Robust Numerical Methods for Contingent Claims under Jump Diffusion Processes

Y. d'Halluin*, P.A. Forsyth†, and K.R. Vetzal‡

March 5, 2003

Abstract

An implicit method is developed for the numerical solution of option pricing models where it is assumed that the underlying process is a jump diffusion. This method can be applied to a variety of contingent claim valuations, including American options, various kinds of exotic options, and models with uncertain volatility or transaction costs. Proofs of timestepping stability and convergence of a fixed point iteration scheme are presented. For typical model parameters, it is shown that the fixed point iteration reduces the error by two orders of magnitude at each iteration. The correlation integral is computed using a fast Fourier transform (FFT) method. Techniques are developed for avoiding wrap-around effects. Numerical tests of convergence for a variety of options are presented.

Keywords: Jump diffusion, implicit discretization, iterative solution

Acknowledgment: This work was supported by the Natural Sciences and Engineering Research Council of Canada, the Social Sciences and Humanities 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

In recent years, there have been significant advances in derivative security pricing, at both the theoretical level and on the empirical side. In terms of theory, the Fourier inversion technique was introduced to the finance literature by Heston (1993) to obtain closed form solutions for the values of vanilla European options in the stochastic volatility context. This approach has been exploited by many authors to derive similar solutions in models where the underlying state variables follow more complicated and typically discontinuous stochastic processes. Examples of this include Bates (1996), who augments the Heston model by assuming that the underlying asset price follows a compound Poisson process with lognormally-distributed jumps, Scott (1997) and Bakshi et al. (1997) who present option valuation models with stochastic volatility, jumps, and stochastic interest rates, and Duffie et al. (2000) who examine a wide variety of models, including generalizations of the above where the volatility of the underlying asset is also described by a jump-diffusion process. The Fourier inversion approach has also been used to obtain closed form solutions for option prices

*Y. d'Halluin is a Ph.D. student with the School of Computer Science at the University of Waterloo, Waterloo ON, Canada N2L 3G1 (e-mail: ydhallui@elora.uwaterloo.ca).

†P.A. Forsyth is a professor with the School of Computer Science at the University of Waterloo, Waterloo ON, Canada N2L 3G1 (e-mail: paforsyt@elora.uwaterloo.ca).

‡K.R. Vetzal is an associate professor with the Centre for Advanced Studies in Finance at the University of Waterloo, Waterloo ON, Canada N2L 3G1 (e-mail: kvetzal@watarts.uwaterloo.ca).

under more general Lévy processes by authors such as Madan et al. (1998), Lewis (2001) and Carr and Wu (2002).

Along with these theoretical advances, there have been numerous empirical studies documenting the importance of jumps. In some cases, the investigation concentrates directly on the underlying state variables (i.e. the "real world" probability measure), whereas other studies use option price data to implicitly estimate parameters of the assumed process for the underlying (i.e. the "risk-neutral" measure). Examples include Jorion (1988) and Bates (1996) in the case of foreign exchange rates, Das (2002) in the case of interest rates, Bakshi and Cao (2002) in the case of options on individual stocks, and, most prominently, a large number of papers in the general area of stock index and stock index futures options. These include Bates (1991, 2000), Bakshi et al. (1997), Pan (2002), Andersen et al. (2002), and Eraker et al. (2003) and references therein.

Much of this research has been motivated by the well-known deficiencies of the benchmark Black-Scholes model for option pricing, i.e. the "volatility smile". One alternative tack, of significant interest to practitioners, is to return to continuous univariate diffusion models but to specify the volatility as a deterministic function of time and the price of the underlying asset. Among several others, the papers by Dupire (1994) and Andersen and Brotherton-Ratcliffe (1998) are particularly useful references. This approach has been criticized because of overfitting and non-stationarity of parameters. Andersen and Andreasen (2000) have recently staked out a middle ground by combining the deterministic volatility approach with lognormally distributed Poisson jumps with constant parameters. They argue that this alleviates many of the concerns, noting that "by letting the jump-part of the process dynamics explain a significant part of the volatility smile/skew, we generally obtain a 'reasonable', stable [deterministic volatility] function, without the extreme short-term variation typical of the pure diffusion approach" (Andersen and Andreasen, 2000, p. 233).

Although this is an impressive array of analytical results and empirical evidence regarding the importance of jumps, it is mostly confined to plain vanilla European style options (and options on futures). A large amount of the empirical evidence is based on S&P 500 index options, which are European style. It is also worth noting that the study by Bakshi and Cao (2002) on individual stocks considers only short-term out-of-the-money options. As these are quite unlikely to be exercised early, analytic European option valuation formulas can be used. An important exception to this focus on European style instruments is the work of Bates (1991, 1996, 2000), who uses an extension of the Barone-Adesi and Whaley (1987) analytic approximation for American option values in the Black-Scholes lognormal diffusion context. However, this approximation was derived only in the constant volatility case with lognormal jumps. While it appears to be reasonably accurate in the cases considered by Bates, it may not be for other parameter values or for the case of long-dated options.[1]

The preceding discussion indicates that more work is needed on numerical approaches in models where the underlying variables follow discontinuous processes. In addition to American style options (with finite maturities), numerical methods are required to value most exotic or path-dependent options of practical significance (e.g. discretely observed barrier, lookback, and Asian options). Numerical techniques are also required when jumps are combined with non-constant local volatilities to calibrate models to observed prices of European options, as in the model of Andersen and Andreasen (2000).

In general, the valuation of a contingent claim under a jump diffusion process requires solving a partial integro-differential equation (PIDE). Very little work has been carried out for numerically pricing options under jump diffusion processes. The method suggested by Amin (1993) is an explicit type approach based on multinomial trees. As is well-known, such methods have timestep limitations due to stability considerations, and are generally only first order correct. Zhang (1997) develops a method which treats the jump integral term in explicit fashion, and the remaining terms in the PIDE implicitly. Unfortunately, rather re-

[1] Barone-Adesi and Whaley (1987) report that the accuracy of the approximation in the Black-Scholes setting deteriorates for options with maturities longer than one year.

strictive stability conditions are required. Meyer (1998) uses the method of lines to value American options where the underlying asset can jump to a finite number of states. More recently, a method based on use of a wavelet transform has been suggested by Matache et al. (2002). The basic idea is to use a wavelet transform to approximate the dense matrix discrete integral operator by dropping small terms. Andersen and Andreasen (2000) used an operator splitting type of approach combined with a fast Fourier transform (FFT) evaluation of a convolution integral to price European options with jump diffusion, where the diffusion terms involved non-constant local volatilities. However, an operator splitting approach cannot easily handle American options or nonlinear option valuation models (e.g. transaction costs or uncertain parameters, as discussed in Wilmott (1998) and references provided there). Andersen and Andreasen (2000) also report results for Bermudan put, forward start, and Asian options.[2] Longstaff and Schwartz (2001) show that their least squares Monte Carlo technique can be used to value Bermudan put options. The use of Monte Carlo techniques for barrier options under jump diffusion has been explored by Metwally and Atiya (2002). However, for low dimensional valuation problems, Monte Carlo methods are much slower than available alternatives.

In this paper, we develop a different approach. Our technique is similar in some respects to Zhang (1997), though less constrained in terms of stability restrictions. Our method also offers a higher rate of convergence than Zhang's. Similar comments apply if we compare our approach to that of Andersen and Andreasen (2000). For some simple cases, their approach might be slightly more efficient than ours, but we offer a more general purpose method which is capable of handling a much wider array of contracts. We confine our attention to the relatively simple case of compound Poisson jump diffusion processes for a single underlying stochastic variable, deferring the treatment of the more complicated cases of general Lévy processes and multiple state variables to future work. As in Andersen and Andreasen (2000), we do not assume constant coefficients for the diffusion part of the process. We prove that the jump diffusion term can be discretized explicitly, and, when coupled with an implicit treatment of the usual PDE, the resulting timestepping method is unconditionally stable. However, in order to achieve second order convergence, an implicit treatment of the jump integral term is preferred. We prove that a simple fixed point iteration scheme can be used to solve the discretized algebraic equations, and that this iteration is globally convergent. In fact, for typical values of the timestep size and Poisson arrival intensity, the error is reduced by two orders of magnitude at each iteration.

We also develop a method for efficiently computing the jump integral term. We make no assumptions about the probability density for the jump term. This general approach requires the evaluation of correlation type integrals, as in Zhang (1997), compared to the convolution integral which is common in the literature. The correlation integral term can be rapidly computed using FFT methods. In contrast with previous work, we do not assume that the grid is equally spaced in either the underlying asset price or its logarithm. This is a major advantage for the pricing of contracts with barrier provisions, which typically require a fine grid spacing near barriers in order to achieve sufficient accuracy. We also show how to eliminate the wrap-around effects which often plague FFT methods.

A major advantage of the method developed here is that it is straightforward to add a jump process to existing option pricing software. In particular, existing software that uses an implicit approach for valuing American options can be simply modified to price American options with jump diffusion. A variety of exotic and path-dependent contracts can be handled in a straightforward way, and nonlinear models such as transaction costs or uncertain parameters can also be easily extended to the jump diffusion case.

The remainder of the paper is structured as follows. Section 2 outlines the underlying model and notation. Section 3 describes the implicit discretization method and its stability properties. Section 4 considers the Crank-Nicolson case, which can be used to achieve higher rates of convergence. The fixed point itera-

[2] Andersen and Andreasen (2000) use Monte Carlo methods rather than their operator splitting approach for forward start and Asian options.

tion technique is discussed in Section 5. A detailed analysis of some of the subtle numerical issues involved is presented in Section 6. Section 7 presents a selection of illustrative results, including cases of European options, American options, digital options, alternative distributions for the proportional jump size, and contracts with barrier features. Section 8 provides a final summary.

2 The Basic Model

This section presents the model for the evolution of the price of the underlying asset and the general form of the PIDE to be solved for option valuation. Let S represent the underlying stock price. Movements in this variable over time are assumed to be described by a jump diffusion process of the form

$$\frac{dS}{S} = \nu dt + \sigma dz + (\eta - 1)dq, \tag{2.1}$$

where ν is the drift rate, σ is the volatility associated with the continuous (Brownian) component of the process, dz is the increment of a Gauss-Wiener process, dq is a Poisson process which is assumed to be independent of the Brownian part (note that $dq = 0$ with probability $1 - \lambda dt$ and $dq = 1$ with probability λdt, where λ is the Poisson arrival intensity), and $\eta - 1$ is an impulse function producing a jump from S to $S\eta$. We denote the expected relative jump size by $\kappa = E(\eta - 1)$. For ease of notation, we have made two simplifications in equation (2.1): (i) we have ignored dividends (it is straightforward to add a continuous dividend yield to the underlying process, or to handle a discrete dividend in the numerical algorithm); and (ii) we have specified σ as a constant, although it can be a deterministic function of S and t.

Under equation (2.1), the stock price S has two sources of uncertainty. The term σdz corresponds to normal levels of uncertainty while the term dq describes exceptional events. If the Poisson event does not occur ($dq = 0$), then equation (2.1) is equivalent to the usual stochastic process of geometric Brownian motion assumed in the Black-Scholes model (with the additional assumption that σ is constant). If, on the other hand, the Poisson event occurs, then equation (2.1) can be written as

$$\frac{dS}{S} \simeq (\eta - 1), \tag{2.2}$$

where $\eta - 1$ is an impulse function producing a jump from S to $S\eta$. Consequently, the resulting sample path for the stock S will be continuous most of the time with finite negative or positive jumps with various amplitudes occurring at discrete points in time.

Let $V(S,t)$ be the value of a contingent claim that depends on the underlying stock price S and time t. As is well-known, the following backward PIDE may be solved to determine V:

$$V_\tau = \frac{1}{2}\sigma^2 S^2 V_{SS} + (r - \lambda\kappa)SV_S - rV + \left(\lambda \int_0^\infty V(S\eta)g(\eta)d\eta - \lambda V\right), \tag{2.3}$$

where $\tau = T - t$ is the time until expiry at date T, r is the continuously compounded risk free interest rate, and $g(\eta)$ is the probability density function of the jump amplitude η with the obvious properties that $\forall\eta$, $g(\eta) \geq 0$ and $\int_0^\infty g(\eta)d\eta = 1$. An important special case is where σ is constant and the jump size distribution is lognormal, this being the well-known model of Merton (1976). For brevity, the details of the derivation of equation (2.3) have been omitted (for further details, see Merton, 1976; Wilmott, 1998; Andersen and Andreasen, 2000, among others).[3] For future convenience, note that equation (2.3) can be rewritten in

[3]It is important to note that we have followed Merton (1976) and assumed that jump risk is diversifiable. This is obviously not suitable in all contexts, the obvious exception being that where the underlying asset is a stock index. In this case, as noted by Naik and Lee (1990) and Bates (1991), one needs to construct a general equilibrium model. Under suitable assumptions on preferences, a PIDE of the same form as (2.3) can be obtained, as in Bates (1991). As our primary interest is the numerical algorithm, we do not pursue this issue in any further detail here.

slightly different form as

$$V_\tau = \frac{1}{2}\sigma^2 S^2 V_{SS} + (r - \lambda\kappa)SV_S - (r+\lambda)V + \lambda \int_0^\infty V(S\eta)g(\eta)d\eta. \tag{2.4}$$

3 Implicit Discretization Methods

This section explores discretization methods for the PIDE, where the terms not involving the jump integral are handled implicitly. A straightforward approach to the numerical solution of equation (2.4) would be to use standard numerical discretization methods for the non-integral terms (as described, for example in Tavella and Randall, 2000), in combination with numerical integration methods such as Simpson's rule or Gaussian quadrature. However, such an approach is computationally expensive, as noted by Tavella and Randall. It is more efficient to transform the integral in equation (2.4) into a correlation integral. This allows efficient FFT methods to be used to evaluate the integral for all values of S.

Let

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

Setting $x = \log(S)$ and using the change of variable $y = \log(\eta)$ gives

$$I = \int_{-\infty}^{\infty} \overline{V}(x+y)\overline{f}(y)dy. \tag{3.2}$$

where $\overline{f}(y) = g(e^y)e^y$ and $\overline{V}(y) = V(e^y)$. Note that $\overline{f}(y)$ is the probability density of a jump of size $y = \log(\eta)$. Equation (3.2) corresponds to the correlation product $\otimes$ of $\overline{V}(y)$ and $\overline{f}(y)$, so we can write (3.2) more succinctly as

$$I = \overline{V} \otimes \overline{f}. \tag{3.3}$$

If $\overline{f}$ is an even function (i.e. $\overline{f}(x) = \overline{f}(-x)$), then (3.3) corresponds to the convolution product.

We can write the correlation integral (3.2) in discrete form as

$$I_i = \sum_{j=-N/2+1}^{j=N/2} \overline{V}_{i+j}\overline{f}_j \Delta y + O\left((\Delta y)^2\right), \tag{3.4}$$

where $I_i = I(i\Delta x)$, $\overline{V}_j = \overline{V}(j\Delta x)$,

$$\overline{f}_j = \int_{x_j - \Delta x/2}^{x_j + \Delta x/2} \overline{f}(x)dx, \tag{3.5}$$

and $x_j = j\Delta x$. Note that we have assumed that $\Delta y = \Delta x$, and that in (3.4) 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 $\overline{V}_{N/2+j}, j > 0$ can be approximated by an asymptotic boundary condition. In practice, since $\overline{f}_j$ decays rapidly for $|j| > 0$, this should not cause any difficulty. Also note that $\overline{V}_{-N/2+j}, j < 0$, can be interpolated from known values V_k, since these points represent values near $S = 0$. Since it will be used in subsequent sections, an important property to note is that

$$\begin{gathered} \overline{f}_j \geq 0\,, \quad \forall j \\ \sum_{j=-N/2+1}^{j=N/2} \overline{f}_j \Delta y \leq 1. \end{gathered} \tag{3.6}$$

This follows because $\overline{f}(y)$ is a probability density function and $\overline{f}_j$ is defined by equation (3.5).

The discrete form of the correlation integral (3.4) uses an equally spaced grid in $\log S$ coordinates. While this is convenient for a FFT evaluation of the correlation integral, it is not particularly suitable for discretizing the PDE. We will use an unequally spaced grid in S coordinates for the PDE discretization $[S_0, \ldots, S_p]$. Let

$$V_i^n = V(S_i, \tau_n). \tag{3.7}$$

Now, $\overline{V}_j$ will not necessarily coincide with any of the discrete values V_k in equation (3.7). Consequently, we will linearly interpolate (using Lagrange basis functions defined on the S grid) to determine the appropriate values, i.e. if

$$S_{\Upsilon(j)} \le e^{j\Delta x} \le S_{\Upsilon(j)+1}, \tag{3.8}$$

then

$$\overline{V}_j = \psi_{\Upsilon(j)} V_{\Upsilon(j)} + \left(1 - \psi_{\Upsilon(j)}\right) V_{\Upsilon(j)+1} + O\left((\Delta S_{\Upsilon(j)+1/2})^2\right), \tag{3.9}$$

where $\psi_{\Upsilon(j)}$ is an interpolation weight, and $\Delta S_{i+1/2} = S_{i+1} - S_i$. 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 simply linearly interpolate the I_i to get the desired value. If $e^{x_{\Pi(k)}} \le S_k \le e^{x_{\Pi(k)+1}}$, then

$$I(S_k) = \phi_{\Pi(k)} I_{\Pi(k)} + (1 - \phi_{\Pi(k)}) I_{\Pi(k)+1} + O\left((e^{x_{\Pi(k)}} - e^{x_{\Pi(k)+1}})^2\right), \tag{3.10}$$

where $\phi_{\Pi(k)}$ is an interpolation weight. Note that

$$\begin{aligned} 0 \le \phi_i \le 1 \\ 0 \le \psi_i \le 1. \end{aligned} \tag{3.11}$$

Combining equations (3.4), (3.9), and (3.10) gives

$$I(S_k) = \sum_{j=-N/2+1}^{j=N/2} \chi(V, k, j) \overline{f}_j \Delta y, \tag{3.12}$$

where $V = [V_0, V_1, \ldots, V_p]'$ and

$$\begin{aligned} \chi(V, k, j) = {} & \phi_{\Pi(k)} \left[\psi_{\Upsilon(\Pi(k)+j)} V_{\Upsilon(\Pi(k)+j)} + (1 - \psi_{\Upsilon(\Pi(k)+j)}) V_{\Upsilon(\Pi(k)+j)+1}\right] \\ & + (1 - \phi_{\Pi(k)}) \left[\psi_{\Upsilon(\Pi(k)+1+j)} V_{\Upsilon(\Pi(k)+1+j)} + (1 - \psi_{\Upsilon(\Pi(k)+1+j)}) V_{\Upsilon(\Pi(k)+1+j)+1}\right]. \end{aligned} \tag{3.13}$$

For future reference, note that $\chi(V, k, j)$ is linear in V, and that if $\iota = [1, 1, \ldots, 1]'$, then it follows from properties (3.11) that

$$\chi(\iota, k, j) = 1 \quad \forall k, j. \tag{3.14}$$

We can now consider the complete discretization of equation (2.4). The integral term is approximated using equation (3.12). We use a fully implicit method for the usual PDE, and then use a weighted timestepping method for the jump integral term. Letting V_i^n denote the solution at node i and time level n, the discrete equations can be written as

$$\begin{aligned} & V_i^{n+1} \left[1 + (\alpha_i + \beta_i + r + \lambda)\Delta\tau\right] - \Delta\tau\beta_i V_{i+1}^{n+1} - \Delta\tau\alpha_i V_{i-1}^{n+1} \\ & \qquad = V_i^n + (1 - \theta_J)\Delta\tau\lambda \sum_{j=-N/2+1}^{j=N/2} \chi(V^{n+1}, i, j) \overline{f}_j \Delta y + \theta_J \Delta\tau\lambda \sum_{j=-N/2+1}^{j=N/2} \chi(V^n, i, j) \overline{f}_j \Delta y. \end{aligned} \tag{3.15}$$

Note that $\theta_J = 0$ corresponds to an implicit handling of the jump integral, whereas $\theta_J = 1$ indicates an explicit treatment of this term.

Discretizing the first derivative term of equation (2.4) with central differences leads to

$$\alpha_{i,central} = \frac{\sigma_i^2 S_i^2}{(S_i - S_{i-1})(S_{i+1} - S_{i-1})} - \frac{(r - \lambda\kappa)S_i}{S_{i+1} - S_{i-1}}$$
$$\beta_{i,central} = \frac{\sigma_i^2 S_i^2}{(S_{i+1} - S_i)(S_{i+1} - S_{i-1})} + \frac{(r - \lambda\kappa)S_i}{S_{i+1} - S_{i-1}}. \tag{3.16}$$

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

$$\alpha_{i,forward} = \frac{\sigma_i^2 S_i^2}{(S_i - S_{i-1})(S_{i+1} - S_{i-1})}$$
$$\beta_{i,forward} = \frac{\sigma_i^2 S_i^2}{(S_{i+1} - S_i)(S_{i+1} - S_{i-1})} + \frac{(r - \lambda\kappa)S_i}{S_{i+1} - S_i}, \tag{3.17}$$

or (backward difference)

$$\alpha_{i,backward} = \frac{\sigma_i^2 S_i^2}{(S_i - S_{i-1})(S_{i+1} - S_{i-1})} - \frac{(r - \lambda\kappa)S_i}{S_{i+1} - S_i}$$
$$\beta_{i,backward} = \frac{\sigma_i^2 S_i^2}{(S_{i+1} - S_i)(S_{i+1} - S_{i-1})}. \tag{3.18}$$

Algorithmically, we decide between a central or forward discretization at each node for equation (3.15) as follows:

$$\begin{array}{l}
\text{If } [\alpha_{i,central} \geq 0 \text{ and } \beta_{i,central} \geq 0] \text{ then} \\
\qquad \alpha_i = \alpha_{i,central} \\
\qquad \beta_i = \beta_{i,central} \\
\text{ElseIf } [\beta_{i,forward} \geq 0] \text{ then} \\
\qquad \alpha_i = \alpha_{i,forward} \\
\qquad \beta_i = \beta_{i,forward} \\
\text{Else} \\
\qquad \alpha_i = \alpha_{i,backward} \\
\qquad \beta_i = \beta_{i,backward} \\
\text{EndIf}
\end{array} \tag{3.19}$$

Note that the test condition (3.19) guarantees that α_i and β_i are non-negative. For typical parameter values and grid spacing, forward or backward 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. As we shall see, requiring that all α_i and β_i are non-negative has important theoretical ramifications.

As $S \to 0$, equation (2.3) reduces to $V_\tau = -rV$, which is simply incorporated into the discrete equations (3.15) 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 done by replacing equation (3.15) at $S = S_{\max} = S_p$ with the specification that V_p^{n+1} is equal to the relevant Dirichlet condition.

We now proceed to consider the stability of the discretization (3.15). In particular, we have the following result:

Theorem 3.1 (Stability of scheme (3.15)). *The discretization method* (3.15) *is unconditionally stable for any choice of* $\theta_J, 0 \leq \theta_J \leq 1$*, provided that*

- $\alpha_i, \beta_i \geq 0$*;*
- *the discrete probability density* $\overline{f}_j$ *has the properties* (3.6)*;*
- *the interpolation weights satisfy* (3.11)*;*
- $r, \lambda \geq 0$.

Proof. Let $V^n = [V_0^n, V_1^n, \ldots, V_p^n]'$ be the discrete solution vector to equation (3.15). Suppose the initial solution vector is perturbed, i.e.

$$\hat{V}^0 = V^0 + E^0, \tag{3.20}$$

where $E^n = [E_0^n, \ldots, E_p^n]'$ is the perturbation vector. Note that $E_p^n = 0$ since Dirichlet boundary conditions are imposed at this node. Then we obtain the following equation for the propagation of the perturbation (noting that χ is a linear operator)

$$\begin{aligned} E_i^{n+1}[1 + (\alpha_i + \beta_i + r + \lambda)\Delta\tau] - \Delta\tau\beta_i E_{i+1}^{n+1} - \Delta\tau\alpha_i E_{i-1}^{n+1} \\ = E_i^n + (1-\theta_J)\Delta\tau\lambda \sum_{j=-N/2+1}^{j=N/2} \chi(E^{n+1}, i, j)\overline{f}_j \Delta y + \theta_J \Delta\tau\lambda \sum_{j=-N/2+1}^{j=N/2} \chi(E^n, i, j)\overline{f}_j \Delta y. \end{aligned} \tag{3.21}$$

Defining

$$\|E\|^n = \max_i |E_i|^n, \tag{3.22}$$

it follows from properties (3.6), (3.11), and (3.14) and $\alpha_i, \beta_i \geq 0$ that

$$\begin{aligned} |E_i^{n+1}|[1 + (\alpha_i + \beta_i + r + \lambda)\Delta\tau] \leq \|E\|^n + (1-\theta_J)\Delta\tau\lambda\|E\|^{n+1} + \theta_J\Delta\tau\lambda\|E\|^n \\ + \Delta\tau\beta_i|E_{i+1}^{n+1}| + \Delta\tau\alpha_i|E_{i-1}^{n+1}|. \end{aligned} \tag{3.23}$$

This implies

$$|E_i^{n+1}|[1 + (\alpha_i + \beta_i + r + \lambda)\Delta\tau] \leq (\Delta\tau\beta_i + \Delta\tau\alpha_i)\|E\|^{n+1} \tag{3.24}$$

$$+ \|E\|^n + (1-\theta_J)\Delta\tau\lambda\|E\|^{n+1} + \theta_J\Delta\tau\lambda\|E\|^n. \tag{3.25}$$

Now, equation (3.25) is valid for all $i < p$. In particular, it is true for node i^*, where

$$\max_i |E_i^{n+1}| = |E_{i^*}^{n+1}|. \tag{3.26}$$

Writing equation (3.25) for $i = i^*$ gives

$$\|E\|^{n+1}[1 + (r + \theta_J\lambda)\Delta\tau] = \|E\|^n(1 + \theta_J\Delta\tau\lambda), \tag{3.27}$$

and thus

$$\begin{aligned} \|E\|^{n+1} &\leq \|E\|^n \frac{(1 + \theta_J\Delta\tau\lambda)}{(1 + (r + \theta_J\lambda)\Delta\tau)} \\ &\leq 1. \quad \blacksquare \end{aligned} \tag{3.28}$$

This result is somewhat surprising, since we can discretize the correlation integral term explicitly ($\theta_J = 1$), yet scheme (3.15) remains unconditionally stable. Note that Zhang (1997) derived a conditionally stable method. The conditional stability was a result of a slightly different timestepping approach compared to that in equation (3.15).

4 Crank-Nicolson Discretization

The discretization method used in the previous section is only first order correct in the time direction. In order to improve the timestepping error, we can use a Crank-Nicolson method. Such an approach results in the following set of discrete equations

$$V_i^{n+1}\left[1+(\alpha_i+\beta_i+r+\lambda)\frac{\Delta\tau}{2}\right]-\frac{\Delta\tau}{2}\beta_i V_{i+1}^{n+1}-\frac{\Delta\tau}{2}\alpha_i V_{i-1}^{n+1}$$
$$=V_i^n\left[1-(\alpha_i+\beta_i+r+\lambda)\frac{\Delta\tau}{2}\right]+(1-\theta_J)\lambda\Delta\tau\sum_{j=-N/2+1}^{j=N/2}\chi(V^{n+1},i,j)\overline{f}_j\Delta y$$
$$+\theta_J\lambda\Delta\tau\sum_{j=-N/2+1}^{j=N/2}\chi(V^n,i,j)\overline{f}_j\Delta y. \quad (4.1)$$

A full Crank-Nicolson method is obtained by setting $\theta_J = 1/2$ in equation (4.1). If we define the matrix M such that

$$-[MV^n]_i = V_i^n(\alpha_i+\beta_i+r+\lambda)\frac{\Delta\tau}{2}-\frac{\Delta\tau}{2}\beta_i V_{i+1}^n-\frac{\Delta\tau}{2}\alpha_i V_{i-1}^n-\frac{\Delta\tau}{2}\lambda\sum_{j=-N/2+1}^{j=N/2}\chi(V^n,i,j)\overline{f}_j\Delta y, \quad (4.2)$$

then we can write equation (4.1) as

$$[I-M]V^{n+1}=[I+M]V^n. \quad (4.3)$$

Alternatively, we can define $B=[I-M]^{-1}[I+M]$, so that equation (4.3) can be written as

$$V^n = B^nV^0. \quad (4.4)$$

Consequently, an initial perturbation vector E^0 will generate a perturbation at the nth step, E^n, given by $E^n = B^nE^0$.

The stability of the operator B is defined in terms of the power boundedness of B. If n is the number of timesteps and p is the number of grid nodes, then given some matrix norm $\|\cdot\|$, we say that B is *strictly stable* if

$$\|B^n\|\le 1 \quad \forall n,p. \quad (4.5)$$

Following Giles (1997), *strong stability* is defined as

$$\|B^n\|\le C \quad \forall n,p, \quad (4.6)$$

and *algebraic stability* is defined as

$$\|B^n\|\le Cn^sp^q \quad \forall n,m. \quad (4.7)$$

where $C,s,q\ge 0$ are constants independent of n and p.

Algebraic stability is obviously a weaker condition than either strict or strong stability. Note that the Lax Equivalence Theorem states that strong stability is a necessary and sufficient condition for convergence for all initial data. Weaker algebraic stability yields convergence only for certain initial data. For a more detailed discussion of this, see Giles (1997).

If μ_i are the eigenvalues of B, then a necessary condition for strong stability is that $|\mu_i|\le 1$, and that any $|\mu_i|=1$ has multiplicity one. From equation (4.2) and properties (3.6), we have that

- The off-diagonals of M are all non-negative.

- The diagonals of M (excluding the last row) are strictly negative.
- Assuming that $r > 0$, $\sum_{j=0}^{j=p} M_{ij} < 0$ for $i = 0, \ldots, p-1$.
- The last row of M is identically zero due to the Dirichlet boundary condition.

It then follows that all the Gerschgorin disks of M are strictly contained in the left half of the complex plane, with one eigenvalue identically zero. Hence all the eigenvalues of B are strictly less than one in magnitude, with one eigenvalue having modulus one. As a result, B satisfies the necessary conditions for strict stability.

However, since B is non-symmetric, this is not sufficient for power boundedness of B (see Borovykh and Spijker, 2000, for a counterexample). As discussed in Kraaijevanger et al. (1987) and Lenferink and Spijker (1991), we can guarantee algebraic stability by examining the γ numerical range of the matrix M. In the case $\gamma = 1$, the numerical range of M coincides with the convex hull of the Gerschgorin disks of M when the maximum norm is used in equation (4.7). These results can be summarized in the following theorem:

Theorem 4.1 (Algebraic Stability of Crank-Nicolson Timestepping). *The Crank-Nicolson discretization* (4.1) *is algebraically stable in the sense that*

$$\|B^n\|_\infty \leq Cn^{1/2} \quad \forall n, p,$$

where C is independent of n, p.

Proof. Since all the Gerschgorin disks of M are in the left half of the complex plane, this follows from the results in Lenferink and Spijker (1991). ∎

In fact, we believe that the algebraic stability estimate is overly pessimistic. For the case of constant coefficients with a log-spaced grid, in Appendix A we show using Von Neumann analysis that Crank-Nicolson timestepping with the correlation product is unconditionally strictly stable. However, it is interesting to note that if we use Crank-Nicolson weighting for the PDE terms and an explicit method for the jump diffusion term ($\theta_J = 1$ in equation (4.1)), then a Von Neumann analysis shows that this method is only conditionally stable ($\lambda\Delta\tau$ must be sufficiently small).

5 Fixed Point Iteration Method

When using an implicit discretization, it is computationally inefficient to solve the full linear system because the correlation product term makes the system dense. Consequently, in this section we will explore the use of a fixed point iteration to solve the linear system which results from an implicit discretization of the correlation product term. This idea was suggested in Tavella and Randall (2000), but no convergence analysis was given.

Define the matrix $\hat{M}$ such that

$$-[\hat{M}V^n]_i = V_i^n(\alpha_i + \beta_i + r + \lambda)\Delta\tau - \Delta\tau\beta_i V_{i+1}^n - \Delta\tau\alpha_i V_{i-1}^n, \tag{5.1}$$

and the vector $\Omega(V^n)$ such that

$$[\Omega(V^n)]_i = \sum_{j=-N/2+1}^{j=N/2} \chi(V^n, i, j)\overline{f}_j \Delta y. \tag{5.2}$$

Note that $\Omega(V^n)$ is a linear function of V^n. Then we can write a fully implicit ($\theta = 0$) or Crank Nicolson ($\theta = 1/2$) discretization as

$$[I - (1-\theta)\hat{M}]V^{n+1} = [I + \theta\hat{M}]V^n + (1-\theta)\lambda\Delta\tau\Omega(V^{n+1}) + \theta\lambda\Delta\tau\Omega(V^n). \tag{5.3}$$

We can then derive the fixed point iteration method

Fixed Point Iteration

$$
\begin{aligned}
&\text{Let } (V^{n+1})^0 = V^n \\
&\text{Let } \hat{V}^k = (V^{n+1})^k \\
&\text{For } k = 0, 1, 2, \ldots \text{ until convergence} \\
&\qquad \text{Solve} \\
&\qquad \left[I - (1-\theta)\hat{M}\right] \hat{V}^{k+1} \\
&\qquad = \left[I + \theta\hat{M}\right] V^n \\
&\qquad + (1-\theta)\lambda\Delta\tau\Omega(\hat{V}^k) + \theta\lambda\Delta\tau\Omega(V^n) \\
&\qquad \text{If } \max_i \frac{|\hat{V}_i^{k+1} - \hat{V}_i^k|}{\max(1, |\hat{V}_i^{k+1}|)} < \textit{tolerance} \text{ then quit} \\
&\text{EndFor}
\end{aligned}
\tag{5.4}
$$

Letting $e^k = V^{n+1} - \hat{V}^k$, the convergence of the fixed point scheme can be summarized in the following theorem:

Theorem 5.1 (Convergence of the fixed point iteration). *Provided that*

- $\alpha_i, \beta_i \geq 0$ *(see Section 3);*
- *the discrete probability density* $\overline{f}_j$ *has the properties* (3.6)*;*
- *the interpolation weights satisfy* (3.11)*;*
- $r > 0, \lambda \geq 0$*;*

then the fixed point iteration (5.4) *is globally convergent, and the maximum error at each iteration satisfies*

$$\|e^{k+1}\|_\infty \leq \|e^k\|_\infty \frac{(1-\theta)\lambda\Delta\tau}{1 + (1-\theta)(r+\lambda)\Delta\tau}.$$

Proof. It is easily seen from iteration (5.4) that e^k satisfies

$$\left[I - (1-\theta)\hat{M}\right] e^{k+1} = (1-\theta)\lambda\Delta\tau\Omega(e^k). \tag{5.5}$$

Following the same steps used to prove Theorem 3.1, we therefore obtain

$$
\begin{aligned}
\|e^{k+1}\|_\infty &\leq \|e^k\|_\infty \frac{(1-\theta)\lambda\Delta\tau}{1 + (1-\theta)(r+\lambda)\Delta\tau} \\
&< 1. \quad \blacksquare
\end{aligned}
\tag{5.6}
$$

Note that typically $\lambda\Delta\tau \ll 1$, so that

$$\|e^{k+1}\|_\infty \simeq \|e^k\|_\infty (1-\theta)\lambda\Delta\tau, \tag{5.7}$$

which will result in rapid convergence. It is also interesting to observe that the number of iterations required for convergence is independent of the number of nodes in the S grid.

6 More Details Regarding Numerical Issues

To complete the discussion of our numerical algorithm, we need to consider issues such as evaluating the jump integral term, interpolation, wrap around effects, and how to incorporate early exercise features. This section discusses these issues in the particular context of the model of Merton (1976) with log-normally distributed jumps. As we have shown, it is possible to write the PIDE as a PDE and a correlation product (equation (3.3)). Note that each iteration of the scheme (5.4) requires evaluation of this correlation integral for all points on the PDE grid.

Fast evaluation of this integral using FFT methods necessitates transformation to an equally spaced grid in $x = \log(S)$ coordinates. If the original PDE grid is equally spaced in $\log(S)$, then there is clearly no difficulty. However, this type of grid spacing is highly inefficient for cases involving discontinuous payoffs or barriers. We therefore prefer not to restrict the type of grid used for the original PDE. Recall that the correlation integral is

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

or, in discrete form

$$I_i = \sum_{j=-N/2+1}^{j=N/2} \overline{V}_{i+j}\overline{f}_j \Delta y + O\left((\Delta y)^2\right),$$

where $I_i = I(i\Delta x)$, $\overline{V}_j = \overline{V}(j\Delta x)$, $\overline{f}_j = \overline{f}(j\Delta y)$. We have also assumed that $\Delta y = \Delta x$, and that $\overline{V}(\log S) = V(S)$.

Now, $\overline{V}_j$ will not necessarily coincide with any of the discrete values V_k in equation (3.15). Consequently, we will linearly interpolate to determine the appropriate values, as in equation (3.9). Since equation (3.4) has the form of a discrete correlation, FFT methods are an obvious choice to compute this efficiently. Assuming that $\overline{f}$ is real, then

$$\mathrm{FFT}(I)_k = (\mathrm{FFT}(\overline{V}))_k (\mathrm{FFT}(\overline{f}))^*_k, \tag{6.1}$$

where $(\cdot)^*$ denotes the complex conjugate. Since $\overline{f}(z)$ is the probability density of $z = \log\eta$, which is a specified function, we can simply precompute $\mathrm{FFT}(\overline{f})$ on the required equally spaced grid in z coordinates. We can then carry out an inverse FFT to obtain the values of the correlation integral on the equally spaced $x = \log S$ grid. A further interpolation step is required to obtain the value of the correlation integral on the original S grid (equation (3.10)).

We can summarize the steps needed to generate the required values $I(S_k)$, $k = 0, \ldots, p$ as follows:

- Interpolate the discrete values of V onto an equally spaced $\log(S)$ grid. This generates the required values of $\overline{V}_j$.
- Carry out the FFT on this data.
- Compute the correlation in the frequency domain (with precomputed $FFT\left(\overline{f}\right)$), using equation (6.1).
- Invert the FFT of the correlation.
- Interpolate the discrete values of $I(x_i)$ onto the original S grid.

(6.2)

Note that as long as linear or higher order interpolation is used, this procedure is second order correct, which is consistent with the discretization error in the PDE and the midpoint rule used to evaluate the integral (3.4).

In principle, we can avoid the interpolation steps in the above procedure if we use special techniques for computing the FFT for unequally spaced data. There are several methods for computing the inverse FFT problem (i.e. given unequally spaced data, determine the Fourier coefficients), as well as the forward FFT problem (given the Fourier coefficients, determine the inverse transform values on an unequally spaced grid) (see, for example Ware, 1998; Duijndam and Schonewille, 1999; Potts et al., 2001). However, it should be noted that we are not particularly interested in obtaining highly accurate estimates of the discrete Fourier coefficients, as we simply need to evaluate the correlation integral correct to second order. Consequently, for our purposes there is no particular benefit in terms of accuracy in using these methods. We will use the interpolation method (6.2) followed by the standard FFT to calculate our illustrative results below in Section 7.

Another issue requiring attention is that the FFT algorithm effectively assumes that the input functions are periodic. This may cause "wrap around pollution" unless special care is taken when implementing the algorithm. The integral (3.2) is approximated on the finite domain

$$I(x) = \int_{y_{\min}}^{y_{\max}} \overline{V}(x+y)\overline{f}(y)dy. \tag{6.3}$$

The PDE part of the PIDE (2.4) is computed using the finite computational domain $[0, S_{\max}]$, using the discrete grid $S_0, S_1, \ldots, S_p$. Initially, we chose

$$\begin{aligned} y_{\max} &= \log(S_{\max}) \\ y_{\min} &= \log(S_1), \end{aligned} \tag{6.4}$$

assuming $S_1 > 0$. Note that $y_{\min} = \log(S_1)$ since normally $S_0 = 0$, so that $\log(S_0) = -\infty$.

Generally, $\overline{f}(y)$ which represents the probability density of a jump of $S \to S\eta$ (where $y = \log\eta$) is rapidly decaying for $|y| \gg 0$. However, $\overline{V}(y)$ does not decay to zero near $y = y_{\min}, y_{\max}$. Typically, $V(S) \leq$ Const. S as $S \to \infty$, and $V(S) \simeq$ Const. as $S \to 0$, or in $y = \log S$ coordinates,

$$\overline{V}(y) \leq \text{Const. } e^y, \quad y \to \infty \tag{6.5}$$

$$\simeq \text{Const.}, \quad y \to -\infty. \tag{6.6}$$

This will cause undesirable wrap around effects if we use an FFT approach to evaluate the integral (6.3), since the discrete Fourier transform (DFT) is effectively applied to the periodic extension of the input functions. To avoid these problems, we will extend the domain of the integral to the left and right, by a size which reflects the width of the probability density. In other words, we will actually evaluate

$$I_{\text{ext}}(x) = \int_{y_{\min}-\Delta y^-}^{y_{\max}+\Delta y^+} \overline{V}(x+y)\overline{f}(y)dy, \tag{6.7}$$

using FFT methods. The unknown values of $\overline{V}(u)$ for $u \in [y_{\max}, y_{\max} + \Delta y^+]$ are estimated using simple linear extrapolation (assuming equation (6.5) holds). The values in the left extension can be determined from interpolation on the original S grid.

This extended domain is then used as input to the forward DFT, the correlation computation (in the spectral domain), and the inverse DFT. The values in the domain extensions are affected by wrap around and are discarded. This causes no difficulty in the right extension. In the left extension, we actually need an estimate of the value $\overline{V}(S_0 = 0)$. This is obtained by extrapolation of the values at S_1, S_2.[4]

[4]Of course, the value of the integral (3.1) is trivial in many cases when $S = 0$, but this approach is always valid.

We can break up the extended integral (6.7) into the original integral and the extensions

$$\begin{aligned} I_{\text{ext}}(x) &= \int_{y_{\min}-\Delta y^-}^{y_{\max}+\Delta y^+} \overline{V}(x+y)\overline{f}(y)dy \\ &= \int_{y_{\max}}^{y_{\max}+\Delta y^+} \overline{V}(x+y)\overline{f}(y)dy + \int_{y_{\min}}^{y_{\max}} \overline{V}(x+y)\overline{f}(y)dy + \int_{y_{\min}-\Delta y^-}^{y_{\min}} \overline{V}(x+y)\overline{f}(y)dy. \end{aligned} \tag{6.8}$$

Now, we assume that $y_{\min}, y_{\max}$ are selected so that the value of the integral at any point $y_{\min} \ll x \ll y_{\max}$ is relatively unaffected by the choice of $y_{\min}, y_{\max}$. However, at $x = y_{\max}$, the integral in the domain extension to the right clearly affects the value, due to the term

$$I(y_{\max}) = \int_{y_{\max}}^{y_{\max}+\Delta y^+} \overline{V}(y_{\max}+y)\overline{f}(y)dy.$$

Since this point is remote from the region of interest, it is not necessary to obtain a highly accurate value for this integral. However, we do not wish the value to be completely unrealistic due to wrap around. We therefore assume that $\overline{V}(y)$ is given by equation (6.5). We then determine Δy^+ by requiring that

$$|e^{\Delta y^+}\overline{f}(\Delta y^+)| < tol_R. \tag{6.9}$$

As for the domain near $y_{\min}$, we use a slightly different approach. If the point $S = 0$ ($\log S \to -\infty$) is included in the original grid, we ignore this point and set $y_{\min} = \log(S_1)$, assuming $S_1 > 0$. Now, near $x = y_{\min}$ the integral in the extended domain (to the left) is given by

$$I(y_{\min}) = \int_{y_{\min}-\Delta y^-}^{y_{\min}} \overline{V}(y_{\min}+y)\overline{f}(y))dy. \tag{6.10}$$

Again, to minimize wrap around, we determine Δy^- so that these effects will be minimized for the integral evaluated at $x = y_{\min}$. Assuming $\overline{V}$ behaves near $y = -\infty$ as in equation (6.6), then we determine Δy^- by

$$|\overline{f}(-\Delta y^-)| < tol_L. \tag{6.11}$$

The domain extensions are illustrated in Figures 1-2. Typically, we chose $tol_L = tol_R = 10^{-6}$.

We next briefly describe how to extend the ideas presented thus far (which have been in the context of European options) to the case of American options. Suppose that we have to value an American style option where the holder of the contract can exercise at any time and receive a payoff of $V^*(S,\tau)$. This pricing problem can be written as the differential linear complementarity problem

$$V_\tau - \left(\frac{1}{2}\sigma^2 S^2 V_{SS} + (r-\lambda\kappa)SV_S - (r+\lambda)V + \lambda\int_0^\infty V(S\eta)g(\eta)d\eta\right) \geq 0 \tag{6.12}$$

$$V - V^* \geq 0, \tag{6.13}$$

where at least one of equations (6.12) and (6.13) must hold with equality. We can easily combine the fixed point iteration with the penalty method described in Forsyth and Vetzal (2002) to solve this complementarity problem.

Finally, we conclude this section by noting that in some cases, a fast Gauss transform (see Greengard and Strain, 1991) could also be used to evaluate the correlation integral with a computational complexity of $O(N)$. The use of this method has been explored in the general option pricing context by Broadie and Yamamoto (2002). In the particular case of jump diffusions, this approach would work for the case where

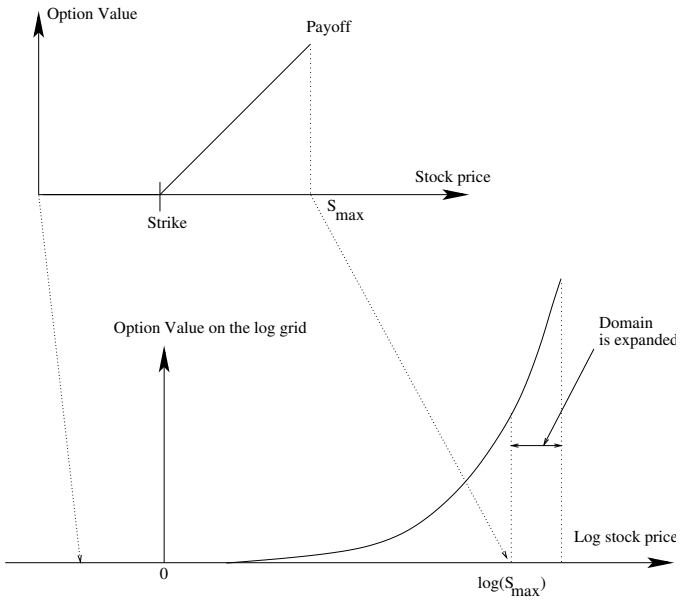


FIGURE 1: *The value of the option is interpolated onto the log-spaced grid. The right hand side boundary of the log-spaced grid* $y_{\max} = \log(S_{\max})$ *is expanded by* Δy^+*, where* Δy^+ *is given by equation* (6.9)*.*

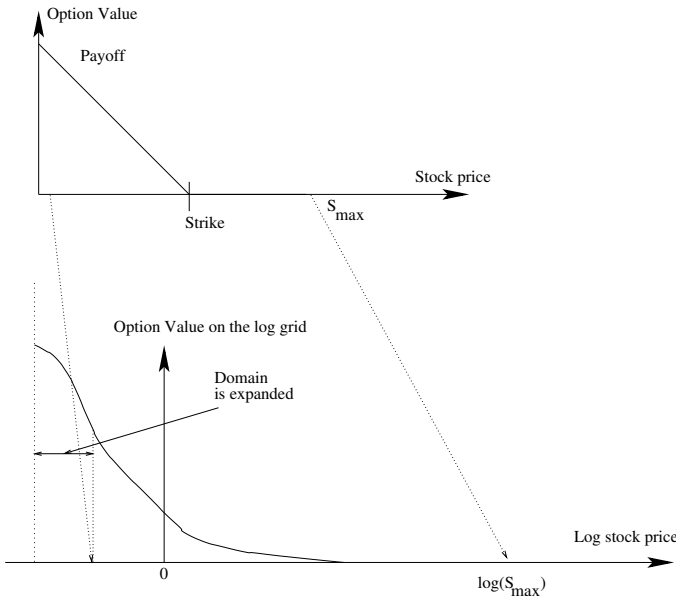


FIGURE 2: *The value of the option is interpolated onto the log-spaced grid. However, the value of the option* $V(S,\tau)$ *at* $S = 0$ *is not used. The left hand side boundary grid point is chosen to be* $\log S_1$ *where* S_1 *is the grid point nearest to* $S = 0$*. This left boundary is then expanded by* Δy^-*, which is given by equation* (6.11)*.*

the jump size is lognormally distributed. It is not clear if it could be applied for other jump distributions.[5] On grounds of generality, at this stage we therefore prefer to use the FFT. Note that all of the theoretical results in terms of convergence rates given previously would be unchanged if the fast Gauss transform were used instead of the FFT. It is possible, however, that the alternative method would be more efficient.

[5]Broadie and Yamamoto (2002) suggest some extensions to the fast Gauss transform in order to apply it to the double exponential jump distribution of Kou (2002), but they do not present any results for these extensions.

Parameter Values	
σ	0.15
r	0.05
γ	0.45
μ	-0.90
λ	0.10
T	0.25
K	100.00

TABLE 1: *Input data used to value European options under the lognormal jump diffusion process. These parameters are approximately the same as those reported in Andersen and Andreasen (2000) using European call options on the S&P 500 stock index in April of 1999.*

7 Results

This section presents numerical results for various options, including vanilla European and American options, digital options, and options with barrier features. Unless stated otherwise, we use the Crank-Nicolson discretization scheme (4.1). The discrete system of equations is solved using the fixed point iteration method (5.4) with a convergence tolerance of 10^{-6}.

We begin by considering European options under the assumptions that the continuous part of the underlying stock price process follows geometric Brownian motion and that the proportional jump size is lognormally distributed. This allows us to check the accuracy of our algorithm against the analytic solution of Merton (1976). Table 1 contains the input parameters (note that the mean of the jump distribution is μ and its standard deviation is γ). These are roughly the same as those estimated by Andersen and Andreasen (2000) using European call options on the S&P 500 stock index in April of 1999. As discussed by Andersen and Andreasen, these implied parameters are not consistent with the historical time series behavior of the underyling index, but that is only to be expected as they are based on the market's implied pricing measure, rather than the historical measure. Note that (in the market's pricing measure), if a jump occurs it is likely to involve a large negative drop, the mean value being a decline of about 55%.

We are particularly interested in the convergence properties of the algorithm as the grid is refined. For each test, as we double the number of grid points we cut the timestep size ($\Delta\tau = .01$ on the coarsest grid) in half. The convergence ratio presented in the tables below is defined in the following way. Let

$$\Delta\tau = \max_n(\tau^{n+1} - \tau^n),$$
$$\Delta S = \max_i(S_{i+1} - S_i).$$

Note that we are allowing here for the possibility of using variable timestep sizes (to be explained below), although most of our tests will simply use a constant timestep size. If we then carry out a convergence study letting $h \to 0$ where

$$\Delta S = \text{Const. } h$$
$$\Delta\tau = \text{Const. } h,$$

then we can assume that the error in the solution (at a given node) is

$$V_{\text{approx}}(h) = V_{\text{exact}} + \text{Const. } h^{\xi}.$$

Size of S grid	No. of Timesteps	Interpolation Scheme					
		Linear		Quadratic		Cubic	
		Value	R	Value	R	Value	R
128	25	3.146361	n.a.	3.145896	n.a.	3.146361	n.a.
255	50	3.148354	n.a.	3.148249	n.a.	3.148354	n.a.
509	100	3.148856	3.973	3.148831	4.039	3.148832	4.175
1017	200	3.148983	3.949	3.148977	3.990	3.148977	3.287
2033	400	3.149015	4.001	3.149014	4.007	3.149014	3.997
4065	800	3.149023	3.997	3.149023	4.002	3.149023	3.997

TABLE 2: *Value of a European put option at $S = 100$ using Crank-Nicolson timestepping for linear, quadratic and cubic interpolation. The interpolation schemes are used to transfer data between the non-uniform S grid and the uniform log-spaced FFT grid. The input parameters are provided in Table 1. The convergence ratio R is defined in equation* (7.1)*. The exact solution is* 3.149026*. The number of points used for the FFT grid is 2^{α}, where α is the smallest integer such that the number of nodes in the non-uniform S grid $p \leq 2^{\alpha}$.*

The convergence ratio is then defined as

$$R = \frac{V_{\text{approx}}(h/2) - V_{\text{approx}}(h)}{V_{\text{approx}}(h/4) - V_{\text{approx}}(h/2)}. \tag{7.1}$$

In the case of quadratic convergence ($\xi = 2$), then $R = 4$, while for linear convergence ($\xi = 1$), $R = 2$.

Recall that interpolation is required to transform data from the clustered PDE grid to the equally spaced $\log S$ grid, and vice versa. In Table 2, we compare linear interpolation (see equations (3.9)-(3.10)) with quadratic and cubic Lagrange interpolation for a vanilla European put option with different numbers of points on the FFT grid.

In Table 2 we observe quadratic convergence to the exact solution for all three interpolation schemes. Note that our earlier theoretical analysis for stability and convergence of the fixed point iteration was based on linear interpolation. This was required because linear interpolation is the only Lagrange interpolation method which has non-negative weights. Although it is not the case for these particular parameter values, our numerical experiments indicate that quadratic interpolation is often more efficient than linear interpolation (although the rate of convergence rate is theoretically the same for both methods). Consequently, in all subsequent examples we will use quadratic interpolation. In Table 3 we show the convergence rate for a call option using the data in Table 1. For each value of S in the table, we observe quite smooth second order convergence.[6]

We now consider the issues raised by the presence of a discontinuity in the payoff. Oscillations are more likely to be a problem in this context if we use Crank-Nicolson timestepping, and, unless care is taken, rates of convergence can be reduced. A detailed discussion of this can be found in Pooley et al. (2002) for the case without jumps. Following Rannacher (1984), it is possible to restore quadratic convergence if any discontinuities in the payoff (arising either due to the payoff function itself in the case of a digital option, or from the application of a discretely observed barrier) are l_2 projected onto the space of linear Lagrange basis functions, and a fully implicit method is used for a small number of timesteps after any discontinuities arise. We will refer to this technique as Rannacher timestepping. While it does ensure quadratic convergence, it

[6] As an aside, all of our tests were performed on an Intel P4 PC running at 2.0GHz with 2 gigabytes of RAM, using the g++ compiler under Red Hat Linux. Solution times for the examples in Table 3 ranged from around 1/100th of a second for the coarse grid with 128 nodes and 25 timesteps to about 1/3 of a second for the case with 1017 spatial nodes and 100 timesteps, and up to about 7 seconds for the finest grid with 4065 nodes and 800 timesteps.

Size of S grid	No. of Timesteps	$S = 90$		$S = 100$		$S = 110$	
		Value	R	Value	R	Value	R
128	25	0.526562	n.a.	4.388091	n.a.	12.641501	n.a.
255	50	0.527379	n.a.	4.390462	n.a.	12.642942	n.a.
509	100	0.527574	4.186	4.391050	4.039	12.643290	4.125
1017	200	0.527622	4.042	4.391197	3.991	12.643377	4.008
2033	400	0.527634	4.046	4.391233	4.005	12.643399	4.059
4065	800	0.527637	4.023	4.391243	4.002	12.643404	4.049

TABLE 3: *Value of a European call option using Crank-Nicolson timestepping. The input parameters are provided in Table 1. The convergence ratio R is defined in equation* (7.1)*. The exact solution is* 0.527638 *at* $S = 90$*,* 4.391246 *at* $S = 100$*, and* 12.643406 *at* $S = 110$*. The number of points used for the FFT grid is* 2^{α}*, where* α *is the smallest integer such that the number of nodes in the non-uniform S grid* $p \leq 2^{\alpha}$*. Quadratic interpolation is used.*

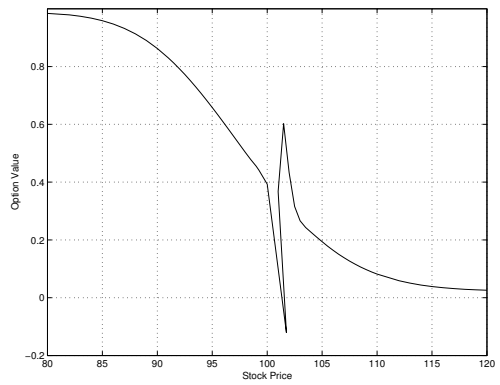
Size of S grid	No. of Timesteps	$S = 90$		$S = 100$		$S = 110$	
		Value	R	Value	R	Value	R
128	25	0.855482	n.a.	0.387139	n.a.	0.077539	n.a.
255	50	0.855034	n.a.	0.387151	n.a.	0.077830	n.a.
509	100	0.854935	4.577	0.387152	7.157	0.077899	4.160
1017	200	0.854910	3.924	0.387153	2.219	0.077917	3.896
2033	400	0.854902	2.824	0.387153	3.590	0.077922	3.970
4065	800	0.854899	3.327	0.387153	4.124	0.077923	4.005

TABLE 4: *Value of a European digital put option using Rannacher timestepping and* l_2 *projection. The input parameters are provided in Table 1. The convergence ratio R is defined in equation* (7.1)*. The exact solution is* 0.854898 *at* $S = 90$*,* 0.387153 *at* $S = 100$*, and* 0.077923 *at* $S = 110$*. The number of points used for the FFT grid is* 2^{α}*, where* α *is the smallest integer such that the number of nodes in the non-uniform S grid* $p \leq 2^{\alpha}$*. Quadratic interpolation is used.*

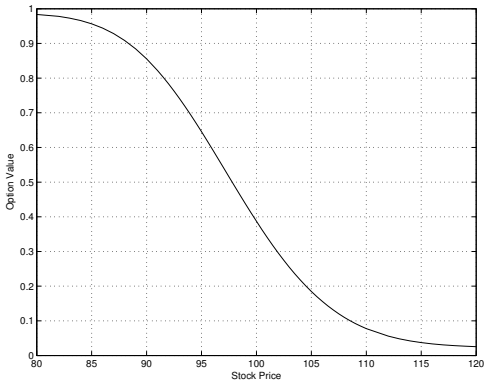
does not guarantee the absence of oscillations. Typically, however, the use of the fully implicit timesteps smooths out the function enough that oscillations are not a problem.

Of course, the forgoing discussion is for the case without jumps. However, we can intuitively expect that the result should also hold for the jump diffusion case since integration is a smoothing operation. Accordingly, we will investigate the application of Rannacher timestepping in the jump diffusion context for a digital put option which pays \$1 at maturity if the underlying stock price is below the strike price. Table 4 gives a convergence study for the digital put with jumps, using Rannacher timestepping (with two fully implicit steps) and l_2 projection. As shown in this table, quadratic convergence is generally achieved, though perhaps a bit more erratically than for the vanilla payoff as shown in Table 3. Figure 3 provides plots of the solution value for a digital put along with its first and second derivatives with respect to the underlying asset value (i.e. delta and gamma) for both Crank-Nicolson and Rannacher timestepping. The plots are all smooth when we use Rannacher timestepping, but there is an oscillation near the discontinuity in the payoff when the solution value is calculated using Crank-Nicolson. Naturally, this leads to even worse behavior in the derivatives, as shown in panels (c) and (e) of Figure 3.

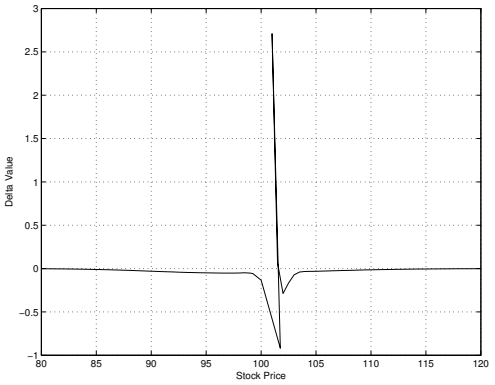
Our next two numerical tests incorporate the use of an automatic timestep size selector as described in Johnson (1987). This offers a couple of advantages. First, it is not generally possible to achieve second order convergence for American options using constant timesteps (Forsyth and Vetzal, 2002). Second, variable timestepping can be very efficient for long-dated options. The procedure is as follows. An initial timestep



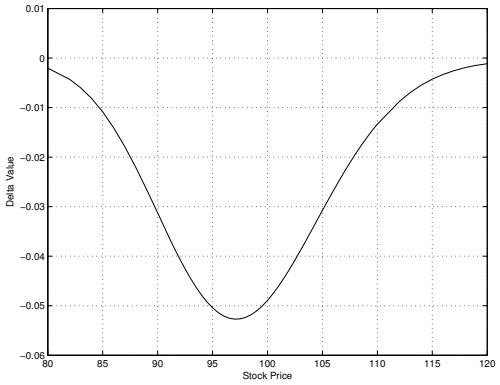
(a) *Digital put value: Crank-Nicolson.*



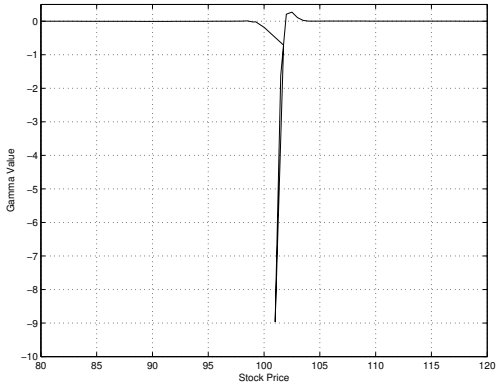
(b) *Digital put value: Rannacher.*



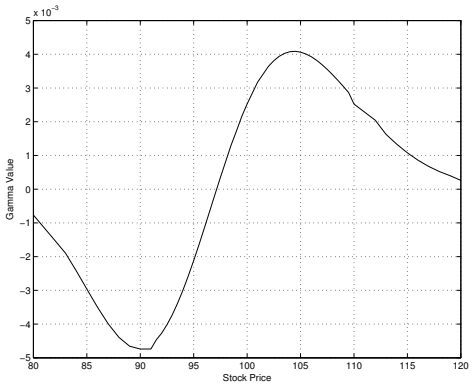
(c) *Digital put delta: Crank-Nicolson.*



(d) *Digital put delta: Rannacher.*



(e) *Digital put gamma: Crank-Nicolson.*



(f) *Digital put gamma: Rannacher.*

FIGURE 3: *Digital put option value, delta and gamma for Rannacher and Crank-Nicolson timestepping. The input parameters are provided in Table 1.*

is given and the next timestep is computed according to

$$\frac{\tau^{n+2}-\tau^{n+1}}{\tau^{n+1}-\tau^{n}} = \frac{d}{\max_i \frac{|V_i^{n+1}-V_i^n|}{\max(1,|V_i^n|)}}, \tag{7.2}$$

where d specifies the maximum relative change allowed. Initially we set $d = .1$, and we divide this value by two at each grid refinement. An initial timestep of .01 on the coarsest grid is used, and this initial timestep is reduced by four at each refinement.

The first test to incorporate variable timesteps involves an alternative distribution for the jump size. Kou (2002) suggests the double exponential distribution for the log jump size, observing that it has desirable analytical properties. In addition to deriving a closed form solution for the value of European vanilla options (as in the Merton (1976) lognormally distributed jump model), Kou notes that analytic solutions can be derived for some exotic options for the double exponential case because the first passage time to a flat boundary can be calculated. This is not possible in the lognormal case.

In the model of Kou (2002),

$$\overline{f}(x) = p\eta_1 \exp(-\eta_1 x)H(x) + q\eta_2 \exp(\eta_2 x)H(-x), \tag{7.3}$$

where $\eta_1 > 1$, $\eta_2 > 0$, $p > 0$, $q = 1-p > 0$, and $H(\cdot)$ is the Heaviside function. As noted by Kou, the condition $\eta_1 > 1$ is used to ensure that the proportional jump and stock price have finite expectation. In this model, $\kappa = E[J-1] = \frac{p\eta_1}{\eta_1-1} + \frac{q\eta_2}{\eta_2+1} - 1$.

To provide a basis for comparison with the lognormal distribution, we attempted to find parameters for the double exponential distribution which match those used for the lognormal given in Table 1. This did not work well for those parameters, as the mean is too far below zero, resulting in only the left tail of the double exponential being used. To remedy this, we shifted the lognormal mean from its value of -.90 in Table 1 to -.10. We then performed a numerical search to find parameters to match the first three central moments of the two distributions as closely as possible. We obtained values of $p = 0.3445$, $\eta_1 = 3.0465$, and $\eta_2 = 3.0775$. Figure 4 shows the double exponential probability density function and the normal probability density function for our parameter values. Note that the double exponential distribution has a discontinuity at zero. This can be expected to cause some problems for our numerical integration using an FFT method.

Table 5 presents numerical convergence tests for pricing a European call option. In an attempt to deal with the discontinuity at zero, the number of points used on the uniform-spaced x grid has been oversampled to a greater extent than in the lognormal case. In particular, the number of points on the FFT grid is $8 \times 2^{\alpha}$, where α is the smallest integer such that 2^{α} is at least equal to the number of nodes in the S grid. Rannacher timestepping is used. In contrast to our earlier examples, we do not obtain second order convergence here. Instead the results indicate convergence at a linear (or perhaps slightly higher) rate to the exact solution.[7] Despite the discontinuity, we observe smooth solution plots for the solution value, delta, and gamma in Figure 5. One other point worthy of mention: as noted in the caption to Table 5, the exact value is 0.672677 at $S = 90$, 3.973479 at $S = 100$, and 11.794583 at $S = 110$. By way of comparison, the exact value under the lognormal jump distribution (given the parameters that match the first three central moments of the double exponential distribution) is 0.681403 at $S = 90$, 4.005789 at $S = 100$, and 11.839672 at $S = 110$. This suggests that pricing differences between these two jump distributions may not be very significant, if the parameters are calibrated similarly. Of course, there may be more significant differences for other parameter values or for different payoff functions.

[7] Some other numerical experiments indicate that we can achieve quadratic convergence in the double exponential case if we restrict the parameters so that the distribution is continuous at zero (i.e. set $p = 0.50, \eta_1 = \eta_2$). This still requires a heavily oversampled FFT grid relative to the lognormal case in order to adequately capture the sharp peak of the distribution.

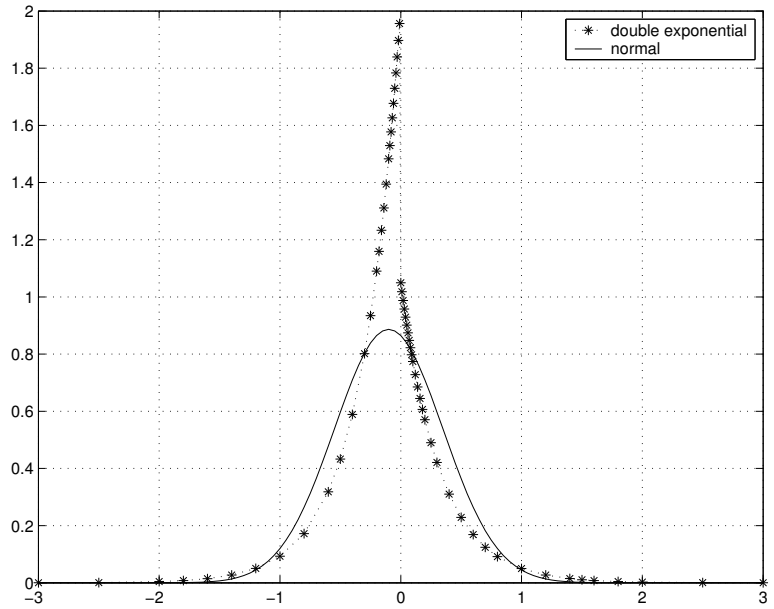


FIGURE 4: *Overall comparison of the normal ($\mu = -.10, \gamma = .45$) and double exponential probability density functions ($p = 0.3445$, $\eta_1 = 3.0465$, $\eta_2 = 3.0775$).*

Our second test involving variable timesteps is the valuation of an American put option. As mentioned in Section 6, this is easily handled in our framework by combining the fixed point iteration with the penalty method described in Forsyth and Vetzal (2002). In order to achieve higher than linear convergence, it is also necessary in this context to use variable timestepping. This is formally shown by Forsyth and Vetzal (2002) (for the no-jump case). The intuition is quite simple: when the exercise boundary is changing rapidly (i.e. near the expiry date), smaller timesteps are required. Table 6 presents the results for the case of lognormally distributed jumps. Once again, we observe second order convergence, and there is no evidence of oscillations in either the solution or its first two derivatives with respect to the underlying stock price (see Figure 6). Also note that in this case the analytic approximation of Bates (1991) is quite accurate for the out of the money case where $S = 110$, about eight cents too low when $S = 100$, and around 70 cents too low for the in the money case with $S = 90$. This suggests that (at least for our parameter values), Bates's approximation is not very accurate (in terms of absolute pricing error), unless the option is deep out of the money.

The last set of results to be presented are for the case of a European call option with a Parisian knock-out feature. The particular case we consider here is an up-and-out call with daily discrete observation dates. This contract ceases to have value if S is above a specified barrier level for a specified number of consecutive monitoring dates. As with many path-dependent contracts, this can be valued by solving a set of one-dimensional problems which exchange information at monitoring dates. See Vetzal and Forsyth (1999) for a complete discussion in the case without jumps. However, it is easy to incorporate jumps by simply adding a jump integral term to each of the one-dimensional problems. Note also that we can use the same procedure to value a variety of similar contracts, such as knock-in contracts, double barrier contracts, delayed barrier options (for which the knock-out condition applies if the total number of observations over the barrier during the life of the contract reaches the specified level), step options (which lose a fraction of their value for each day the underlying asset lies above the barrier level, see Linetsky (1999) for more details), etc.[8]

[8]Other path-dependent contracts such as Asian options can also be handled using this approach of solving a set of one-dimensional problems. An additional complication in the case of Asian options is that a suitable form of interpolation is required

Size of S grid	No. of Timesteps	$S = 90$		$S = 100$		$S = 110$	
		Value	R	Value	R	Value	R
128	34	0.671314	n.a.	3.969969	n.a.	11.78927	n.a.
255	65	0.672213	n.a.	3.972476	n.a	11.79248	n.a.
509	132	0.672535	2.791	3.973107	3.972	11.79367	2.688
1017	266	0.672630	3.358	3.973322	2.936	11.79416	2.431
2033	533	0.672660	3.225	3.973407	2.511	11.79438	2.244
4065	1067	0.672670	2.917	3.973445	2.281	11.79448	2.130

TABLE 5: *Value of a European vanilla call option using Rannacher timestepping with variable timestep sizes for the double exponential probability density function* (7.3)*. The timesteps are selected using equation* (7.2)*, with* $d = 0.1$ *on the coarsest grid, and divided by two for each grid refinement. The input parameters are* $\sigma = 0.15$*,* $r = 0.05$*,* $\lambda = 0.1$*,* $T = 0.25$*,* $K = 100$*,* $\eta_1 = 3.0465$*,* $\eta_2 = 3.0775$*, and* $p = .3445$*. The convergence ratio R is defined in equation* (7.1)*. The exact solution is* 0.672677 *at* $S = 90$*,* 3.973479 *at* $S = 100$*, and* 11.794583 *at* $S = 110$*. The number of points used for the FFT grid is* $8 \times 2^{\alpha}$*, where* α *is the smallest integer such that the number of nodes in the non-uniform S grid* $p \leq 2^{\alpha}$*. Quadratic interpolation is used.*

Size of S grid	No. of Timesteps	$S = 90$		$S = 100$		$S = 110$	
		Value	R	Value	R	Value	R
128	32	10.000000	n.a.	3.236354	n.a.	1.417613	n.a.
255	58	10.002938	n.a.	3.240286	n.a	1.419269	n.a.
509	117	10.003519	5.058	3.241045	5.182	1.419676	4.077
1017	235	10.003791	2.137	3.241207	4.699	1.419774	4.139
2033	470	10.003815	11.653	3.241243	4.463	1.419798	4.143
4065	940	10.003822	3.213	3.241251	4.331	1.419803	4.127

TABLE 6: *Value of an American put option using Rannacher timestepping with variable timestep sizes. The timesteps are selected using equation* (7.2)*, with* $d = 0.1$ *on the coarsest grid, and divided by two for each grid refinement. The input parameters are provided in Table 1. The convergence ratio R is defined in equation* (7.1)*. The approximate analytic values from Bates (1991) are* 9.304946 *at* $S = 90$*,* 3.163112 *at* $S = 100$*, and* 1.411669 *at* $S = 110$*. The number of points used for the FFT grid is* 2^{α}*, where* α *is the smallest integer such that the number of nodes in the non-uniform S grid* $p \leq 2^{\alpha}$*. Quadratic interpolation is used.*

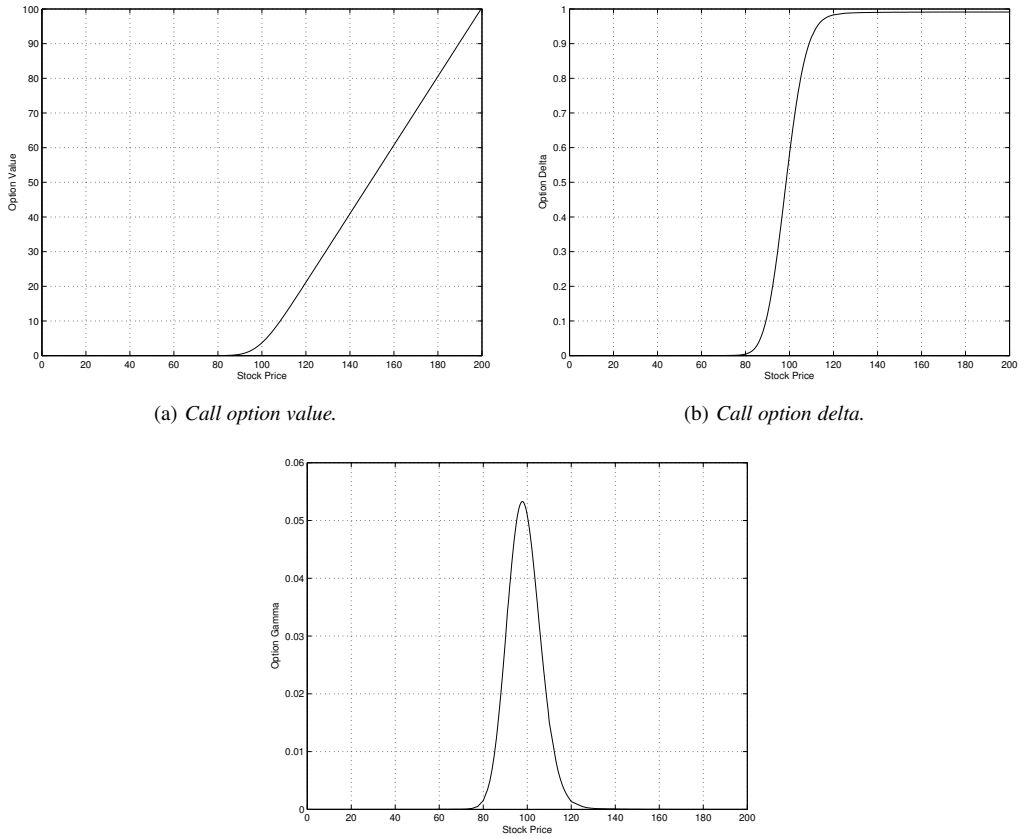
For our test, the barrier is set at $S = 120$ and the required number of consecutive daily observations for knock-out is 10. We consider the lognormal jump distribution case with the same input parameters as in Table 1. Note that we specify the barrier observation interval as 1/250, based on 250 trading days per year. In Table 7, we present our convergence results. We use constant timestepping ($\Delta\tau = .002$ on the coarse grid) and the solution is l_2 projected after each barrier observation date. Rannacher timestepping is used after each observation. As expected, quadratic convergence is obtained.

In Figure 7, we compare the solutions of a Parisian call knock-out option with discrete daily observation dates with and without jumps. To ensure a consistent basis for comparison, we use the following procedure:

1. Given some parameters (in this example we use the values provided in Table 1), compute the analytical solution V_{jump} at the strike $K = 100$ of a vanilla European call option.

2. Use a constant volatility Black-Scholes model with no jumps to determine the implied volatility

because the number of possible values for the average value of the underlying stock grows exponentially, and it is not possible to solve a one-dimensional problem for every possible value of the average. See Zvan et al. (1999) for a thorough analysis in the no jump case.



(a) *Call option value.*

(b) *Call option delta.*

(c) *Call option gamma.*

FIGURE 5: *Call option value, delta and gamma for Rannacher timestepping using the double exponential probability density function* (7.3)*. The input parameters are provided in the caption to Table 5.*

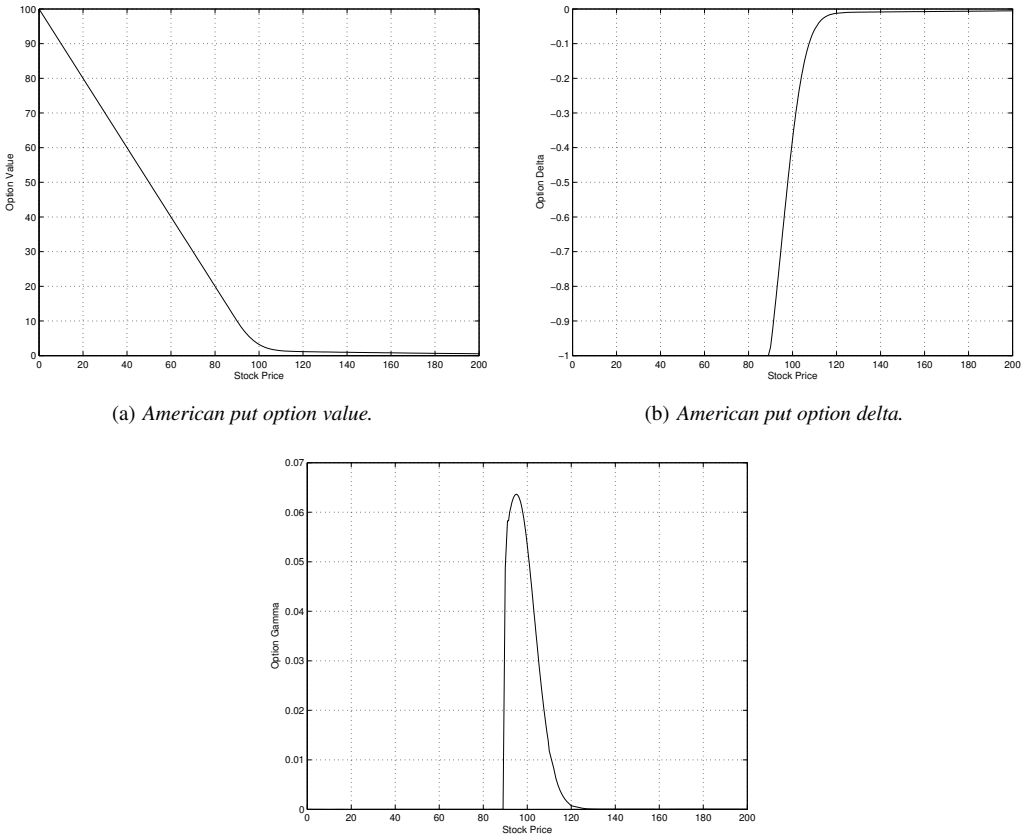
σ_{implied} which matches the option price to the jump diffusion value V_{jump} at the strike K.

3. Value the Parisian knock-out call option with discrete daily observation dates with jumps using the same parameters as in Step 1.

4. Value the Parisian knock-out call option with discrete daily observation dates using a constant volatility model (no jumps) but with the implied volatility $\sigma_{implied}$ estimated in Step 2.

We observe in Figure 7 that the difference in pricing can be significant for these parameter values, depending on the underlying asset price. The largest differences are near $S = 110$, where the model with jumps produces values of about 8.78 (as shown in Table 7), but the values for the no-jump model are around 7.25. For S ranging between about 98 and around 119, the jump model produces higher option values, but outside this range (in either direction) the model without jumps produces higher values.

It is worth concluding this section by making some comparisons with other methods which have been proposed in the literature. When pricing options under the jump diffusion process, the main computational cost is the evaluation of the integral term of (2.4). The approach presented in Andersen and Andreasen (2000) is based on a FFT-ADI finite difference method. This method evaluates the convolution integral twice at each timestep, thus requiring a total of four FFT computations (two forward FFTs, and two reverse FFTs). Note that the method in Andersen and Andreasen (2000) is second order accurate. If N is the number



(a) *American put option value.*

(b) *American put option delta.*

(c) *American put option gamma.*

FIGURE 6: *American put option value, delta and gamma for Rannacher timestepping with* 128 *points on the non-uniform S grid and an initial timestep of* 0.01.

of timesteps, and p the number of nodes in the S grid, then both Andersen and Andreasen's method and the method in this work have complexity $O(Np \log p)$.

In Table 8, we see that the number of iterations required for convergence (at each timestep) depends on the convergence tolerance. For a typical convergence tolerance of 10^{-6}, at most three iterations per step are required (on average). In this case, about six FFT computations are required per timestep. Consequently, for vanilla European options (with jumps), the method of Andersen and Andreasen (2000) may be more efficient than the pure Crank-Nicolson timestepping method developed here.

However, in the case of American options, it is not clear how the approach in Andersen and Andreasen (2000) could be modified to handle the early exercise constraint implicitly, unless some form of iteration is used. In contrast, our technique can handle implicit treatment of the American constraint in a straightforward fashion.

The technique developed in Matache et al. (2002) uses a wavelet method for the evaluation of the jump integral term. This has complexity $O(Np\ (\log p)^2)$, in contrast to a complexity of $O(Np\ \log p)$ (for one dimensional problems) for the method developed here. In addition, it is not obvious how to generalize the technique in Matache et al. (2002) to nonlinear cases such as uncertain volatility or transaction costs models, which can easily be handled using our method.

Finally, we note that the method in Zhang (1997) uses an explicit evaluation of the correlation integral term, and hence is only first order accurate. The approach of Meyer (1998) is restricted to cases where the

Size of S grid	No. of Timesteps	$S = 90$		$S = 100$		$S = 110$	
		Value	R	Value	R	Value	R
101	125	0.524766	n.a.	4.193418	n.a	8.762555	n.a.
201	250	0.523168	n.a	4.212131	n.a	8.779253	n.a.
401	500	0.522761	3.930	4.216747	4.053	8.782267	5.540
801	1000	0.522660	4.002	4.217902	3.997	8.783008	4.068
1601	2000	0.522634	4.015	4.218192	3.990	8.783199	3.875

TABLE 7: *Value of an up-and-out Parisian call option using Rannacher timestepping with constant timesteps ($\Delta\tau = .002$ on the coarsest grid) and l_2 projection. The input parameters are given in Table 1. The barrier is set at $S = 120$ and 10 consecutive daily observations are required to knock-out. The convergence ratio R is defined in equation* (7.1)*. The number of points used for the FFT grid is 2^α, where α is the smallest integer such that the number of nodes in the non-uniform S grid $p \leq 2^\alpha$. Quadratic interpolation is used.*

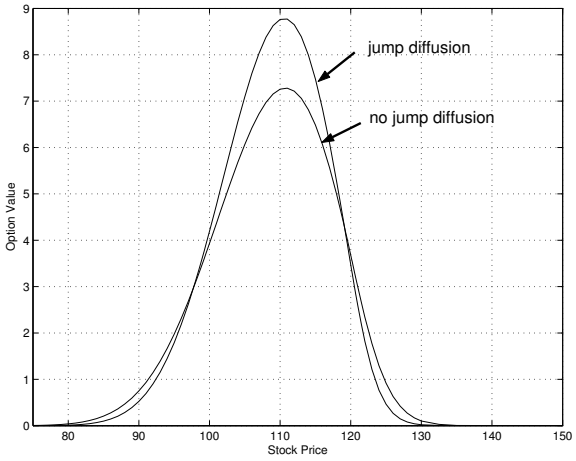


FIGURE 7: *Parisian knock-out call option with discrete daily observation dates with and without jumps. The barrier is set at $S = 120$ and the number of consecutive daily observations to knock-out is* 10*.*

underlying asset can only jump to a (small) finite number of states.

8 Conclusion

In this paper, we have shown that an explicit evaluation of the correlation integral in the jump diffusion PIDE, coupled with an implicit discretization of the usual PDE terms, is unconditionally stable. However, since this method is only first order correct, an implicit method is preferred. We show that Crank-Nicolson timestepping is algebraically stable, and in the special case of an equally spaced $\log S$ grid with constant parameters, we can prove that Crank-Nicolson timestepping is strictly stable.

If implicit timestepping is used, then the direct evaluation of the correlation integral appearing in the PIDE would require a dense matrix solve. To avoid this computational complexity, a fixed point iteration method is developed. For typical parameter values, this fixed point iteration converges very quickly (the error is reduced by two orders of magnitude at each iteration).

Each fixed point iteration requires evaluation of the correlation integral. We use Lagrange interpolation to transfer the data on the clustered PDE grid to an equally spaced $\log S$ grid. An FFT method is then used to evaluate the correlation integral, and Lagrange interpolation is used to transfer data back to the PDE grid. We demonstrate how to extend the $\log S$ grid to avoid FFT wrap around effects. This is done by taking into

Number of points N	Timesteps	Iterations ($tol = 10^{-6}$)	Iterations ($tol = 10^{-8}$)
128	25	77	100
255	50	150	200
509	100	300	390
1017	200	600	600
2033	400	1091	1200
4065	800	1600	2400

TABLE 8: *Number of iterations for a European call option under jump diffusion using Crank-Nicolson timestepping. The input parameters are provided in Table 1. The convergence tolerance* tol *is defined in equation* (5.4)*.*

account the properties of the jump size probability density.

The methods developed in this paper can be applied to arbitrary jump size probability densities. Furthermore, we have demonstrated that this method can be used to obtain implicit solutions to American options with jump diffusion. Since the method used to handle the jump diffusion term implicitly is a simple fixed point iteration, then it is a very simple matter to modify an existing exotic option pricing library to handle the jump diffusion case. All that is required is that a function be added to the library which, given the current vector of discrete option prices, returns the vector value of the correlation integral. This vector is then added to the right hand side of the fixed point iteration.

There are several obvious avenues for future research. One would be a detailed analysis of pricing and hedging various types of exotic options under a jump diffusion process. Similarly, it would be interesting to explore the effects of uncertain parameters or transactions costs, as described in Wilmott (1998) for the diffusion case. Another possibility would be to extend the analysis to more complex models for the evolution of the underlying state variable. Among the candidates here are more general Lévy processes than the jump diffusion case, or multifactor models such as those recently explored by Eraker et al. (2003), which feature stochastic volatility with Poisson jumps in both the state variable itself and its volatility.

Appendix

A Von Neumann Stability Analysis

In this Appendix, we will carry out a Von Neumann stability analysis for Crank-Nicolson timestepping in the special case of constant parameters and an equally spaced grid in $\log S$ coordinates.

From equations (2.4) and (3.2),

$$V_\tau = \frac{1}{2}\sigma^2 S^2 V_{SS} + (r - \lambda\kappa)SV_S - (r+\lambda)V + \lambda\int_{-\infty}^{\infty} \overline{V}(y) f\left(y - \log S\right)\, dy. \tag{A.1}$$

where $\overline{V}(x,\tau) = V(\exp(x),\tau)$ and $\overline{f}(y) = f(\exp(y))$. Using the change of variable $x = \log(S)$ and substituting into (A.1), we obtain

$$\overline{V}_\tau = \frac{1}{2}\sigma^2\overline{V}_{xx} + (r - \lambda\kappa - \frac{1}{2}\sigma^2)\overline{V}_x - (r+\lambda)\overline{V} + \lambda\int_{-\infty}^{\infty} \overline{V}(y)\overline{f}(y-x)dy. \tag{A.2}$$

From equation (A.2), it can be observed that the integral part of the PIDE is simply a correlation product. Using the correlation operator $\otimes$ from equation (3.3), equation (A.2) can be written as

$$\overline{V}_\tau = \frac{1}{2}\sigma^2\overline{V}_{xx} + (r - \lambda\kappa - \frac{1}{2}\sigma^2)\overline{V}_x - (r+\lambda)\overline{V} + \lambda\overline{V}\otimes\overline{f}. \tag{A.3}$$

A Crank-Nicolson discretization of equation (A.3) is

$$\begin{aligned}\frac{\overline{V}_i^{n+1}-\overline{V}_i^n}{\Delta\tau} &= \frac{\lambda}{2}\left[(\overline{V}\otimes\overline{f})_i^n+(\overline{V}\otimes\overline{f})_i^{n+1}\right]\\ &+\frac{1}{2}\left[\frac{1}{2}\sigma^2\left(\frac{\overline{V}_{i+1}^{n+1}-2\overline{V}_i^{n+1}+\overline{V}_{i-1}^{n+1}}{\Delta x^2}\right)+\left(r-\lambda\kappa-\frac{1}{2}\sigma^2\right)\left(\frac{\overline{V}_{i+1}^{n+1}-\overline{V}_{i-1}^{n+1}}{2\Delta x}\right)-(r+\lambda)\overline{V}_i^{n+1}\right]\\ &+\frac{1}{2}\left[\frac{1}{2}\sigma^2\left(\frac{\overline{V}_{i+1}^{n}-2\overline{V}_i^{n}+\overline{V}_{i-1}^{n}}{\Delta x^2}\right)+\left(r-\lambda\kappa-\frac{1}{2}\sigma^2\right)\left(\frac{\overline{V}_{i+1}^{n}-\overline{V}_{i-1}^{n}}{2\Delta x}\right)-(r+\lambda)\overline{V}_i^{n}\right]. \quad \text{(A.4)}\end{aligned}$$

Equation (A.4) can be written as

$$\begin{aligned}&\overline{V}_i^{n+1}\left[1+(\alpha+\beta+r+\lambda)\frac{\Delta\tau}{2}\right]-\frac{\Delta\tau}{2}\beta\overline{V}_{i+1}^{n+1}-\frac{\Delta\tau}{2}\alpha\overline{V}_{i-1}^{n+1}\\ &\quad=\overline{V}_i^n\left[1-(\alpha+\beta+r+\lambda)\frac{\Delta\tau}{2}\right]+\frac{\Delta\tau}{2}\beta\overline{V}_{i+1}^{n}+\frac{\Delta\tau}{2}\alpha\overline{V}_{i-1}^{n}+\frac{\Delta\tau}{2}\lambda\left[(\overline{V}\otimes\overline{f})_i^n+(\overline{V}\otimes\overline{f})_i^{n+1}\right], \quad \text{(A.5)}\end{aligned}$$

where

$$\alpha=\frac{\sigma^2}{2\Delta x^2}-\frac{r-\lambda\kappa-\frac{\sigma^2}{2}}{2\Delta x} \quad \text{(A.6)}$$

$$\beta=\frac{\sigma^2}{2\Delta x^2}+\frac{r-\lambda\kappa-\frac{\sigma^2}{2}}{2\Delta x}. \quad \text{(A.7)}$$

Let $\hat{V}^n=[\overline{V}_0^n,\overline{V}_1^n,\ldots,\overline{V}_p^n]'$ be the discrete solution vector to equation (A.3). Suppose the initial solution vector is perturbed, i.e. $\hat{V}^0=\overline{V}^0+E^0$, where $E^n=[E_0^n,\ldots,E_p^n]'$ is the perturbation vector. Note that $E_p^n=0$ since Dirichlet boundary conditions are imposed at this node. Then, from equation (A.5), we we obtain the following equation for the propagation of the perturbation

$$\begin{aligned}&E_i^{n+1}\left[1+(\alpha+\beta+r+\lambda)\frac{\Delta\tau}{2}\right]-\frac{\Delta\tau}{2}\beta E_{i+1}^{n+1}-\frac{\Delta\tau}{2}\alpha E_{i-1}^{n+1}\\ &\quad=E_i^n\left[1-(\alpha+\beta+r+\lambda)\frac{\Delta\tau}{2}\right]+\frac{\Delta\tau}{2}\beta E_{i+1}^{n}+\frac{\Delta\tau}{2}\alpha E_{i-1}^{n}+\frac{\Delta\tau}{2}\lambda\left[(E\otimes\overline{f})_i^n+(E\otimes\overline{f})_i^{n+1}\right]. \quad \text{(A.8)}\end{aligned}$$

In the following we determine the stability of our discretization scheme using the von Neumann approach (Richtmyer and Morton, 1967). In order to apply the Fourier transform method, we assume that the boundary conditions can be replaced by periodicity conditions. We define the inverse discrete Fourier transform (DFT) as follows (note that we have selected a particular scaling factor)

$$E_i^n=\frac{1}{X_N}\sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}}C_k^n\exp\left(\sqrt{-1}\,\frac{2\pi}{N}ik\right) \quad \text{(A.9)}$$

$$f_i=\frac{1}{X_N}\sum_{l=-\frac{N}{2}+1}^{\frac{N}{2}}F_l\exp\left(\sqrt{-1}\,\frac{2\pi}{N}il\right), \quad \text{(A.10)}$$

where C_k and F_l correspond respectively to the discrete Fourier coefficients of E and f, and $X_N=x_{N/2}-x_{-N/2+1}$ is the width of the domain along the x-axis. Note that the notation C_k^n should be interpreted as $(C_k)^n$, i.e. in this case n is a power, not a superscript.

The forward transforms are

$$C_k^n = \frac{X_N}{N} \sum_{i=-\frac{N}{2}+1}^{\frac{N}{2}} E_i^n \exp\left(-\sqrt{-1}\,\frac{2\pi}{N} ik\right) \tag{A.11}$$

$$F_l = \frac{X_N}{N} \sum_{i=-\frac{N}{2}+1}^{\frac{N}{2}} f_i \exp\left(-\sqrt{-1}\,\frac{2\pi}{N} il\right). \tag{A.12}$$

The discrete correlation is given by

$$(E \otimes \overline{f})_i^n = \frac{X_N}{N} \sum_{j=-\frac{N}{2}+1}^{\frac{N}{2}} E_j^n f_{j-i}, \tag{A.13}$$

which is second order accurate. Substituting (A.9) and (A.10) into (A.13), we obtain

$$\begin{aligned}(E \otimes \overline{f})_i^n &= \frac{X_N}{N} \sum_{j=-\frac{N}{2}+1}^{\frac{N}{2}} \frac{1}{X_N} \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} C_k^n \exp\left(\sqrt{-1}\,\frac{2\pi}{N} jk\right) \frac{1}{X_N} \sum_{l=-\frac{N}{2}+1}^{\frac{N}{2}} F_l \exp\left(\sqrt{-1}\,\frac{2\pi}{N}(j-i)l\right) \\ &= \frac{1}{X_N}\frac{1}{N} \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} \sum_{l=-\frac{N}{2}+1}^{\frac{N}{2}} C_k^n F_l \exp\left(-\sqrt{-1}\,\frac{2\pi}{N} il\right) \sum_{j=-\frac{N}{2}+1}^{\frac{N}{2}} \exp\left(\sqrt{-1}\,\frac{2\pi}{N} jk\right) \exp\left(\sqrt{-1}\,\frac{2\pi}{N} jl\right).\end{aligned}$$

Using the orthogonality condition

$$\sum_{j=-\frac{N}{2}+1}^{\frac{N}{2}} \exp\left(\sqrt{-1}\,\frac{2\pi}{N} jk\right) \exp\left(\sqrt{-1}\,\frac{2\pi}{N} jl\right) = \begin{cases} N & \text{if } l = -k \\ 0 & \text{otherwise} \end{cases}, \tag{A.14}$$

we find that

$$(E \otimes \overline{f})_i^n = \frac{1}{X_N} \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} C_k^n F_{-k} \exp\left(\sqrt{-1}\,\frac{2\pi}{N} ik\right). \tag{A.15}$$

Substituting (A.9) and (A.15) into (A.8) gives

$$\begin{aligned}&\frac{1}{X_N} \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} C_k^{n+1} \exp\left(\sqrt{-1}\,\frac{2\pi}{N} ik\right) \left[1 + (\alpha+\beta+r+\lambda)\frac{\Delta\tau}{2}\right] \\ &\quad - \frac{\Delta\tau}{2}\beta \frac{1}{X_N} \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} C_k^{n+1} \exp\left(\sqrt{-1}\,\frac{2\pi}{N}(i+1)k\right) - \frac{\Delta\tau}{2}\alpha \frac{1}{X_N} \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} C_k^{n+1} \exp\left(\sqrt{-1}\,\frac{2\pi}{N}(i-1)k\right) \\ &\qquad = \\ &\qquad \frac{1}{X_N} \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} C_k^n \exp\left(\sqrt{-1}\,\frac{2\pi}{N} ik\right) \left[1 - (\alpha+\beta+r+\lambda)\frac{\Delta\tau}{2}\right] \\ &\quad + \frac{\Delta\tau}{2}\beta \frac{1}{X_N} \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} C_k^n \exp\left(\sqrt{-1}\,\frac{2\pi}{N}(i+1)k\right) + \frac{\Delta\tau}{2}\alpha \frac{1}{X_N} \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} C_k^n \exp\left(\sqrt{-1}\,\frac{2\pi}{N}(i-1)k\right) \\ &\qquad + \frac{\Delta\tau}{2}\lambda \left[\frac{1}{X_N} \sum_{k=-\frac{N}{2}}^{\frac{N}{2}} C_k^n F_{-k} \exp\left(\sqrt{-1}\,\frac{2\pi}{N} ik\right) + \frac{1}{X_N} \sum_{k=-\frac{N}{2}}^{\frac{N}{2}} C_k^{n+1} F_{-k} \exp\left(\sqrt{-1}\,\frac{2\pi}{N} ik\right)\right]\end{aligned} \tag{A.16}$$

Because of linearity, each Fourier component can be treated separately. Equation (A.16) becomes

$$\begin{aligned}
C_k^{n+1}\exp\left(\sqrt{-1}\,\frac{2\pi}{N}ik\right)\left[1+(\alpha+\beta+r+\lambda)\,\frac{\Delta\tau}{2}\right] & \\
-\frac{\Delta\tau}{2}\beta C_k^{n+1}\exp\left(\sqrt{-1}\frac{2\pi}{N}(i+1)k\right)-\frac{\Delta\tau}{2}\alpha C_k^{n+1}\exp\left(\sqrt{-1}\frac{2\pi}{N}(i-1)k\right) & \\
= & \\
C_k^{n}\exp\left(\sqrt{-1}\,\frac{2\pi}{N}ik\right)\left[1-(\alpha+\beta+r+\lambda)\,\frac{\Delta\tau}{2}\right] & \\
+\frac{\Delta\tau}{2}\beta C_k^{n}\exp\left(\sqrt{-1}\frac{2\pi}{N}(i+1)k\right)+\frac{\Delta\tau}{2}\alpha C_k^{n}\exp\left(\sqrt{-1}\frac{2\pi}{N}(i-1)k\right) & \\
+\frac{\Delta\tau}{2}\lambda\left[C_k^{n}F_{-k}\exp\left(\sqrt{-1}\,\frac{2\pi}{N}ik\right)+C_k^{n+1}F_{-k}\exp\left(\sqrt{-1}\frac{2\pi}{N}ik\right)\right]. & \quad \text{(A.17)}
\end{aligned}$$

Dividing equation (A.17) by $C_k^n\exp\left(\sqrt{-1}\,\frac{2\pi}{N}ik\right)$, we obtain

$$\begin{aligned}
C_k\left[1+(\alpha+\beta+r+\lambda)\frac{\Delta\tau}{2}\right]-\frac{\Delta\tau}{2}\beta C_k\exp\left(\sqrt{-1}\,\frac{2\pi}{N}k\right)-\frac{\Delta\tau}{2}\alpha C_k\exp\left(-\sqrt{-1}\,\frac{2\pi}{N}k\right)-C_k\frac{\lambda\Delta\tau}{2}F_{-k} & \\
= & \\
\left[1-(\alpha+\beta+r+\lambda)\frac{\Delta\tau}{2}\right]+\frac{\Delta\tau}{2}\beta\exp\left(\sqrt{-1}\,\frac{2\pi}{N}k\right)+\frac{\Delta\tau}{2}\alpha\exp\left(-\sqrt{-1}\,\frac{2\pi}{N}k\right)+\frac{\Delta\tau}{2}\lambda F_{-k}. & \quad \text{(A.18)}
\end{aligned}$$

Factoring the C_k term, equation A.18 becomes

$$C_k=\frac{\left[1-(\alpha+\beta+r+\lambda)\frac{\Delta\tau}{2}\right]+\frac{\Delta\tau}{2}\beta\exp\left(\sqrt{-1}\,\frac{2\pi}{N}k\right)+\frac{\Delta\tau}{2}\alpha\exp\left(-\sqrt{-1}\,\frac{2\pi}{N}k\right)+\frac{\Delta\tau}{2}\lambda F_{-k}}{\left[1+(\alpha+\beta+r+\lambda)\frac{\Delta\tau}{2}\right]-\frac{\Delta\tau}{2}\beta\exp\left(\sqrt{-1}\,\frac{2\pi}{N}k\right)-\frac{\Delta\tau}{2}\alpha\exp\left(-\sqrt{-1}\,\frac{2\pi}{N}k\right)-\frac{\Delta\tau}{2}\lambda F_{-k}}. \qquad \text{(A.19)}$$

Recalling (A.7), it follows that

$$\begin{aligned}
\alpha+\beta+r+\lambda &= \frac{\sigma^2}{2\Delta x^2}-\frac{\left(r-\lambda\kappa-\frac{1}{2}\sigma^2\right)}{2\Delta x}+\frac{\sigma^2}{2\Delta x^2}+\frac{\left(r-\lambda\kappa-\frac{1}{2}\sigma^2\right)}{2\Delta x}+r+\lambda \\
&= \frac{\sigma^2}{\Delta x^2}+r+\lambda,
\end{aligned}$$

and

$$\begin{aligned}
\Delta\tau\beta\exp\left(\sqrt{-1}\,\frac{2\pi}{N}k\right)+\Delta\tau\alpha\exp\left(-\sqrt{-1}\,\frac{2\pi}{N}k\right) &= \frac{\sigma^2\Delta\tau}{2\Delta x^2}\left[\exp\left(\sqrt{-1}\,\frac{2\pi}{N}k\right)+\exp\left(-\sqrt{-1}\,\frac{2\pi}{N}k\right)\right] \\
&\quad+\frac{\Delta\tau\left(r-\lambda\kappa-\frac{1}{2}\sigma^2\right)}{2\Delta x} \\
&\quad\times\left[\exp\left(\sqrt{-1}\,\frac{2\pi}{N}k\right)+\exp\left(-\sqrt{-1}\,\frac{2\pi}{N}k\right)\right] \\
&= \frac{\sigma^2\Delta\tau}{\Delta x^2}\cos\left(\frac{2\pi}{N}k\right) \\
&\quad+\frac{\sqrt{-1}\,\Delta\tau\left(r-\lambda\kappa-\frac{1}{2}\sigma^2\right)}{\Delta x}\sin\left(\frac{2\pi}{N}k\right).
\end{aligned}$$

Using the above results in (A.19), we find

$$C_k = \frac{\left[1-\left(\frac{\sigma^2}{\Delta x^2}+r+\lambda\right)\frac{\Delta\tau}{2}\right]+\frac{1}{2}\left[\frac{\sigma^2\Delta\tau}{\Delta x^2}\cos\left(\frac{2\pi}{N}k\right)+\sqrt{-1}\,\frac{\Delta\tau}{\Delta x}\left(r-\lambda\kappa-\frac{1}{2}\sigma^2\right)\sin\left(\frac{2\pi}{N}k\right)+\Delta\tau\lambda F_{-k}\right]}{\left[1+\frac{1}{2}\left(\frac{\sigma^2}{\Delta x^2}+r+\lambda\right)\Delta\tau\right]-\frac{1}{2}\left[\frac{\sigma^2\Delta\tau}{\Delta x^2}\cos\left(\frac{2\pi}{N}k\right)+\sqrt{-1}\,\frac{\Delta\tau}{\Delta x}\left(r-\lambda\kappa-\frac{1}{2}\sigma^2\right)\sin\left(\frac{2\pi}{N}k\right)+\Delta\tau\lambda F_{-k})\right]}. \quad (A.20)$$

Letting

$$F^R_{-k} = \mathrm{Re}(F_{-k})$$
$$F^I_{-k} = \mathrm{Im}(F_{-k}),$$

equation (A.20) gives

$$|C_k|^2 = \frac{\left[1-\left(\frac{\sigma^2}{\Delta x^2}+r+\lambda\right)\frac{\Delta\tau}{2}+\frac{\sigma^2\Delta\tau}{2\Delta x^2}\cos\left(\frac{2\pi}{N}k\right)+\frac{\Delta\tau}{2}\lambda F^R_{-k}\right]^2+\left[\frac{\Delta\tau}{2\Delta x}\left(r-\lambda\kappa-\frac{1}{2}\sigma^2\right)\sin\left(\frac{2\pi}{N}k\right)+\lambda\frac{\Delta\tau}{2}F^I_{-k}\right]^2}{\left[1+\left(\frac{\sigma^2}{\Delta x^2}+r+\lambda\right)\frac{\Delta\tau}{2}-\frac{\sigma^2\Delta\tau}{2\Delta x^2}\cos\left(\frac{2\pi}{N}k\right)-\frac{\Delta\tau}{2}\lambda F^R_{-k}\right]^2+\left[\frac{\Delta\tau}{2\Delta x}\left(r-\lambda\kappa-\frac{1}{2}\sigma^2\right)\sin\left(\frac{2\pi}{N}k\right)+\lambda\frac{\Delta\tau}{2}F^I_{-k}\right]^2}, \quad (A.21)$$

or

$$|C_k|^2 = \frac{\left[1-\frac{r\Delta\tau}{2}-\frac{\sigma^2\Delta\tau}{2\Delta x^2}\left(1-\cos\left(\frac{2\pi}{N}k\right)\right)-\frac{\Delta\tau\lambda}{2}\left(1-F^R_{-k}\right)\right]^2+\left[\frac{\Delta\tau}{2\Delta x}\left(r-\lambda\kappa-\frac{1}{2}\sigma^2\right)\sin\left(\frac{2\pi}{N}k\right)+\lambda\frac{\Delta\tau}{2}F^I_{-k}\right]^2}{\left[1+\frac{r\Delta\tau}{2}+\frac{\sigma^2\Delta\tau}{2\Delta x^2}\left(1-\cos\left(\frac{2\pi}{N}k\right)\right)+\frac{\Delta\tau\lambda}{2}\left(1-F^R_{-k}\right)\right]^2+\left[\frac{\Delta\tau}{2\Delta x}\left(r-\lambda\kappa-\frac{1}{2}\sigma^2\right)\sin\left(\frac{2\pi}{N}k\right)+\lambda\frac{\Delta\tau}{2}F^I_{-k}\right]^2}. \quad (A.22)$$

Note that

$$F_{-k} = \frac{X_N}{N}\sum_{j=-\frac{N}{2}+1}^{\frac{N}{2}}\overline{f}_j\exp\left(\sqrt{-1}\,\frac{2\pi}{N}kj\right). \quad (A.23)$$

Then, from (3.6), we have

$$\frac{X_N}{N}\sum_{j=-\frac{N}{2}+1}^{\frac{N}{2}}\overline{f}_j \le 1, \quad (A.24)$$

so that

$$|F_{-k}| \le 1, \quad (A.25)$$

and hence

$$-1 \le F^R_{-k} \le +1. \quad (A.26)$$

It then follows that ($\forall k \in -N/2+1,\ldots,+N/2$)

$$\left|1+\frac{r\Delta\tau}{2}+\frac{\sigma^2\Delta\tau}{2\Delta x^2}\left(1-\cos\left(\frac{2\pi}{N}k\right)\right)+\frac{\Delta\tau\lambda}{2}\left(1-F^R_{-k}\right)\right| \ge \left|1-\frac{r\Delta\tau}{2}-\frac{\sigma^2\Delta\tau}{2\Delta x^2}\left(1-\cos\left(\frac{2\pi}{N}k\right)\right)-\frac{\Delta\tau\lambda}{2}\left(1-F^R_{-k}\right)\right|, \quad (A.27)$$

and consequently $|C_k| < 1, \forall k$, so the scheme is unconditionally strictly stable.

References

Amin, K. I. (1993). Jump diffusion option valuation in discrete time. *Journal of Finance* 48, 1833–1863.

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

Andersen, L. B. G. and R. Brotherton-Ratcliffe (1998). The equity option volatility smile: An implicit finite difference approach. *Journal of Computational Finance* 1, 3–37.

Andersen, T. G., L. Benzoni, and J. Lund (2002). An empirical investigation of continuous-time equity return models. *Journal of Finance* 57, 1239–1284.

Bakshi, G. and C. Cao (2002). Risk-neutral kurtosis, jumps, and option pricing: Evidence from 100 most actively traded firms on the CBOE. Working paper, Smith School of Business, University of Maryland.

Bakshi, G., C. Cao, and Z. Chen (1997). Empirical performance of alternative option pricing models. *Journal of Finance* 52, 2003–2049.

Barone-Adesi, G. and R. E. Whaley (1987). Efficient analytic approximation of American option values. *Journal of Finance* 42, 301–320.

Bates, D. S. (1991). The crash of '87: Was it expected? the evidence from options markets. *Journal of Finance* 46, 1009–1044.

Bates, D. S. (1996). Jumps and stochastic volatility: Exchange rate processes implicit in Deutsche mark options. *Review of Financial Studies* 9, 69–107.

Bates, D. S. (2000). Post-'87 crash fears in the S&P 500 futures option market. *Journal of Econometrics* 94, 181–238.

Borovykh, N. and M. N. Spijker (2000). Resolvent conditions and bounds on the powers of matrices, with relevance to the numerical stability of initial value problems. *Journal of Computational and Applied Mathematics* 125, 41–56.

Broadie, M. and Y. Yamamoto (2002). Application of the fast Gauss transform to option pricing. Working paper, Graduate School of Business, Columbia University.

Carr, P. and L. Wu (2002). Time-changed Lévy processes and option pricing. *Journal of Financial Economics*, forthcoming.

Das, S. R. (2002). The surprise element: Jumps in interest rates. *Journal of Econometrics* 106, 27–65.

Duffie, D., J. Pan, and K. Singleton (2000). Transform analysis and asset pricing for affine jump-diffusions. *Econometrica* 68, 1343–1376.

Duijndam, A. and M. Schonewille (1999). Nonuniform fast Fourier transform. *Geophysics* 64, 539–551.

Dupire, B. (1994). Pricing with a smile. *Risk* 7 (January), 18–20.

Eraker, B., M. Johannes, and N. Polson (2003). The impact of jumps in volatility and returns. *Journal of Finance*, forthcoming.

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

Giles, M. B. (1997). On the stability and convergence of discretizations of initial value PDEs. *IMA Journal of Numerical Analysis* 17, 563–576.

Greengard, L. and J. Strain (1991). The fast Gauss transform. *SIAM Journal on Scientific and Statistical Computing* 12, 79–94.

Heston, S. L. (1993). A closed-form solution for options with stochastic volatility with applications to bond and currency options. *Review of Financial Studies* 6, 327–343.

Johnson, C. (1987). *Numerical Solutions of Partial Differential Equations by the Finite Element Method*. Cambridge University Press.

Jorion, P. (1988). On jump processes in the foreign exchange and stock markets. *Review of Financial Studies* 1, 427–445.

Kou, S. G. (2002). A jump-diffusion model for option pricing. *Management Science* 48, 1086–1101.

Kraaijevanger, J. F. B. M., H. W. J. Lenferink, and M. N. Spijker (1987). Stepsize restrictions for stability in the numerical solution of ordinary and partial differential equations. *Journal of Computational and Applied Mathematics* 20, 67–81.

Lenferink, H. W. and M. N. Spijker (1991). On the use of stability regions in the numerical analysis of initial value problems. *Mathematics of Computation* 57, 221–237.

Lewis, A. L. (2001). A simple option formula for general jump-diffusion and other exponential Lévy processes. Working paper, Envision Financial Systems.

Linetsky, V. (1999). Step options. *Mathematical Finance* 9, 55–96.

Longstaff, F. A. and E. S. Schwartz (2001). Valuing American options by simulation: A simple least squares approach. *Review of Financial Studies* 14, 113–147.

Madan, D. B., P. P. Carr, and E. C. Chang (1998). The variance gamma process and option pricing. *European Finance Review* 2, 79–105.

Matache, A.-M., T. von Petersdorff, and C. Schwab (2002). Fast deterministic pricing of options on Lévy driven assets. RiskLab research report, ETH, Zurich.

Merton, R. C. (1976). Option pricing when underlying stock returns are discontinuous. *Journal of Financial Economics* 3, 125–144.

Metwally, S. A. K. and A. F. Atiya (2002). Using Brownian bridge for fast simulation of jump-diffusion processes and barrier options. *Journal of Derivatives* 10, 43–54.

Meyer, G. H. (1998). The numerical valuation of options with underlying jumps. *Acta Mathematica Universitatis Comenianae* 67, 69–82.

Naik, V. and M. Lee (1990). General equilibrium pricing of options on the market portfolio with discontinuous returns. *Review of Financial Studies* 3, 493–521.

Pan, J. (2002). The jump-risk premia implicit in options: Evidence from an integrated time-series study. *Journal of Financial Economics* 63, 3–50.

Pooley, D. M., P. A. Forsyth, and K. R. Vetzal (2002). Convergence remedies for non-smooth payoffs in option pricing. *Journal of Computational Finance*, forthcoming.

Potts, D., G. Steidl, and M. Tasche (2001). Fast Fourier transforms for nonequispaced data: A tutorial. In J. J. Benedetto and P. J. S. G. Ferreira (Eds.), *Modern Sampling Theory: Mathematics and Application*, Chapter 12, pp. 251–274. Birkhauser.

Rannacher, R. (1984). Finite element solution of diffusion problems with irregular data. *Numerische Mathematik* 43, 309–327.

Richtmyer, R. D. and K. W. Morton (1967). *Difference Methods for Initial-Value Problems*. Interscience Publishers, New York.

Scott, L. O. (1997). Pricing stock options in a jump-diffusion model with stochastic volatility and interest rates: Applications of Fourier inversion methods. *Mathematical Finance* 7, 413–426.

Tavella, D. and C. Randall (2000). *Pricing Financial Instruments: The Finite Difference Method*. John Wiley & Sons, Inc.

Vetzal, K. R. and P. A. Forsyth (1999). Discrete Parisian and delayed barrier options: A general numerical approach. *Advances in Futures and Options Research* 10, 1–16.

Ware, A. F. (1998). Fast approximate Fourier transforms for irregularly spaced data. *SIAM Review* 40, 838–856.

Wilmott, P. (1998). *Derivatives: The Theory and Practice of Financial Engineering*. John Wiley & Sons Ltd., West Sussex, England.

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

Zvan, R., P. A. Forsyth, and K. R. Vetzal (1999). Discrete Asian barrier options. *Journal of Computational Finance* 3, 41–67.