

No Fear of Jumps

Y. d'Halluin

School of Computer Science, University of Waterloo, Waterloo ON, Canada
(e-mail: ydhallui@elora.uwaterloo.ca).

D.M. Pooley

ITO 33 SA, 39 rue Lhomond, 75005 Paris France (e-mail: david@ito33.com).

P.A. Forsyth

School of Computer Science, University of Waterloo, Waterloo ON, Canada
(e-mail: paforsyt@elora.uwaterloo.ca).

Abstract

Jump diffusion based models have recently increased in popularity. In this article, we develop robust and efficient techniques for the numerical solution of option pricing

models where the underlying process is a jump diffusion process. The numerical techniques can be applied to a variety of contingent claim valuations. Numerical examples for European, American and Parisian options are provided.

1 Introduction

In 1973, the Black-Scholes model revolutionized derivative pricing (Black and Scholes, 1973). Using only a volatility and an interest rate, Robert Black and Myron Scholes developed an arbitrage free pricing formula that does not require knowledge of investor beliefs about the underlying stock's expected return. However, over the years practitioners have recognized the limitations of the Black-Scholes model. In particular, the constant volatility assumption is insufficient to capture the smile or skew that is exhibited by the implied volatilities of traded financial options.

To better capture these volatility profiles, numerous avenues of research have been explored which either extend the Black-Scholes model or explore completely new approaches. Among these extensive works, the jump diffusion model (Merton, 1976) and the stochastic volatility model (which could include jumps as well) (Bates, 1996; Scott, 1997; Bakshi et al., 1997) appear to be the most popular among practitioners. Unfortunately, a large portion of the literature devoted to these approaches is limited to analytical or quasi-analytical solutions for vanilla options. Very few of these methods can be extended to price exotic or

path-dependent options. For these more complicated scenarios, numerical partial differential equation techniques must be used.

The objective of this paper is to present a robust and efficient numerical method for solving the partial integro differential equation (PIDE) which arises from the jump diffusion model. We limit ourselves to pricing options under the jump diffusion model, but this framework is also applicable to credit risk models or more complex valuation models such as stochastic volatility with jumps. In the latter case, one simply has to solve a two dimensional PIDE problem, and apply the techniques presented below for the jump diffusion part in the stock direction. A major advantage of the methods introduced here is that they are easily added to existing numerical option pricing software. In particular, software that uses an implicit approach for valuing American options can be easily modified to price American options with jump diffusion.

The title of this paper is obviously based on the very readable article "Fear of Jumps" by Lewis (2002). This article was mostly analytical in nature, and relied on an equilibrium based approach to option pricing. In contrast, the article presented here has a numerical focus for pricing options under jump diffusion. Further, we attempt to convince the reader that adding a jump component to pricing software can be approached with "no fear".

Alternatively, this paper could have been entitled “Fear of No Jumps”, as our examples are intended to show that a jump component adds essential features to a pricing model. Without these features, one should be concerned about the accuracy and stability of the pricing framework.

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), at least in the case of American options.

In this article, the PIDE presented by (Merton, 1976; Andersen and Andreasen, 2000) is studied exclusively. While it is true that Merton’s assumption about jump risk being diversifiable does not hold for index based options, and in this case one must use an equilibrium based method (Lewis, 2002) or a mean variance hedging approach (Ayache et al., 2004), the PIDEs resulting in either case are essentially identical. Consequently, the numerical techniques presented here can be applied.

This article is organized as follows. In section 2, the numerical method for solving the option pricing PIDE which results from a jump diffusion model is presented. In section 3, a wide variety of numerical examples of exotic, path-dependent contracts are presented. In particular, we include numerical examples for American, and Parisian options. Finally, section 4 contains concluding remarks.

2 Mathematical Model

This section provides an overview of the mathematical modeling issues that arise in a jump diffusion framework. The presentation and notation closely follows that of d’Halluin et al. (2003). However, particular attention is paid here to the practical issues that arise in a numerical implementation. Further, since the goal of this paper is somewhat illustrative, several proofs and technical details have been omitted. The reader is referred to d’Halluin et al. (2003) and the references therein for a complete treatment of the theory of option pricing in a jump diffusion framework.

In the usual (no jumps) Black-Scholes model for option pricing (Black and Scholes, 1973; Merton, 1976), the underlying asset price S evolves according to

$$\frac{dS}{S} = \mu dt + \sigma dZ, \quad (2.1)$$

where μ is the (real) drift rate, σ is the volatility, and dZ is the increment of a Gauss-Wiener process. Let $V(S, t)$ be the value of a contingent claim that depends on the underlying asset S and time t . By appealing to the principle of no-arbitrage, a partial differential equation (PDE) for the value of V can be derived:

$$V_\tau = \frac{1}{2}\sigma^2 S^2 V_{SS} + rSV_S - rV, \quad (2.2)$$

where $\tau = T - t$ is the time remaining until expiry T , and r is the continuously compounded risk-free interest rate. Equation (2.2) is simply a second order parabolic PDE of one space dimension and one time dimension.

This equation has been the subject of countless studies, and is well-understood from a variety of viewpoints (financial, mathematical, numerical). Letting

$$\mathcal{L}V = \frac{1}{2}\sigma^2 S^2 V_{SS} + rSV_S - rV \quad (2.3)$$

equation (2.2) can be written in the simple form

$$V_\tau = \mathcal{L}V. \quad (2.4)$$

It is assumed that the reader is familiar with the numerical solution of PDEs of the form (2.4). Software for this problem is easily written, and off-the-shelf implementations are readily available.

Nevertheless, the process specified by equation (2.1) is not sufficient to explain observed market behaviour (Bakshi and Cao, 2002). In reality, stock prices have been observed to have large instantaneous jumps. Such behaviour can be modeled by the risk-neutral process (Merton, 1976)

$$\frac{dS}{S} = (r - \lambda\kappa) dt + \sigma dZ + (\eta - 1)dq, \quad (2.5)$$

where dq is a Poisson process (independent of the Brownian motion), and $\eta - 1$ is an impulse function producing a jump from S to $S\eta$. If λ is the arrival intensity of the Poisson process, then $dq = 0$ with probability $1 - \lambda dt$, and $dq = 1$ with probability λdt . The expected jump size can be denoted by $\kappa = E[\eta - 1]$, where E is the expectation operator.

As is well known, the fair price of a contingent claim $V(S, t)$ under a process of the form (2.5) is given by the following partial integro differential equation (PIDE):

$$V_\tau = \frac{1}{2}\sigma^2 S^2 V_{SS} + (r - \lambda\kappa)SV_S - rV + \lambda \int_0^\infty V(S\eta)g(\eta)d\eta - \lambda V. \quad (2.6)$$

In equation (2.6), $g(\eta)$ is the probability density function of the jump amplitude η . The probability density function is assumed to have the usual distribution properties, such as $\forall \eta, g(\eta) \geq 0$ and $\int_0^\infty g(\eta)d\eta = 1$. Letting

$$\hat{\mathcal{L}}V = \frac{1}{2}\sigma^2 S^2 V_{SS} + (r - \lambda\kappa)SV_S - (r + \lambda)V, \quad (2.7)$$

equation (2.6) can be written as

$$V_\tau = \hat{\mathcal{L}}V + \lambda \int_0^\infty V(S\eta)g(\eta)d\eta. \quad (2.8)$$

As with $\mathcal{L}V$, the behaviour of $\hat{\mathcal{L}}V$ is well understood. Further, it should be straightforward to modify any reasonably designed software that can handle numerically $\mathcal{L}V$ to compute $\hat{\mathcal{L}}V$. Of a more difficult nature is the integral term in equation (2.8).

The obvious approach for the numerical computation of the integral term is to use standard numerical integration methods such as Simpson’s rule or Gaussian quadrature. Unfortunately, for a numerical grid of size n , these techniques are $\mathcal{O}(n^2)$. For real-time pricing software, and especially for calibration routines, quicker algorithms are desirable.

To this end, the integral term of equation (2.8) should be computed in a way that is

- efficient (better than $\mathcal{O}(n^2)$),
 - robust,
 - flexible (can be used with nonlinear pricing models, and/or exotic options),
 - easily added to existing option pricing software.
- All of these properties are satisfied if
- the integral term is evaluated by FFTs, thereby only requiring $\mathcal{O}(n \log n)$ operations per timestep,
 - the integral term is applied implicitly, thereby increasing stability and allowing the possibility of second order convergence.

The FFT evaluation of the integral and the implicit treatment of the resulting terms will be discussed separately below. Following these, an extension to American options will be provided, as well as a brief description of credit risk. Examples which use the techniques described below are provided in section 3.

It should be noted that in some cases, the integral term can be evaluated directly in $\mathcal{O}(n)$ time using fast Gauss transform (FGT) techniques (Greengard and Strain, 1991). While this technique works for the case where jump sizes are lognormally distributed, it is not clear if it works for more general distributions. Furthermore, numerical experiments show that for any practical grid size the FFT approach for evaluating the integral term is faster than the FGT method. (Note that the integral needs only to be evaluated with an accuracy consistent with the discretization of the PDE).

2.1 FFT evaluation

Before the integral term of equation (2.8) can be evaluated by FFTs, it must be manipulated into the form of a correlation integral. This process is described in section 2.1.1. Once this process is done, at least two numerical issues remain. First, standard FFT algorithms require an equally spaced grid, whereas an efficient PDE grid will be unequally spaced. Interpolation must be used to move from one grid to the other. Second, since the input functions to the FFT routines will be non-periodic, wrap-around pollution can negatively affect the solution. These numerical issues are discussed in section 2.1.2.

2.1.1 Manipulation

Ignoring the leading λ , the integral term in equation (2.8) is

$$I(S) = \int_0^\infty V(S\eta)g(\eta)d\eta. \quad (2.9)$$

The goal is to turn this expression into a correlation product which can be evaluated by FFT techniques. Letting $x = \log(S)$ and applying the change of variable $y = \log(\eta)$, we obtain

$$I = \int_{-\infty}^{+\infty} \bar{V}(x+y)\bar{f}(y)dy, \quad (2.10)$$

where $\bar{f}(y) = g(e^y)e^y$ and $\bar{V}(y) = V(e^y)$. The $\bar{f}(y)$ term can be interpreted as the probability density of a jump of size $y = \log \eta$. Conveniently, equation (2.10) corresponds to the correlation product $\bar{V}(y) \otimes \bar{f}(y)$. In discrete form, equation (2.10) becomes

$$I_i = \sum_{j=-N/2+1}^{j=N/2} \bar{V}_{i+j}\bar{f}_j\Delta y + \mathcal{O}((\Delta y)^2), \quad (2.11)$$

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

$$\bar{f}_j = \bar{f}(j\Delta y) = \frac{1}{\Delta x} \int_{x_j-\Delta x/2}^{x_j+\Delta x/2} \bar{f}(x)dx \quad (2.12)$$

It has been assumed that $\Delta y = \Delta x$.

Assuming that \bar{f} is real (a safe assumption for financial applications), the discrete correlation of equation (2.11) can be evaluated using FFTs since

$$I_i = \text{IFFT} \left((\text{FFT}(\bar{V})) (\text{FFT}(\bar{f}))^* \right)_i \quad (2.13)$$

where $(\cdot)^*$ denotes the complex conjugate. For efficiency, $\text{FFT}(\bar{f})$ can be pre-computed and stored. During each timestep (or each iteration of an iterative method), an FFT and an inverse FFT must be computed.

2.1.2 Numerical issues

A typical grid for the discretization of $\hat{\mathcal{L}}V$ in equation (2.8) will be unequally spaced in S coordinates. For example, small mesh spacing will be used near strikes or barriers, with large mesh spacing elsewhere. However, the discrete form of the correlation integral (2.11) requires an equally spaced grid in $\log(S)$ coordinates. It is highly unlikely that these two grids are fully compatible. Hence, values must be interpolated between the two grids.

In particular, values of V on the unequally spaced S grid must be interpolated onto an equally spaced $\log(S)$ grid. The computation of equation (2.13) can then be performed¹. Finally, the resulting equally spaced \bar{V} data needs to be interpolated back onto the unequally spaced S grid. The overall process is summarized in algorithm (1). If linear or

Algorithm 1 Method for computing the integral term of equation (2.8) by FFTs.

Interpolate the discrete values of V onto an equally spaced $\log(S)$ grid. This generates the required values of \bar{V}_j .

Carry out the FFT on the \bar{V} data

Compute the correlation in the frequency domain (with pre-computed $\text{FFT}(\bar{f})$ values), using equation (2.13).

Invert the FFT of the correlation

Interpolate the discrete values of $I(x_i)$ back onto the original S grid

higher order interpolation is used, algorithm (1) is second order correct. This is consistent with the discretization error in the PDE and the midpoint rule used to evaluate the integral in equation (2.11).

For the actual FFT evaluation, standard algorithms assume periodic input data. If the input data is not periodic (as with the current application), then the discrete Fourier transform is effectively applied to the periodic extension of the input functions. This can lead to undesirable “wrap around pollution”, which manifests itself with erroneous values in the solution.

To avoid wrap around effects, the domain of the integral in equation (2.8) can be extended to the left and right by amounts Δy^- and Δy^+ . The integral then becomes

$$I_{\text{ext}} = \int_{y_{\min} - \Delta y^-}^{y_{\max} + \Delta y^+} \bar{V}(x+y) \bar{f}(y) dy, \quad (2.14)$$

where $y_{\max} = \log(S_{\max})$, $y_{\min} = \log(S_{\min})$, and $[S_{\min}, S_{\max}]$ are selected appropriately. Unknown values in the range $[y_{\max}, y_{\max} + \Delta y^+]$ can be obtained by linear extrapolation. This assumes that the far field behaviour of the option pricing problem is linear. Values in the range $[y_{\min} - \Delta y^-, y_{\min}]$ can be obtained from interpolation on the original S grid, assuming an $S_0 = 0$ grid point has been maintained.

Once the FFT has been performed in the extended domain, values in the extensions are discarded. Because of the extension, values in the original domain will have been less affected by wrap-around pollution.

2.2 Implicit evaluation

We now look at the numerical evaluation of equation (2.8). Let Ψ_i^n denote the discrete form of the integral evaluated at timestep n using data V^n (one can think of Ψ as an application of algorithm (1)). To solve equation (2.8), the $\hat{\mathcal{L}}V$ term must also be discretized. This can be done by any standard method, such as finite differences, finite volumes, or finite elements. Let the discrete form of $\hat{\mathcal{L}}V$ at timestep n be given by $(\hat{\mathcal{L}}V)_i^n$. A general discretized form of equation (2.8) can then be written as

$$\frac{V_i^{n+1} - V_i^n}{\Delta \tau} = (1 - \theta) (\hat{\mathcal{L}}V)_i^{n+1} + \theta (\hat{\mathcal{L}}V)_i^n + (1 - \theta_j) \lambda \Psi_i^{n+1} + \theta_j \lambda \Psi_i^n, \quad (2.15)$$

where

- θ is a time-weighting parameter for $\hat{\mathcal{L}}$
- $\theta = 0$ is fully implicit
- $\theta = 1/2$ is Crank-Nicolson
- $\theta = 1$ is fully explicit
- θ_j is time-weighting for the jump term Ψ
- $\theta_j = \{0, 1/2, 1\}$.

Let M denote the discretization matrix stencil such that

$$-[MV]_i^n = (\hat{\mathcal{L}}V)_i^n. \quad (2.16)$$

Equation (2.15) becomes

$$[I - \Delta \tau(1 - \theta)M]V^{n+1} = [I + \Delta \tau \theta M]V^n + (1 - \theta_j) \lambda \Delta \tau \Psi_i^{n+1} + \theta_j \lambda \Delta \tau \Psi_i^n. \quad (2.17)$$

For standard PDE discretization techniques, the matrix M in equation (2.17) is tridiagonal. Tridiagonal systems are quick and easy to solve. However, an implicit treatment of the jump term ($\theta_j \neq 1$) causes Ψ_i^{n+1} to lead to a highly undesirable dense matrix (all nodal values are coupled in equation (2.10)). On the other hand, a fully explicit treatment of the jump term is easy to adapt to existing code, since only the right hand side vector needs to be updated. However, while still stable, only first order convergence is possible.

To allow for an implicit treatment of jumps, a fixed point iteration method must be used. A description of the method is given in algorithm (2). At iteration k known data is used to construct the jump term. Since only the right hand side is affected, a simple tridiagonal system needs to be solved at each iteration.

Algorithm 2 Fixed point iteration.

```

Let  $(V^{n+1})^0 = V^n$ 
Let  $\hat{V}^k = (V^{n+1})^k$ 
Let  $\hat{\Psi}^k = (\Psi^{n+1})^k$ 
Construct vector  $\Psi^n$  using algorithm (1)
for  $k = 0, 1, 2, \dots$  until convergence do
    Construct vector  $\hat{\Psi}^k$  using algorithm (1)
    Solve  $[I - (1 - \theta)M]\hat{V}^{k+1} = [I + \theta M]V^n + (1 - \theta_j)\lambda\hat{\Psi}^k + \lambda\theta_j\Psi^n$ 
    if  $\max_i \frac{|\hat{V}_i^{k+1} - \hat{V}_i^k|}{\max(1, |\hat{V}_i^{k+1}|)}$  < tolerance then
        quit
    end if
end for
    
```

Under some fairly mild assumptions - that the discretization of $\hat{\mathcal{L}}$ form an M -matrix, the probability density function has certain standard properties, the interpolation weights are positive, and that r and λ are positive - it can be proven that algorithm (2) is globally convergent (d'Halluin et al., 2003). Further, the error at each iteration is reduced by approximately $(1 - \theta)\lambda\Delta\tau$, indicating convergence in a small number of iterations (i.e. for typical values, 3 iterations are sufficient).

2.3 American options

American options can be solved by a simple penalty approach. Details of the penalty approach can be found in (Forsyth and Vetzal, 2002). Further details with regards to jump diffusion models can be found in (d'Halluin et al., 2003). Briefly, the penalty approach involves adding a penalty term

to the pricing PDE. Equation (2.8) then becomes

$$V_\tau = \hat{\mathcal{L}}V + \lambda \int_0^\infty V(S\eta)g(\eta)d\eta + \rho \max(V^* - V, 0). \quad (2.18)$$

In the limit as $\rho \rightarrow \infty$, the solution satisfies $V \geq V^*$. The American constraint is enforced by setting V^* to the payoff of the option.

In the discrete equations, ρ is set independently at each node. If the value at a node i drops below V_i^* (the payoff), then ρ_i is set to a large number. This essentially adds an extra source term to the PDE, thereby increasing the value at the particular node. If the value at a node is greater than V^* , then ρ_i is set to zero, and the regular PDE is solved. This can also be thought of as constraint switching. Wherever the value drops below the V^* threshold, the constraint is switched on and applied. If the value is above the threshold, the constraint is switched off.

As with the evaluation of the integral term, the penalty constraint can be applied explicitly or implicitly. An explicit evaluation simply uses data at the previous timestep to determine when the constraint is activated. An implicit evaluation could use a fixed point iteration (or other nonlinear solving method) to apply the constraint using data at the current timestep. If the jump term is already being evaluated using an iterative method, little or no extra cost is incurred by the penalty method. Convergence of the penalty approach for American options in a jump diffusion framework was proven in (d'Halluin et al., 2003).

2.4 Credit risk

Until this point, jumps in stock price associated with the jump diffusion model have been assumed to occur for arbitrary exceptional events. However, a special jump in asset level occurs in the case of bankruptcy. In pricing corporate and convertible bonds, it is of interest to determine the risk adjusted hazard rate of bankruptcy. If it is assumed that the stock price of a firm jumps to zero on default, then λ_h can be interpreted as the risk adjusted hazard rate of bankruptcy (or default in the case of bonds). In this case, the PDE satisfied by vanilla puts/calls in the presence of a single jump to bankruptcy is given by

$$V_\tau = \frac{1}{2}\sigma^2 S^2 V_{SS} + (r + \lambda_h)SV_S - (r + \lambda_h)V + \lambda_h V(0, \tau). \quad (2.19)$$

Equation (2.19) can be derived by hedging arguments, or by setting κ to -1 and the jump probability density function $g(\eta)$ to the delta function $\delta(0)$ (1 at $\eta = 0$, zero elsewhere) in the usual Merton jump diffusion model.

It is usually assumed that $\lambda_h = \lambda_h(S, t)$, with $\lambda_h(S, t)$ being determined by calibration to observed market prices for vanilla options and credit instruments. Since option prices are usually available for a range of strikes, more information is provided about default rates than is usually available from simply examining credit instruments. Note that equation (2.19) suggests that default risk has an effect on the pricing of vanilla options. As well, if the possibility of a single jump to bankruptcy is assumed, then a hedging portfolio consisting of the option, an underlying asset, and an additional option can be constructed which eliminates both the diffusion risk (a delta hedge) as well as the jump risk (since the jump has only one possible outcome).

3 Results

The examples of this section are intended to compare the regular Black-Scholes model and the jump-diffusion model. To ensure a consistent basis for comparison, the following procedure is used:

1. Given some jump diffusion parameters, compute the (numerical) at-the-money price V_{jump} of European put option.
2. Using a constant volatility Black-Scholes model, determine the implied volatility σ_{implied} which matches the option price to the jump diffusion value V_{jump} at the strike K .
3. Value the option using a constant volatility model (no jumps) using the implied volatility σ_{implied} estimated in Step 2.

The first example prices a European put option with and without jumps. Parameters are provided in Table 1. Results are shown in Figure 1. The implied volatility value for the Black-Scholes model is 0.1886. By construction, the prices of the Black-Scholes model and the Merton jump model are equal at the strike price. In-the-money values are larger for the Black-Scholes model, but only slightly. Of interest is the fact that the jump model prices deep out-of-the-money options significantly higher. This reflects the fact that a jump event can dramatically change the moneyness of an option to a much larger extent than a simple diffusion only model.

The delta and gamma plots for the two models are similar, although the jump model plots show greater variation. This indicates that a delta hedge of the jump model may need more frequent rebalancing. Nevertheless, jumps introduce market incompleteness, and simple delta hedging will definitely fail. Optimal hedging in incomplete markets is preferred (Henrotte, 2002; Ayache et al., 2004). In any case, hedging will require accurate delta and gamma information. It is essential that the numerical scheme produce smooth delta and gamma values.

TABLE 1. INPUT DATA USED TO VALUE VARIOUS 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

volatility: σ	0.15
risk-free rate: r	0.05
jump standard deviation: γ	0.45
jump mean: μ_{mean}	-0.90
jump intensity: λ	0.10
time to expiry: T	0.25
strike: K	1.00

The second example is a repeat of the first, except that an American put option is priced instead of a European put option. The implied volatility value used is the same as in the previous example: $\sigma_{\text{implied}} = 0.1886$. Results are similar, except that delta values now reach and remain at -1 for low stock prices, while gamma values jump to

zero. This jump to zero occurs at the free boundary between the early-exercise region and the regular pricing region. The early exercise region is further to the right for the Merton jump model, indicating that jumps cause an increase in the probability that the option should be exercised early.

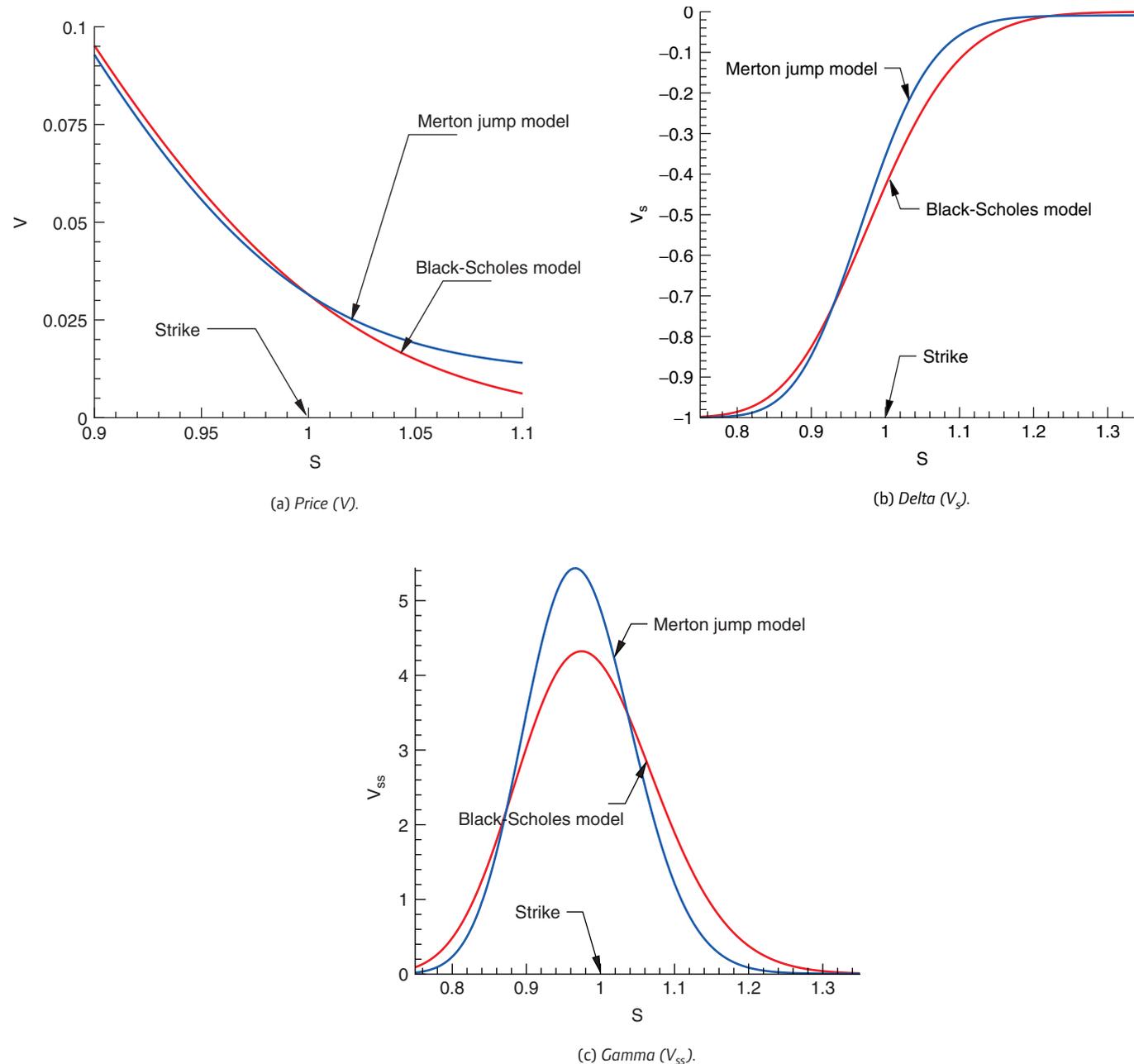


Figure 1: Put option price (V), delta (V_s) and gamma (V_{ss}). The input data is contained in Table 1

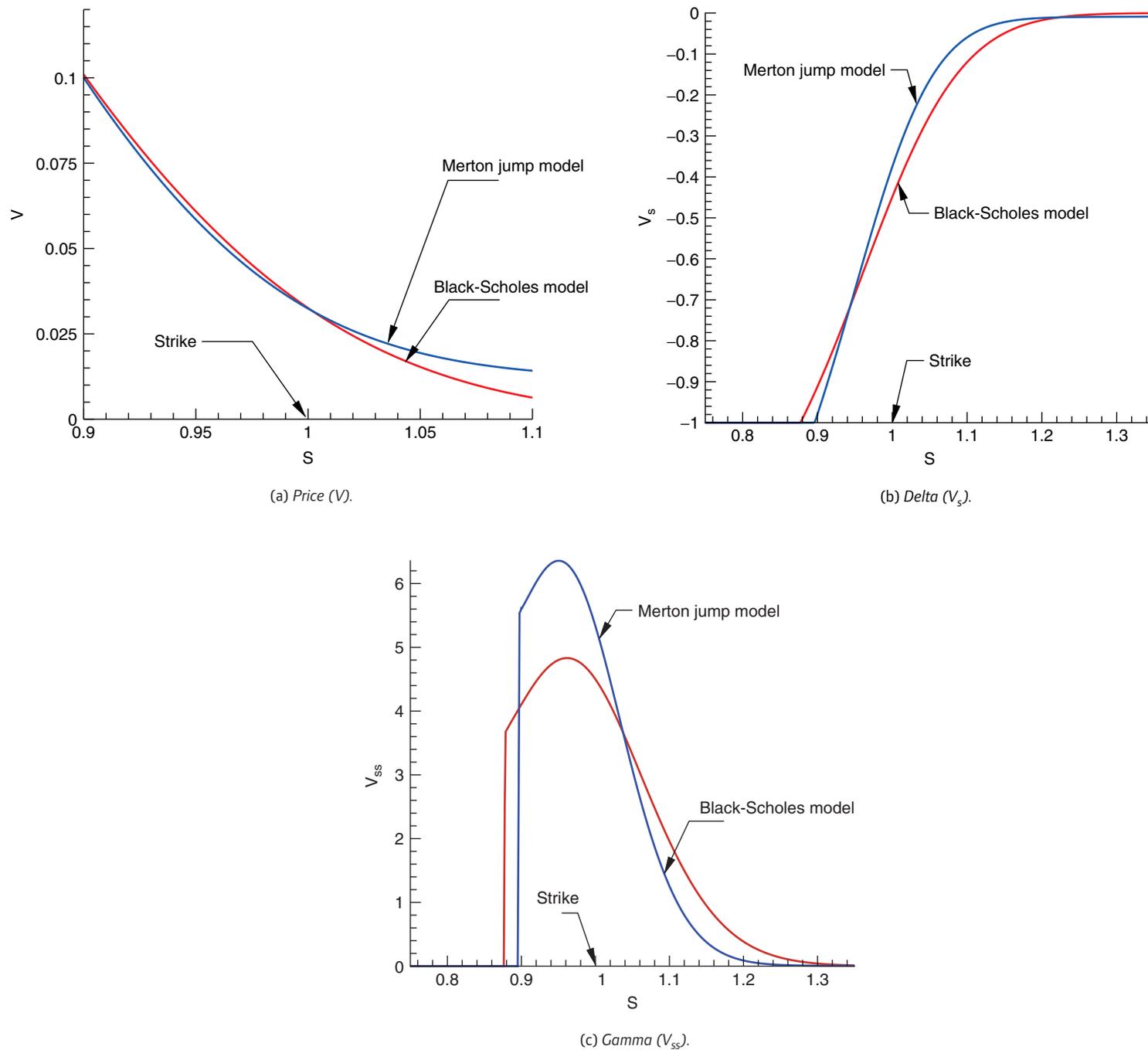


Figure 2: American put option price (V), delta (V_S) and gamma (V_{SS}). The input data is contained in Table 1

The last example is for a Parisian knock-out call option. The particular case considered 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. This can be

valued by solving a set of one-dimensional problems which exchange information at monitoring dates (Vetzal and Forsyth, 1999). Base parameters are the same as in Table 1. The knock-out barrier is placed at $S = 1.20$, while the number of consecutive days above the barrier until knock-out is

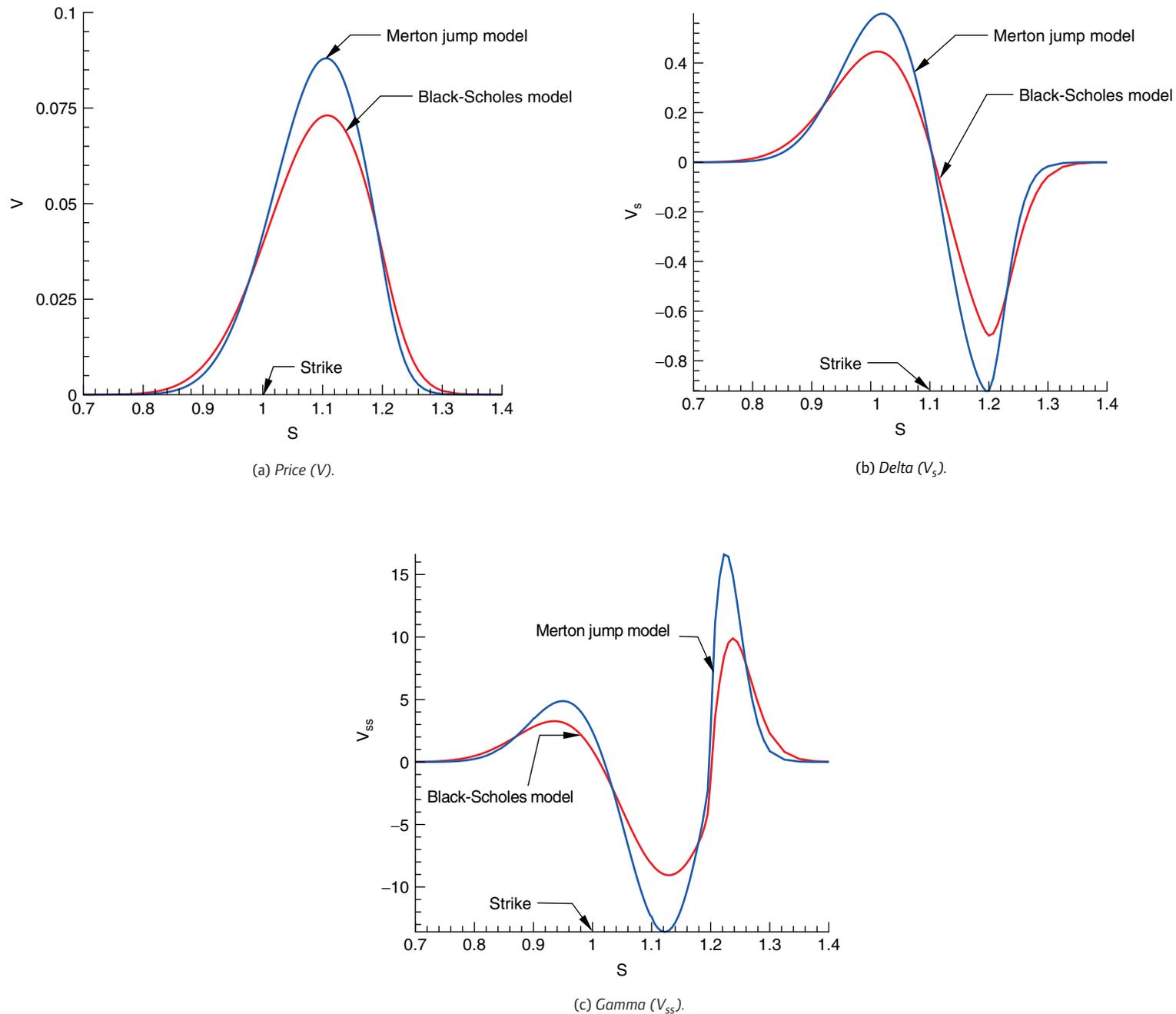


Figure 3: Parisian knock-out call option (V), delta (V_s) and gamma (V_{ss}) with discrete daily observation dates with and without jumps. The barrier is set at $S = 1.20$ and the number of consecutive daily observations to knock-out is 10. The input data is contained in Table 1

set to 10. The implied volatility value is 0.1886. It is interesting to note that the Merton jump model gives smaller prices for stock values below the strike and above the barrier. This is somewhat in contradiction to the

put options, for which deep out-of-the-money prices were higher for the jump model. Nevertheless, the differences are small, and the delta and gamma plots show the far field behaviour to be quite similar.

The greatest price difference occurs between the strike and barrier levels. Presumably a jump in this region hides the effect of the (upper) barrier, whereas a pure diffusion model will have its value decreased by the barrier. However, it is difficult to intuitively predict the effect of jumps on prices. For convex payoffs, jumps increase the value of an option. For non-convex payoffs, as is the case for the Parisian knock-out call, it is not clear what effect jumps will have on the price.

4 Conclusion

This article has demonstrated the numerical evaluation of the PIDE resulting from the Merton jump-diffusion model in option pricing. The integral term of the pricing equation was evaluated using efficient FFT techniques. The issues of interpolation between unequally spaced PDE grids and equally spaced FFT grids, as well as wrap-around pollution effects, were briefly discussed. A fixed point iteration method was used to obtain an implicit timestepping method without resorting to a full dense matrix solve. Extensions to American options and credit risk were also mentioned.

Perhaps the biggest advantage of the techniques described in this paper is the ease with which they can be added to an existing exotic option pricing library. 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. This method can even be applied to any jump size probability density function.

The numerical examples showed the effect of jumps on various option values. For European and American put options, the jump diffusion model increases deep out-of-the-money prices. Changes to the hedging parameters—delta and gamma—were also noted. The stability of the methods was alluded to by the smooth delta and gamma plots. An example of a Parisian knock-out option was also provided.

An important issue not addressed in this paper is hedging jump diffusion models. Since the market is incomplete, simple delta hedging can give large errors. In this case optimal hedging in incomplete markets must be used (Ayache et al., 2004).

FOOTNOTE & REFERENCES

1. Methods exist for computing an FFT on unequally spaced data. However, these methods do not appear to be more efficient than the straight forward approach suggested here.
 - 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.
 - Ayache, E., P. Henrotte, S. Nassar and X. Wang (2004, January). Can anyone solve the smile problem? To appear in *Wilmott* magazine.
 - 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.
 - Bates, D. S. (1996). Jumps and stochastic volatility: Exchange rate processes implicit in Deutsche mark options. *Review of Financial Studies* 9, 69—107.
 - Black, F. and M. Scholes (1973). The pricing of options and corporate liabilities. *Journal of Political Economy* 81, 637—659.
 - d’Halluin, Y., P. A. Forsyth and G. Labahn (2003). A penalty method for American options with jump diffusion processes. www.scicom.uwaterloo.ca/~ydhallui/jump_amer.pdf, accepted to be published in *Numerische Mathematik*.
 - d’Halluin, Y., P. A. Forsyth, and K. R. Vetzal (2003). Robust numerical methods for contingent claims under jump diffusion processes. www.scicom.uwaterloo.ca/~ydhallui/jump.pdf, submitted to *IMA Journal of Numerical Analysis*.
 - 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.
 - Greengard, L. and J. Strain (1991). The fast Gauss transform. *SIAM Journal on Scientific and Statistical Computing* 12, 79—94.
 - Henrotte, P. (2002, July). Dynamic mean variance analysis. Working paper.
 - Lewis, A. (2002, December). Fear of jumps. In *Wilmott*, pp. 60—67. Wiley.
 - Merton, R. C. (1976). Option pricing when underlying stock returns are discontinuous. *Journal of Financial Economics* 3, 125—144.
 - 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.
 - 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.
 - Zhang, X. L. (1997). Numerical analysis of American option pricing in a jump-diffusion model. *Mathematics of Operations Research* 22, 668—690.