assuming no jumps). The tolerance used in algorithm (4.5) is $tolerance = 10^{-6}$, and, as suggested in [12], we use $Large = 1/tolerance$ in equation (4.2). It is shown in [12] that the relative error in enforcing the American constraint is approximately $O(1/Large)$, so that the computed results should have at least six digit accuracy. This was verified in some numerical tests with $Large = 10^{10}$, which showed no change in the solution (compared to $Large = 10^6$) to about eight digits. We have set $S_{max} = 10K$, where K is the strike. Some experiments with solutions computed with $S_{max} = 50K$ resulted in no change to the solution to eight digits.

Table 2 shows the results for a convergence study. The timestep selector (5.1) is used. The modification for Crank-Nicolson timestepping suggested in [25] (initial two steps fully implicit, followed by Crank-Nicolson thereafter) is used, since the payoff is non-smooth.

On each grid refinement, new nodes are inserted between each pair of coarse grid nodes. The timestep selector parameter dnorm (5.1) and the initial timestep are also halved on each refinement, so that the timesteps are approximately halved on each refinement. Since the ratio of changes in Table 2 appears to be approaching four as the grid and timesteps are refined, this indicates that

Nodes	Timesteps	Itns	Value	Change	Ratio
127	40	120	3.14666646		
254	99	250	3.14849763	.0018312	
508	216	456	3.14889738	.0003998	4.6
1016	448	896	3.14899401	.0000967	4.1
2032	913	1826	3.14901783	.0000238	4.1

Table 3: *Value of a European put, jump diffusion, $S = 100$, $t = 0$, data as in Table 1. Itns is the total number of iterations required in algorithm (4.5), for all timesteps. Change is the change from one level of refinement and the next. Ratio is ratio of changes. Crank-Nicolson timestepping is used with the timestep selector defined by* (5.1)*, where dnorm* $= .05$ *and the initial timestep* $dt_{init} = .005$*, on the coarsest grid.*

convergence is approximately quadratic in ΔS and $\Delta\tau$, where

$$\begin{aligned} \Delta S &= \max_i (S_{i+1} - S_i) \\ \Delta \tau &= \max_n \left(\tau^{n+1} - \tau^n\right) \ . \end{aligned} \tag{5.2}$$

Table 2 also indicates that the average number of iterations per timestep for algorithm (4.5) is of the order $2 - 3$.

Table 3 shows similar convergence results for a European option using the same data as in Table 1. Comparing Tables 2 and 3, we can see that the average number of iterations per step is also $2 - 3$, indicating that the influence of the penalty term on the iteration (4.5) is quite small.

Figure 1 compares the jump diffusion solution (jumps) for an American option with a constant volatility Black-Scholes solution (no jumps). The volatility used in the no-jump model is the implied volatility which reproduces the jump model price at $S = K = 100$ for a European call. Note that the jump model is significantly more valuable than the non-jump model at $S = 110$, due to the high probability that a downward jump in the asset price can occur. Figure 2 show the delta (V_S) and gamma (V_{SS}) for the jump and no-jump models. Delta and gamma are hedging parameters [28].

5.2 American Butterfly Example

A more challenging numerical example is given by the solution to an American butterfly. A butterfly option has the payoff

$$V^* = \max(S - K_1, 0) - 2\max(S - (K_1 + K_2)/2, 0) + \max(S - K_2, 0) \ . \tag{5.3}$$

In our example, we choose $K_1 = 90, K_2 = 110$. This payoff can be constructed by holding a long position in two calls struck at K_1, K_2, and short position in two calls struck at $(K_1 + K_2)/2$. In our example, we assume the existence of an American style contract which specifies the payoff (5.3), and we assume that the option can only be early exercised as a unit.

Table 4 shows a convergence study for the American butterfly. On each refinement, new nodes are inserted between each two coarse grid nodes and the timestep size is approximately halved. Two timestepping methods were used. The implicit American constraint used the algorithm (4.5). The explicit American constraint used the following modification. Using the notation introduced

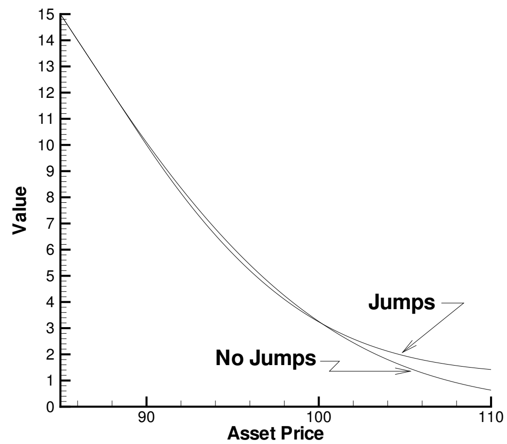

Figure 1: *American put option value, jump diffusion model compared with model with no jumps. The no jump model has an implied volatility which gives the same price as the jump model for a European option at the money. Data as in Table 1.*

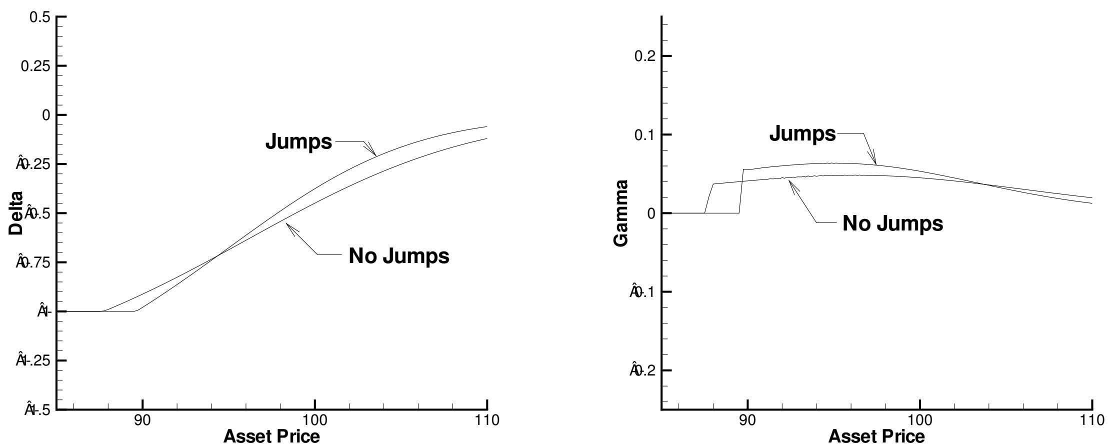

Figure 2: *American put option delta (V_S), and gamma (V_{SS}), jump diffusion model compared with model with no jumps. The no jump model has an implied volatility which gives the same price as the jump model for a European option at the money. Data as in Table 1.*

in equation (4.3), we iterate for $\mathcal{V}^{k+1}$ (setting the penalty term to zero)

$$[I - \frac{A}{2}]\mathcal{V}^{k+1} = [I + \frac{A}{2}]V^n + \frac{\lambda \Delta \tau}{2} B\mathcal{V}^k + \frac{\lambda \Delta \tau}{2} BV^n. \tag{5.4}$$

After the iteration has converged, we then set

$$V^{n+1} = \max(V^*, \mathcal{V}^{k+1}) \ . \tag{5.5}$$

In this case, we would expect that the time truncation error is $O(\Delta\tau)$. In fact, this is clearly demonstrated in Table 4, since the ratio of changes appears to be asymptotically 4 for the implicit American approach (which indicates quadratic convergence) compared to the asymptotic ratio of 2 (linear convergence) for the explicit American method. It is interesting to see from Table 4 that the number of iterations for the implicit American method is only slightly greater than for the explicit American technique. This indicates that we can impose the American constraint implicitly at very little computational expense compared to an explicit constraint method.

For comparison, we also show in Table 4 the results for a fully implicit discretization of the PDE term, an explicit evaluation of the correlation integral, and an explicit application of the American constraint. More precisely,

$$\begin{aligned} [I - A]\mathcal{V}^{n+1} &= V^n + \lambda \Delta \tau B V^n \\ V^{n+1} &= \max(V^*, \mathcal{V}^{n+1}) \ . \end{aligned} \tag{5.6}$$

This method is unconditionally stable (a straightforward extension of the proofs in [8] shows this), and is clearly the cheapest method (per timestep). However, convergence is clearly only first order. As shown in [12], an explicit application of the American constraint can result in oscillations in gamma near the exercise boundary.

Figure 3 shows the value of an American butterfly, with the jump diffusion model (jumps) and the constant volatility Black-Scholes model (no-jumps). As described above, the constant volatility Black-Scholes model uses an implied volatility which reproduces the jump model price at $S = 100$ for a vanilla European call. The corresponding delta and gamma are shown in Figure 4.

6 Conclusion

In this article, we have developed an iterative method for solution of the discrete penalized equations which result from discretization of the differential-integral complementarity problem for pricing American options on assets which follow a jump diffusion process. We have also derived sufficient conditions for the global convergence of this iteration (at each timestep).

Unlike previous work, the method developed here uses implicit timestepping for both the correlation integral term and the American constraint. Consequently, we expect higher order convergence (in terms of timestepping error) compared with previous methods which treat the correlation integral or the American constraint explicitly. Quadratic convergence is observed in our numerical tests, compared to linear convergence which occurs if explicit methods are used.

A sufficient condition for global convergence of the iterative method for solution of the discretized penalized jump diffusion equations (at each timestep) is that the coefficient matrix is an M-matrix. However, on the basis of numerous numerical experiments, this does not appear to be a necessary condition. For single factor options, the commonly used $V_{SS} = 0$ boundary condition destroys the M-matrix property. For two factor options (such as stochastic volatility models), the M-matrix property no longer holds if there is a non-zero correlation between the asset price and

Nodes	Timesteps	Itns	Value	Change	Ratio
Implicit American constraint					
127	44	133	5.2490795		
254	111	249	5.2511148	.0020353	
508	246	546	5.2515158	.0004010	5.1
1016	511	1130	5.2515839	.0000689	5.8
2032	1042	2280	5.2516010	.0000171	4.0
Method (5.4-5.5)					
127	43	129	5.2296200		
254	111	222	5.2429331	.0179502	
508	246	492	5.2475702	.0046371	3.9
1016	511	1022	5.2496169	.0020467	2.3
2032	1041	2082	5.2506144	.0009975	2.1
Method (5.6)					
127	43	43	5.1997845		
254	111	111	5.2310288	.0312443	
508	246	246	5.2423694	.0113406	2.8
1016	511	511	5.2471667	.0047973	2.4
2932	1041	1041	5.2493929	.0022262	2.0

Table 4: *Value of an American butterfly, $S = 105$, $t = 0$, jump diffusion, data as in Table 1. Implicit constraint, American constraint solved implicitly. Algorithm (5.4-5.5): American constraint imposed explicitly. Algorithm (5.6): fully implicit PDE, explicit correlation integral, explicit American constraint. Itns is the total number of iterations required in algorithm (4.5), for all timesteps. Change is the change from one level of refinement and the next. Ratio is ratio of changes. Data as in Table 1. Crank-Nicolson timestepping is used with the timestep selector defined by* (5.1)*, where $dnorm = .05$ and the initial timestep $dt_{init} = .005$, on the coarsest grid.*

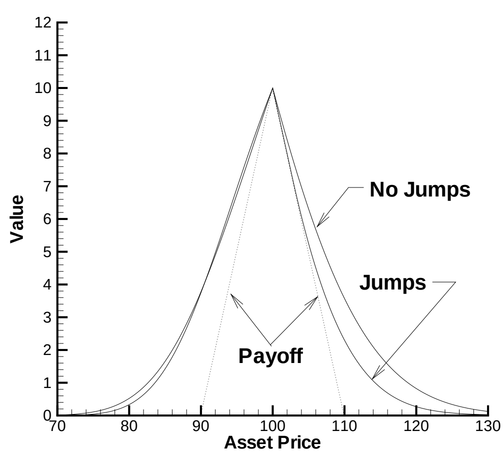

Figure 3: *American butterfly option value, jump diffusion model compared with model with no jumps. The no jump model has an implied volatility which gives the same price as the jump model for a European option at the money. Data as in Table 1.*

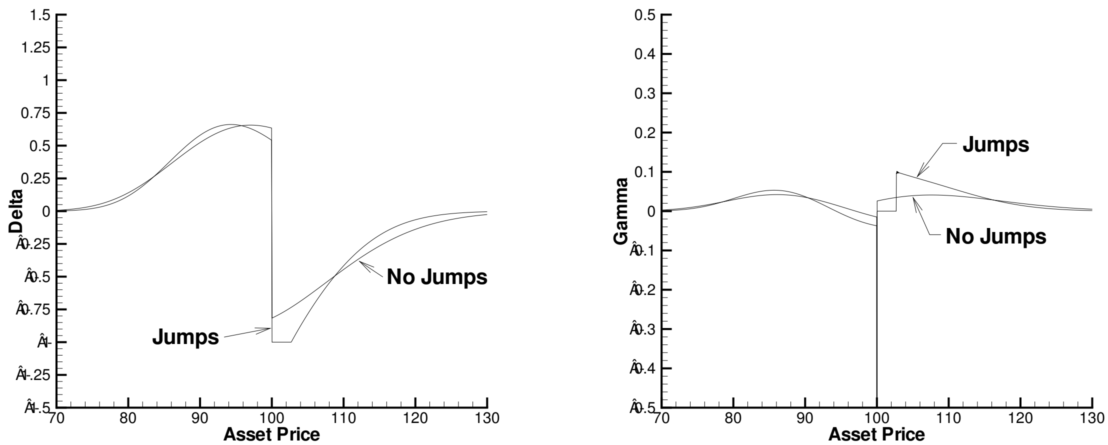

Figure 4: *American butterfly option delta* (V_S)*, gamma (*V_{SS}*), jump diffusion model, compared with model with no jumps. The no jump model has an implied volatility which gives the same price as the jump model for a European option at the money. Data as in Table 1.*

the volatility [33]. However, we have observed that the penalty method for imposing the American constraint appears to be globally (and rapidly) convergent for models with stochastic volatility, but no jumps [33].

A model which includes stochastic volatility as well as jumps in asset price and volatility is thought to be an excellent model of asset price evolution. We conjecture that a suitable generalization of the penalty iteration in this two factor case will also be rapidly convergent, even though the coefficient matrix is not an M-matrix. In addition, the coefficient matrix is no longer tridiagonal (a two dimensional PDE). We will be reporting on results for American option pricing with stochastic volatility and jumps in future work.

References

[1] K. Amin. Jump diffusion option valuation in discrete time. *Journal of Finance*, 48:1833–1863, 1993.

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

[3] E. Ayache, P.A. Forsyth, and K.R. Vetzal. Next generation models for convertible bonds with credit risk. *Wilmott Magazine*, pages 68–77, December 2002.

[4] M. Broadie and Y. Yamamoto. Application of the Fast Gauss transform to option pricing. 2002. working paper, Columbia School of Business.

[5] T.F. Coleman, Y. Li, and A. Verma. Reconstructing the unknown local volatility function. *Journal of Computational Finance*, 2:77–102, 1999.

[6] R.W. Cottle, J.-S. Pang, and R.E. Stone. *The Linear Complementarity Problem.* Academic Press, 1992.

[7] C.W. Cryer. The efficient solution of linear complemetarity problems for tridiagonal Minkowski matrices. *ACM Transactions on Mathematical Software*, 9:199–214, 1983.

[8] Y. d'Halluin, P.A. Forsyth, and K.R. Vetzal. Robust numerical methods for contingent claims under jump diffusion processes. www.scicom.uwaterloo.ca/~paforsyt/jump.pdf, submitted to Review of Financial Studies.

[9] Y. d'Halluin, P.A. Forsyth, K.R. Vetzal, and G. Labahn. A numerical PDE approach for pricing callable bonds. *Applied Mathematical Finance*, 8:49–77, 2001.

[10] A. Dutt and V. Rokhlin. Fast Fourier transforms for nonequispaced data. *SIAM Journal of Scientific Computation*, 14:1368–1393, November 1993.

[11] E.M. Elliot and J.R. Ockendon. *Weak and Variational Methods for Moving Boundary Problems*. Pitman, 1982.

[12] P.A. Forsyth and K.R. Vetzal. Quadratic convergence of a penalty method for valuing American options. *SIAM Journal on Scientific Computation*, 23:2096–2123, 2002.

[13] L. Greengard and J. Strain. The fast Gauss transform. *SIAM Journal on Scientific Computing*, 12:79–94, 1991.

[14] J. Hull. *Options, Futures, and Other Derivatives*. Prentice Hall, Inc., Upper Saddle River, NJ, 3rd edition, 1997.

[15] C. Johnson. *Numerical Solutions of Partial Differential Equations By the Finite Element Method*. Cambridge University Press, Cambridge, 1987.

[16] S. G. Kou. A jump diffusion model for option pricing. *Management Science*, 48:1086–1101, August 2002.

[17] A. Lewis. Fear of jumps. *Wilmott Magazine*, pages 60–67, December 2002.

[18] R.C. Merton. Option pricing when underlying stock returns are discontinuous. *Journal of Financial Economics*, 3:125–144, 1976.

[19] G.H. Meyer. The numerical valuation of options with underlying jumps. *Acta Math. Univ. Comenianae*, 67:69–82, 1998.

[20] S. Mulinacci. An approximation of American option prices in a jump diffusion model. *Stochastic Processes and their Applications*, 62:1–17, 1996.

[21] H. Pham. Optimal stopping of controlled jump diffusion processes: a viscosity solution approach. *Journal of Mathematical Systems, Estimation and Control*, 8:1–27, 1998.

[22] D.M. Pooley, P.A. Forsyth, and K.R. Vetzal. Numerical convergence properties of option pricing PDEs with uncertain volatility. To appear in IMA Journal of Numerical Analysis, 2003.

[23] D. Potts, G. Steidl, and M. Tasche. Fast Fourier transforms for nonequispaced data: A tutorial, 2000. in Modern Sampling Theory: Mathematics and Application, J. J. Benedetto and P. Ferreira, eds., ch. 12, pp. 253 - 274, Birkhauser.

[24] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes: The Art of Scientific Computing.* Cambridge University Press, Cambridge (UK) and New York, 2nd edition, 1992.

[25] R. Rannacher. Finite element solution of diffusion problems with irregular data. *Numerische Mathematik*, 43:309–327, 1984.

[26] D. Tavella and C. Randall. *Pricing financial instruments: the finite difference method.* John Wiley & Sons, Inc, 2000.

[27] A. F. Ware. Fast approximate Fourier transforms for irregularly spaced data. *SIAM Review*, 40:838–856, 1998.

[28] P. Wilmott. *Derivatives.* John Wiley and Sons Ltd, Chichester, 1998.

[29] H. Windcliff, P.A. Forsyth, and K.R. Vetzal. Shout options: a framework for pricing contracts which can be modified by the investor. *Journal of Computational and Applied Mathematics*, 134:213–241, 2001.

[30] H. Windcliff, P.A. Forsyth, and K.R. Vetzal. Valuation of segregated funds: shout options with maturity extensions. *Insurance: Mathematics and Economics*, 29:1–21, 2001.

[31] H. Windcliff, P.A. Forsyth, and K.R. Vetzal. Analysis of the stability of the linear boundary condition for the Black-Scholes equation. 2003. submitted to the Journal of Computational Finance.

[32] X.L. Zhang. Numerical analysis of American option pricing in a jump-diffusion model. *Mathematics of Operations Research*, 22:668–690, 1997.

[33] R. Zvan, P.A. Forsyth, and K.R. Vetzal. Penalty methods for American options with stochastic volatility. *Journal of Computational and Applied Mathematics*, 91:199–218, 1998.

[34] R. Zvan, P.A. Forsyth, and K.R. Vetzal. A finite element approach to the pricing of discrete lookbacks with stochastic volatility. *Applied Mathematical Finance*, 6:87–106, 1999.

[35] R. Zvan, P.A. Forsyth, and K.R. Vetzal. A finite volume approach for contingent claims valuation. *IMA Journal of Numerical Analysis*, 21:703–731, 2001.