Option valuation in a stochastic volatility jump-diffusion model

Fernanda D’Ippoliti\textsuperscript{a}, Enrico Moretto\textsuperscript{a,b}, Sara Pasquali\textsuperscript{a}, and Barbara Trivellato\textsuperscript{a,c}

\textsuperscript{a}CNR-IMATI, Via Bassini 15, IT - 20133 Milano
\textsuperscript{b}University of Parma, Via J.F. Kennedy 6, IT - 43100 Parma
\textsuperscript{c}Polytechnic University of Turin, Corso Duca degli Abruzzi 24, IT - 10129 Turin

Abstract

We extend the stochastic volatility model in Moretto et al. [MPT05] to a stochastic volatility jump-diffusion model. We provide a closed-form solution for the price of European-type options and develop a Monte Carlo method to approximate the price of more complex options.

Key words: Monte Carlo simulation, option pricing, stochastic volatility jump-diffusion models

AMS: 91B28

1 Introduction

Heston [Hes93] introduced a path-breaking methodology to obtain semi-closed formulae of European-style option derivatives when the underlying is characterized, for instance, by stochastic volatility. Bakshi and Madan [BM00] proposed a general technique that encompasses Heston’s result and allows to evaluate a broad class of derivative assets under a general framework. These developments are due to the fact that there is a huge empirical evidence that volatility is far from being constant as in Black and Scholes [BS73]. For instance, in the last decade, there has been convincing evidence that stochastic volatility processes with jumps in returns are important to model index return and volatility (Bates [Bat96]) appropriately, but they are incapable of fully capturing the empirical features of equity index returns and option prices (Bakshi et al. [BCC97], Bates [Bat00] and Pan [Pan02]). As stated in Eraker et al. [EJP03], empirical evidence indicates that the stochastic volatility and jumps in the underlying are not sufficient to reproduce real market behavior of returns such as, for example, Bates [Bat00], Duffie et al. [DPS00] and Pan [Pan02]. One way to tackle this issue is to introduce jumps into the volatility dynamics as well, as in Eraker et al. [EJP03]). We generalize stochastic volatility model Moretto et al. [MPT05], that, in the spirit of Heston [Hes93], it admits a closed-form solution
for European-style options. It is well known that, unfortunately, for more complex options, a closed-form solution does not exist anymore. Following Broadie and Kaya [BK06], we describe a Monte Carlo method to approximate their values. Broadie and Kaya propose an “exact simulation” of the jump-diffusion processes: loosely speaking, their technique consists in generating a sample from the final value of the variance, and then, using Fourier inversion methods, they get a sample from the integral of the variance. These two quantities allow to generate a sample for the stock price. Differently from Broadie and Kaya, we generate the integral of the variance through a rejection sampling algorithm.

The method is applied to the DJ Euro Stoxx 50 market and we base the test of goodness of fit on European options. We plan to evaluate barrier and American options in a future work.

2 Stochastic Volatility Jump-diffusion Model

Let \((\Omega, \mathcal{F}, \mathbb{P})\) a complete probability space where \(\mathbb{P}\) is the historical measure and consider \(t \in [0,T]\). We suppose that a bidimensional standard Wiener process \(W = (W_1, W_2)\) and two compound Poisson processes \(Z_S\) and \(Z_v\) are defined. We assume that \(W_1, W_2, Z_S\) and \(Z_v\) are mutually independent. Under \(\mathbb{P}\), the dynamics of the stock price \(S(t)\) is

\[
dS(t) = S(t^-) \left[ \mu \, dt + \sigma_S \, dW_1(t) + \xi \sqrt{v(t)} \, dW_2(t) + dZ_S(t), \right],
\]

(1)

where \(\sqrt{v(t)}\) is the volatility process whose dynamics is specified later, and the parameters \(\mu, \sigma_S, \xi \in \mathbb{R}\) are real constants. The process \(Z_S(t)\) has constant intensity \(\lambda > 0\) (annual frequency of jumps) and log-normal distribution of jump sizes, that is, denoting with \(J_S\) the relative jump size, then \(\log(1 + J_S)\) is distributed according to the \(\mathcal{N}(\log(1 + j_S) - \frac{1}{2} \delta_S^2, \delta_S^2)\) law, where \(j_S\) is the unconditional mean.

Note that \(J_S > 0\) implies that the stock price remains positive for all \(t \in [0,T]\).

The process \(v(t)\) evolves according to

\[
dv(t) = k^*(\theta^* - v(t)) \, dt + \sigma_v \sqrt{v(t)} \, dW_2(t) + dZ_v(t),
\]

(2)

where \(Z_v(t)\) has the same intensity \(\lambda\) of \(Z_S\) with exponentially distributed jump size \(J_v\) with mean \(j_v\), and the parameters \(k^*, \theta^*\) and \(\sigma_v\) are real constants.

Variance \(v(t)\) is a mean reverting process where \(k^*, \theta^*\) and \(\sigma_v\) are respectively the speed of adjustment, the long-run mean and the variation coefficient. If \(k^*, \theta^*, \sigma_v > 0, 2k^* \theta^* \geq \sigma_v^2, v(0) \geq 0\) and \(J_v > 0\), then the process \(v(t)\) is positive for all \(t \in [0,T]\), with probability 1 (Lamberton and Lapeyre [LL97]). Jumps in both asset price and variance occur concurrently and the counting process will be denoted by \(N_t\).
Note that the model (1)-(2) is a generalization of the stochastic volatility model presented in Moretto et al. [MPT05], obtained introducing jumps in the dynamics of both the underlying and the volatility.

It is well known that the market is incomplete. Consequently, the principle of absence of arbitrage does not lead to a uniquely defined price. One obtains actually an entire range of prices and the structure of the investors choice has to come into play to determine the pricing measure or, equivalently, the market price of risk. As in Duffie et al. [DPS00], we make an ad-hoc choice of the market price of risk such that a risk-neutral measure \( \mathbb{Q} \) equivalent to \( \mathbb{P} \) exists. This choice involves the following form of the jump transform \( \zeta(c_1, c_2) \) of the bivariate jump-size distribution \( (J_Y, J_v) \):

\[
\zeta(c_1, c_2) = \frac{1}{2} \left( \exp \left\{ \left( \log(1 + jS) - \frac{1}{2} \delta^2_S \right) c_1 + \frac{1}{2} \delta^2_S c_1^2 \right\} + \frac{1}{1 - j \lambda c_2} \right),
\]

where \( c_1, c_2 \in \mathbb{R} \). Under \( \mathbb{Q} \), the processes \( S \) and \( v \) evolve according to

\[
dS(t) = S(t^-) \left[ (r - \lambda jS) \, dt + \sigma_S \, dW_1(t) + \xi \sqrt{v(t)} \, dW_2(t) + dZ_S(t) \right],
\]

\[
dv(t) = k^*(\theta^* - v(t)) \, dt + \sigma_v \sqrt{v(t)} \, dW_2(t) + dZ_v(t),
\]

where \( r \) is the riskless rate.

### 3 A closed formula for European-style options

In this section we derive a closed-form solution for the price \( C(S, v, t) \) of a European call option with strike price \( K \) and maturity \( T \) written on the underlying asset \( S \). The price of a European put option written on a non-paying dividend underlying asset is then obtained by applying the put-call parity

\[
P(S, v, t) = C(S, v, t) - S + Ke^{r(T-t)},
\]

where \( r \) is the riskless rate.

Assuming that, under \( \mathbb{Q} \), \( S \) and \( v \) evolve according to (4) and (5) respectively, the pricing equation for the value of any asset \( U = U(S, v, t) \) is

\[
\frac{1}{2} \left( \sigma_S^2 + \xi^2 v \right) S^2 \frac{\partial^2 U}{\partial S^2} + \xi \sigma_v \sigma_S S \frac{\partial^2 U}{\partial S \partial v} + \frac{1}{2} \sigma_v^2 v \frac{\partial^2 U}{\partial v^2} + (r - \lambda jS) S \frac{\partial U}{\partial S}
\]

\[
+ \left[ k^* (\theta^* - v) \right] \frac{\partial U}{\partial v} - rU + \frac{\partial U}{\partial t} + \lambda U \left( \zeta(S, v) - 1 \right) = 0,
\]

where the jump transform \( \zeta \) is defined by (3). By analogy with the Black-Scholes and Heston formulæ, and following the scheme in Moretto et al. [MPT05], the European call option turns out to be of the form

\[
U(S, v, t) = SP_1(S, v, t) - Ke^{-r(T-t)}P_2(S, v, t),
\]

where
After some algebra, (9) and (10) become
\[ \frac{\partial U_1}{\partial S} = P_1 + S \frac{\partial P_1}{\partial S}, \quad \frac{\partial U_2}{\partial S} = Ke^{-r(T-t)} \frac{\partial P_2}{\partial S}, \]
\[ \frac{\partial^2 U_1}{\partial S^2} = 2 \frac{\partial P_1}{\partial S} + S \frac{\partial^2 P_1}{\partial S^2}, \quad \frac{\partial^2 U_2}{\partial S^2} = Ke^{-r(T-t)} \frac{\partial^2 P_2}{\partial S^2}, \]
\[ \frac{\partial U_1}{\partial v} = \frac{\partial P_1}{\partial v}, \quad \frac{\partial U_2}{\partial v} = Ke^{-r(T-t)} \frac{\partial P_2}{\partial v}, \]
\[ \frac{\partial^2 U_1}{\partial v^2} = S \frac{\partial^2 P_1}{\partial v^2}, \quad \frac{\partial^2 U_2}{\partial v^2} = Ke^{-r(T-t)} \frac{\partial^2 P_2}{\partial v^2}, \]
\[ \frac{\partial^2 U_1}{\partial S \partial v} = \frac{\partial^2 P_1}{\partial S \partial v}, \quad \frac{\partial^2 U_2}{\partial S \partial v} = Ke^{-r(T-t)} \left( \frac{\partial P_2}{\partial v} + \frac{\partial P_2}{\partial S} \right), \]

replacing (8) into (6), it follows that \( P_1 \) satisfies the following PDE
\[ \frac{1}{2} \left( \sigma^2 S + \xi^2 v \right) S^2 \left( 2 \frac{\partial P_1}{\partial S} + S \frac{\partial^2 P_1}{\partial S^2} \right) + \xi \sigma_v v S \left( \frac{\partial P_1}{\partial v} + S \frac{\partial^2 P_1}{\partial S \partial v} \right) \]
\[ + \frac{1}{2} \sigma^2 v S \frac{\partial^2 P_1}{\partial v^2} + (r - \lambda j S) S \left( P_1 + S \frac{\partial P_1}{\partial S} \right) + S \frac{\partial P_1}{\partial v} \]
\[ + [k^* (\theta^* - v)] \left( S \frac{\partial P_1}{\partial v} \right) - r S P_1 + \lambda S P_1 \left[ \zeta(S, v) - 1 \right] = 0, \]

and \( P_2 \) satisfies
\[ \frac{1}{2} \left( \sigma^2 S + \xi^2 v \right) S^2 K e^{-r(T-t)} \frac{\partial^2 P_2}{\partial S^2} + \xi \sigma_v v S K e^{-r(T-t)} \frac{\partial^2 P_2}{\partial S \partial v} \]
\[ + \frac{1}{2} \sigma^2 v S K e^{-r(T-t)} \frac{\partial^2 P_2}{\partial v^2} + (r - \lambda j S) S K e^{-r(T-t)} \frac{\partial P_2}{\partial S} \]
\[ + [k^* (\theta^* - v)] K e^{-r(T-t)} \frac{\partial P_2}{\partial v} + K e^{-r(T-t)} \left( r P_2 + \frac{\partial P_2}{\partial t} \right) \]
\[ - K e^{-r(T-t)} r P_2 + K e^{-r(T-t)} \lambda P_2 \left[ \zeta(S, v) - 1 \right] = 0. \]

After some algebra, (9) and (10) become
\[ \frac{1}{2} \left( \sigma^2 S + \xi^2 v \right) S^2 \frac{\partial^2 P_1}{\partial S^2} + \xi \sigma_v v S \frac{\partial^2 P_1}{\partial S \partial v} + \frac{1}{2} \sigma^2 v \frac{\partial^2 P_1}{\partial v^2} \]
\[ + \left[ (r - \lambda j S) + \left( \sigma^2 S + \xi^2 v \right) \right] S \frac{\partial P_1}{\partial S} + \xi \sigma_v v \frac{\partial P_1}{\partial v} + \frac{\partial P_1}{\partial t} \]
\[ - \lambda j S P_1 + \lambda P_1 \left[ \zeta(S, v) - 1 \right] = 0, \]

and
\[ \frac{1}{2} \left( \sigma^2 S + \xi^2 v \right) S^2 \frac{\partial^2 P_2}{\partial S^2} + \xi \sigma_v v S \frac{\partial^2 P_2}{\partial S \partial v} + \frac{1}{2} \sigma^2 v \frac{\partial^2 P_2}{\partial v^2} + (r - \lambda j S) S \frac{\partial P_2}{\partial S} \]
\[ + [k^* (\theta^* - v)] \frac{\partial P_2}{\partial v} + \frac{\partial P_2}{\partial t} + \lambda P_2 \left[ \zeta(S, v) - 1 \right] = 0. \]

Now, consider
\[ \tilde{P}_j(z) := P_j(e^z), \quad j = 1, 2 \]

and note that
\[ \frac{\partial P_j}{\partial S} = \frac{1}{S} \frac{\partial \tilde{P}_j}{\partial Y}, \quad \frac{\partial^2 P_j}{\partial S^2} = \frac{1}{S^2} \left( \frac{\partial^2 \tilde{P}_j}{\partial Y^2} - \frac{\partial \tilde{P}_j}{\partial Y} \right), \quad \frac{\partial^2 P_j}{\partial S \partial v} = \frac{1}{S} \frac{\partial^2 \tilde{P}_j}{\partial Y \partial v}. \]
Replacing (14) into (11) and (12), we obtain the following PDEs

\[
\frac{1}{2} \left( \sigma_S^2 + \xi^2 v \right) \left( \frac{\partial^2 \tilde{P}_j}{\partial Y^2} - \frac{\partial \tilde{P}_j}{\partial Y} \right) + \xi \sigma_v v \frac{\partial^2 \tilde{P}_j}{\partial Y \partial v} + \frac{1}{2} \sigma_v^2 v \frac{\partial^2 \tilde{P}_j}{\partial v^2} + \left[ (r - \lambda_j s) + \left( \sigma_S^2 + \xi^2 v \right) \right] \frac{\partial \tilde{P}_j}{\partial v} + [k^* (\theta^* - v) + \xi \sigma_v v] \frac{\partial \tilde{P}_j}{\partial v} + \lambda \tilde{P}_1 [\zeta(S, v) - 1] = 0, \tag{15}
\]

and

\[
\frac{1}{2} \left( \sigma_S^2 + \xi^2 v \right) \left( \frac{\partial^2 \tilde{P}_j}{\partial Y^2} - \frac{\partial \tilde{P}_j}{\partial Y} \right) + \xi \sigma_v v \frac{\partial^2 \tilde{P}_j}{\partial Y \partial v} + \frac{1}{2} \sigma_v^2 v \frac{\partial^2 \tilde{P}_j}{\partial v^2} + \left[ (r - \lambda_j s) + \frac{1}{2} \left( \sigma_S^2 + \xi^2 v \right) \right] \frac{\partial \tilde{P}_j}{\partial v} + [k^* (\theta^* - v)] \frac{\partial \tilde{P}_j}{\partial v} + \lambda \tilde{P}_2 [\zeta(S, v) - 1] = 0. \tag{16}
\]

After some algebra, (15) and (16) become

\[
\frac{1}{2} \left( \sigma_S^2 + \xi^2 v \right) \frac{\partial^2 \tilde{P}_j}{\partial Y^2} + \xi \sigma_v v \frac{\partial^2 \tilde{P}_j}{\partial Y \partial v} + \frac{1}{2} \sigma_v^2 v \frac{\partial^2 \tilde{P}_j}{\partial v^2} + \left[ (r - \lambda_j s) + \frac{1}{2} \left( \sigma_S^2 + \xi^2 v \right) \right] \frac{\partial \tilde{P}_j}{\partial v} + \lambda \tilde{P}_1 [\zeta(S, v) - 1] = 0, \tag{17}
\]

and

\[
\frac{1}{2} \left( \sigma_S^2 + \xi^2 v \right) \frac{\partial^2 \tilde{P}_j}{\partial Y^2} + \xi \sigma_v v \frac{\partial^2 \tilde{P}_j}{\partial Y \partial v} + \frac{1}{2} \sigma_v^2 v \frac{\partial^2 \tilde{P}_j}{\partial v^2} + \left[ (r - \lambda_j s) - \frac{1}{2} \left( \sigma_S^2 + \xi^2 v \right) \right] \frac{\partial \tilde{P}_j}{\partial v} + \lambda \tilde{P}_2 [\zeta(S, v) - 1] = 0. \tag{18}
\]

The equations (17) and (18) can be written in one unique equation as follows

\[
\frac{1}{2} \left( \sigma_S^2 + \xi^2 v \right) \frac{\partial^2 \tilde{P}_j}{\partial Y^2} + \xi \sigma_v v \frac{\partial^2 \tilde{P}_j}{\partial Y \partial v} + \frac{1}{2} \sigma_v^2 v \frac{\partial^2 \tilde{P}_j}{\partial v^2} + \left[ (r - \lambda_j s) + \frac{1}{2} a_j (\sigma_S^2 + \xi^2 v) \right] \frac{\partial \tilde{P}_j}{\partial v} + \lambda \tilde{P}_j [\zeta(S, v) - 1 - c_j s] = 0, \quad j = 1, 2, \tag{19}
\]

where \(a_1 = 1, \quad a_2 = -1, \quad b_1 = 1, \quad b_2 = 0, \quad c_1 = -1 \quad \text{and} \quad c_2 = 0.

Note that \(\tilde{P}_j, \quad j = 1, 2,\) are the conditional probabilities that the option expires in-the-money, that is

\[
\tilde{P}_j (Y, v; \log K) = \mathbb{Q} \{ Y(T) \geq \log K | Y(t) = Y, v(t) = v \}, \quad j = 1, 2, \tag{20}
\]

where \(Y(t) := \log S(t), \) and \((Y, v)\) evolves according to

\[
dY(t) = \left( (r - \lambda_j s) + \frac{1}{2} a_j (\sigma_S^2 + \xi^2 v(t)) \right) dt + \sigma_S dW_1(t) + \xi \sqrt{v(t)} dW_2(t) + dZ_Y(t), \tag{21}
\]

\[
dv(t) = k^* (\theta^* - v(t)) dt + \xi \sigma_v \sqrt{v(t)} dW_2(t) + dZ_v(t), \quad j = 1, 2. \tag{22}
\]

Using a Fourier transform method one gets

\[
\tilde{P}_j (Y, v; \log K) = \frac{1}{2} + \frac{1}{\pi} \int_0^\infty \Re \left( \frac{e^{-iu_1 \log K} \varphi_j (Y, v; u_1, 0)}{iu_1} \right) du_1, \quad j = 1, 2, \tag{23}
\]

where the characteristic functions \(\varphi_j (Y, v; u_1, u_2)\) also satisfy the PDE (19). The practice to solving this kind of equations is to guess the general form of the solution and set some boundary conditions. Following Heston [Hes93] and Duffie et al. [DPS00], we guess

\[
\varphi_j (Y, v; u_1, u_2) = \exp [C_j (\tau; u_1, u_2) + J_j (\tau; u_1, u_2) + D_j (\tau; u_1, u_2)v + iu_1 Y], \quad j = 1, 2, \tag{24}
\]
where $\tau = T - t$. The explicit expressions of the characteristic functions are obtained as solutions to (19) with terminal condition

$$
\varphi_j(Y,v,T;u_1,u_2) = \exp\{iu_1 Y(T)\}, \quad j = 1, 2. \tag{25}
$$

In particular we have (see Appendix for details)

$$
C_j(\tau;u_1,u_2) = \left[ \left( \frac{1}{2}a_j iu_1 \sigma^2_S - \frac{1}{2}u_1^2 \sigma^2_S + iu_1 \tau \right) \sigma^2_v - k^* \theta^* E_j \right] \frac{e^{\tau}}{\sigma^2_v} + \frac{2k^* \theta^*}{\sigma^2_v} \left[ \sqrt{\Delta_j \tau} - \log \left( \frac{E_j + e^{\sqrt{\Delta_j \tau}}(-B_j + \sqrt{\Delta_j})}{2 \sqrt{\Delta_j}} \right) \right], \quad j = 1, 2, \tag{26}
$$

$$
J_j(\tau;u_1,u_2) = -\lambda \tau(iu_1 js + js c_j - 1) + \frac{1}{2} \lambda \tau \left( \exp \left\{ \left( \log(1 + js) - \frac{1}{2} \delta^2_S \right) iu_1 - \frac{1}{2} \delta^2_S u_1^2 \right\} \right)
+ \frac{\lambda \sigma^2}{2} \sqrt{\Delta_j \left( \frac{1}{\sigma^2_v + js E_j} \frac{E_j - 2 j \sigma^2_v - \sqrt{\Delta_j E_j}}{(E_j - j \sigma^2_v + js E_j)(\sqrt{\Delta_j E_j} - B_j) e^{\sqrt{\Delta_j E_j}}} \right)} , \quad j = 1, 2, \tag{27}
$$

$$
D_j(\tau;u_1,u_2) = \frac{-B_j - \sqrt{\Delta_j}}{\sigma^2_v} + \frac{2 \sqrt{\Delta_j} (B_j + \Delta_j)}{\sigma^2_v (B_j + \sqrt{\Delta_j} + e^{\sqrt{\Delta_j E_j}}(-B_j + \sqrt{\Delta_j E_j}))}, \quad j = 1, 2, \tag{28}
$$

with

$$
B_j = iu_1 \sigma_v \xi - k^* + b_j \xi \sigma_v,
\Delta_j = B_j^2 - \sigma^2_v iu_1 \xi^2 (a_j + iu_1), \quad j = 1, 2
\quad E_j = B_j + \sqrt{\Delta_j}.
$$

Finally, the densities $\tilde{p}_j(Y,v,t;\log K)$ of the distribution functions $\tilde{F}_j(Y,v,t;\log K) = 1 - \tilde{P}_j(Y,v,t;\log K)$ are then

$$
\tilde{p}_j(Y,v,t;\log K) = -\frac{1}{\pi} \int_0^\infty \Re \left( -e^{-iu_1 \log K} \varphi_j(Y,v,t;u_1,0) \right) du_1, \quad j = 1, 2. \tag{29}
$$

### 4 Generating Sample Paths

Financial models usually specify the dynamics of the state variables as stochastic differential equations (SDEs). If these SDEs do not yield closed-form pricing formulae, then numerical approximations (Monte Carlo simulations, among others) can be used. However, the approximation of continuous-time processes by discrete-time processes introduces bias into the simulated solutions and this bias causes several important problems when estimating the prices of derivative securities (Kamrad and Ritchken [KR91]).
Following Broadie and Kaya [BK06], we give a Monte Carlo simulation estimator to compute option price derivatives without discretizing the processes $S$ and $v$. The main idea is that by appropriately conditioning on the paths generated by the variance and jump processes, the evolution of the asset price can be represented as a series of lognormal random variables. The method is called Exact Simulation Algorithm for the Stochastic Volatility with Contemporaneous Jumps (SVCJ) Model.

Let $0 = t_0 < t_1 < \ldots < t_M = T$ be a partition of the interval $[0, T]$ into $M$ possibly unequal segments of length $\Delta t_i = t_i - t_{i-1}$, for each $i = 1, \ldots, M$. We are assuming that we eventually want to price a path-dependent option whose payoff is a function of the asset price vector $(S(t_0), \ldots, S(t_M))$ (note that we can take $M = 1$ for a path-independent option). To illustrate the algorithm, it will be useful to consider the integral form of the dynamics of $S$ and $v$ under $Q$, namely

$$S(t_i) = S(t_{i-1}) \exp \left\{ (r - \lambda j_S - \frac{1}{2} \sigma^2)(t_i - t_{i-1}) - \frac{1}{2} \xi^2 \int_{t_{i-1}}^{t_i} v(q) dq \right\} \exp \left\{ \sigma \int_{t_{i-1}}^{t_i} dW_1(q) + \xi \int_{t_{i-1}}^{t_i} \sqrt{v(q)} \ dW_2(q) + \sum_{k=N(t_{i-1})+1}^{N(t_i)} \log(1 + J_{S}^{(k)}) \right\},$$

and

$$v(t_i) = v(t_{i-1}) + k^* \delta^*(t_i - t_{i-1}) - k^* \int_{t_{i-1}}^{t_i} v(q) dq + \sigma_v \int_{t_{i-1}}^{t_i} \sqrt{v(q)} \ dW_2(q) + \sum_{k=N(t_{i-1})+1}^{N(t_i)} J_{v}^{(k)}.$$

where $N_i$ is the counting process of the (contemporaneous) jumps.

**Remark.** If $N(t_i) - (N(t_{i-1}) + 1) = n_i$, since $\log(1 + J_{S}^{(k)})$ have the normal distribution $\mathcal{N}(\log(1 + j_S) - \frac{1}{2} \delta^2_S, \delta^2_S)$ for every $k$, then

$$\sum_{k=1}^{n_i} \log(1 + J_{S}^{(k)}) \sim \mathcal{N} \left( n_i \left( \log(1 + j_S) - \frac{1}{2} \delta^2_S \right), n_i \delta^2_S \right),$$

that is, $\sum_{k=1}^{n_i} \log(1 + J_{S}^{(k)})$ can be represented as $n_i \left( \log(1 + j_S) - \frac{1}{2} \delta^2_S \right) + \sqrt{n_i} \delta_S R$, where $R$ is a standard normal random variable.

Now, consider two consecutive time steps $t_{i-1}$ and $t_i$ on the time grid and suppose to know $v(t_{i-1})$. The algorithm can be summarized as follows.

**Step 1.** Generate a Poisson random variable with mean $\lambda(t_i - t_{i-1})$ and simulate the number of jumps $n_i$. Determine the time of the next jump after $t_{i-1}$ and denote this time as $\tau_{i,1}$. Set $u := t_{i-1}$ and $t := \tau_{i,1}$ ($u < t$). If $t > t_i$, skip Step 5 and Step 6.

**Step 2.** Generate a sample from the distribution of $v(t)$ given $v(u)$.

7
Step 3. Generate a sample from the distribution of \( \int_u^t v(q) dq \) given \( v(u) \) and \( v(t) \).

Step 4. Recover \( \int_u^t \sqrt{v(q)} dW_2(q) \) from (31) given \( v(u), v(t) \), and \( \int_u^t v(q) dq \).

Step 5. If \( t \leq t_i \), generate \( J_v \) by sampling from an exponential distribution with mean \( j_v \). Update the variance value by setting \( \tilde{v}(t) = v(t) + J_v^{(1)} \), where \( J_v^{(1)} \) is the first jump size of the variance.

Step 6. If \( t < t_i \), determine the time of the next jump \( \tau_{i,2} \) after \( \tau_{i,1} \). If \( \tau_{i,2} \leq t_i \), set \( u := \tau_{i,1} \) and \( t := \tau_{i,2} \). Repeat the iteration Step 2-Step 5 up to \( t_i \).

If \( \tau_{i,2} > t_i \), set \( u := \tau_{i,1} \) and \( t := t_i \). Repeat one time the iteration Step 2-Step 4.

Step 7. Define the average variance between \( t_{i-1} \) and \( t_i \) as
\[
\sigma_i^2 = \frac{n_i \delta_t^2 + \sigma_3^2(t_i - t_{i-1})}{t_i - t_{i-1}},
\]
and an auxiliary variable
\[
\beta_i = \exp \left\{ n_i \log (1 + j_S) - \lambda j_S(t_i - t_{i-1}) - \frac{\xi^2}{2} \int_{t_{i-1}}^{t_i} v(q) dq + \xi \int_{t_{i-1}}^{t_i} \sqrt{v(q)} dW_2(q) \right\}.
\]
Using the definitions (33) and (34), the value \( S(t_i) \) given \( S(t_{i-1}) \) can be written as
\[
S(t_i) = S(t_{i-1}) \beta_i \exp \left\{ \left( r - \frac{\sigma_i^2}{2} \right) (t_i - t_{i-1}) + \sigma_i \sqrt{t_i - t_{i-1}} R \right\},
\]
where \( R \) is a standard normal random variable. From (35), it follows that \( S(t_i) \) is a lognormal random variable.

4.1 Sampling from \( v(t) \) given \( v(u) \)

It is known (see [CIR85]) that the distribution of \( v(t) \) given \( v(u) \) for some \( u < t \) is, up to a scale factor, a non-central chi-squared distribution. Then
\[
v(t) = \frac{\sigma_v^2 (1 - e^{-k^*(t-u)})}{4k^*} \chi_d^2 \left( \frac{4 k^* e^{-k^*(t-u)}}{\sigma_v^2 (1 - e^{-k^*(t-u)})} v(u) \right), \quad u < t
\]
where \( \chi_d^2(\eta) \) denotes the non-central chi-squared random variable with \( d \) degrees of freedom and non-centrality parameter \( \eta \). In particular, in our case
\[
d = \frac{4 \theta^* k^*}{\sigma_v^2}.
\]
For further details, see Section 3.1 of [BK06].
4.2 Sampling from $\int_u^t v(q) dq$ given $v(t)$ and $v(u)$

Once we have a sample for $v(t)$, we want to sample from the distribution of $\int_u^t v(q) dq$ given $v(t)$ and $v(u)$. Broadie and Kaya [BK06] derive the cumulative function of the integral from the conditional characteristic function and then they compute the distribution function numerically. Distinct from this approach, we write the density function from the conditional characteristic function and then we apply the rejection sampling.

The conditional characteristic function is

$$\Phi(a) = \mathbb{E}^Q \left[ \exp \left( ia \int_u^t v(q) dq \right) | v_u, v_t \right]$$

\begin{align*}
&= \frac{\gamma(a)e^{-0.5(\gamma(a)-k^*)(t-u)} \left( 1-e^{-k^*(t-u)} \right)}{k^*(1-e^{-\gamma(a)(t-u)})} \\
&\quad \exp \left\{ \frac{v_u + v_t}{\sigma^2} \left[ \frac{k^*(1+e^{-k^*(t-u)})}{1-e^{-k^*(t-u)}} - \frac{\gamma(a)(1+e^{-\gamma(a)(t-u)})}{1-e^{-\gamma(a)(t-u)}} \right] \right\}
\end{align*}

where $\gamma(a) = \sqrt{k^*+4\sigma^2ia}$, $d$ is given in (37), and $I_\nu$ is the modified Bessel function of the first kind (for details, see Section 3.2 of [BK06]).

The density function of integral $\int_u^t v(q) dq$ given $v(u)$ and $v(t)$ is so given by

$$f(x) = \frac{2}{\pi} \int_0^\infty \cos(ax) \Re[\Phi(a)] da,$$

where $\Re(z)$ is the real part of $z \in \mathbb{C}$.

The rejection sampling generates sampling values from the density function (39) by using an instrumental density function $g(x)$ such that $f(x) < cg(x)$ for some constant $c > 1$. Specifically:

Step 1. Sample $x$ from $g(x)$ and $u$ from $U \sim U ((0,1))$;

Step 2. Check whether $u < \frac{f(x)}{cg(x)}$;

2a. If this holds, accept $x$ as a realization of $f(x)$;

2b. If this does not hold, reject $x$ and repeat the sampling procedure.

5 Conditional Monte Carlo for SVCJ Model

To improve the efficiency of the exact simulation algorithm, we use a conditional Monte Carlo method. This method is applicable to path-dependent derivatives that have closed-form solutions under the Black-Scholes method (see Williard [Wil97]).

Let $C(S(0), k, r, T, \sigma)$ denote the Black-Scholes formula for a European call option, with maturity
The option price is given by
\[ E\left[ e^{-rT}(S(T) - K)^+ \right] = E\left[ e^{-rT}(S(T) - K)^+ | W_2, J_S \right] = E[C(S(0)\beta_M, K, r, T, \sigma_M)], \]
where \( \sigma_M \) is defined in (33), and \( \beta_M \) in (34). Note that \( \beta_M \) is constant conditional on paths of \( W_2 \) and \( J_S \).

6 Path-dependent Options

The method introduced in Section 4 and Section 5 can be used for the evaluation of more complex options, which do not have a closed-form solutions. In particular, we consider barrier options and American-style options.

6.1 Barrier options

Barrier options are derivatives securities with the defining characteristic that the payoff may be zero, depending on whether or not an underlying variable crosses a specified barrier during the life of the option. There are two broad types of barrier options: “knock-out” options, which pay either zero or a rebate when there is a barrier crossing, and “knock-in” options, which pay either zero or a rebate unless there is a barrier crossing. The payoff of barrier options is discontinuous over the space of all paths of the underlying variables and, only in simple cases, there are analytical formulae for their price. Consequently, it is often necessary to price them via simulation: a straightforward simulation proceeds by dividing the lifetime of the option into several time steps. Each path begins with the state vector at a specified initial value, and uses an approximate, discretized version of the dynamics to propagate this random vector forward at each time step. A simulation for a knock-out option has the special feature that if at any time the underlying process crosses the barrier, the path is abandoned, because it is already known that it results in a zero payoff. If a path survives by never crossing the barrier, then its payoff is determined at the terminal value of the state vector. This standard Monte Carlo approach to pricing knock-out options suffers from a peculiar defect which creates a possibility for improving the method: some simulated paths survive, allowing for a positive payoff, while other paths fail to survive and have zero payoff. The lower the probability of survival, the more simulated payoffs are zero. This can make the average payoff among the surviving paths, and hence the variance among all paths, quite large relative to the price.

All the variance due to the possibility of knock-out could be removed by different methods (see Boyle et al. [BBG97] for a review of variance reduction techniques). We chose the conditional Monte Carlo technique introduced in Section 5 to improve the efficiency of the simulation.
method described in Section 4. This method exploits the following variance reducing property of conditional expectation: for any random variables $X$ and $Y$, $\text{var}[E[X|Y]] \leq \text{var}[X]$, with strictly inequality expect in trivial cases. A necessary condition to apply this method is that the derivatives have closed-form solutions under Black and Scholes model.

Now, we illustrate the conditional Monte Carlo method for discrete barrier options. The discounted payoff for a discrete knock-out barrier option with strike $K$, expiration $T$ and barrier $H > S(0)$ is given by

$$e^{-rT}(S(T) - K)^+ 1_{\{\max_{1 \leq i \leq M} S(t_i) < H\}}, \quad (41)$$

where $S(t_i)$ is the asset price at time $t_i$ for a time partition $0 = t_0 < t_1 < \ldots < t_M = T$. Let $C(S(0), K, r, T, \sigma)$ denote the Black and Scholes price of an European call option with constant volatility $\sigma$, maturity $T$, strike $K$, and written on an asset with initial price $S(0)$. As in Section 5, we use the law of iterated expectations, and obtain the following unconditional price of the option

$$\mathbb{E}\left[e^{-rT}(S(T) - K)^+ 1_{\{\max_{1 \leq i \leq M} S(t_i) < H\}}\right] =$$

$$\mathbb{E}\left[\mathbb{E}\left[e^{-rT}(S(T) - K)^+ 1_{\{\max_{1 \leq i \leq M} S(t_i) < H\}} | W_2, J_S\right]\right] =$$

$$\mathbb{E}\left[C(S(0)\beta_M, K, r, T, \sigma_M) 1_{\{\max_{1 \leq i \leq M} S(t_i) < H\}}\right]. \quad (42)$$

This approach can also be used to generate an unbiased estimator of Greeks.

### 6.1.1 Greeks of Discrete Barrier Options

In this section we illustrate the likelihood ratio (LR) method to generate an unbiased estimate of some Greeks of discrete barrier options, namely delta, gamma and rho.

We first review how the LR estimator is derived in a usual simulation setting. Suppose that $p \in \mathbb{R}^n$ is a vector of parameters with probability density $g_p(X)$, where $X$ is a random vector that determines the payoff function $f(X)$. The option price is given by

$$\alpha(p) = \mathbb{E}^Q[f(X)], \quad (43)$$

and we are interested in finding the derivative $\alpha'(p)$. From (43), one gets

$$\alpha'(p) = \frac{d}{dp} \mathbb{E}^Q[f(X)] = \int_{R^n} f(x) \frac{d}{dp} g_p(x) dx = \int_{R^n} f(x) \frac{g'_p(x)}{g_p(x)} g_p(x) dx = \mathbb{E}^Q \left[f(X) \frac{g'_p(x)}{g_p(x)}\right]. \quad (44)$$

The expression $f(X) \frac{g'_p(x)}{g_p(x)}$ is an unbiased estimator of $\alpha'(p)$ and the quantity $\frac{g'_p(x)}{g_p(x)}$ is called the score function. Note that this latter does not depend on the form of the payoff function and each option Greek is computed according to which quantity is considered as parameter in the expression of $g$.

As in the previous section, we consider a discrete knock-out barrier option with strike $K$, expiration $T$ and barrier $H > S(0)$, whose payoff is given by

$$e^{-rT}(S(T) - K)^+ 1_{\{\max_{1 \leq i \leq M} S(t_i) < H\}},$$
where $S(t_i)$ is the asset price at time $t_i$ for a time partition $0 = t_0 < t_1 < \ldots < t_M = T$. From (44), it follows that the LR estimators for the option Greeks are given by the product of the discounted payoff $f(X)$ and the score function. In our model, the first one is given by (41) and the score function is determined by using the key idea of Conditional Monte Carlo method (see Section 5): by appropriately conditioning on the paths generated by the variance and jump processes, the evolution of the asset price $S$ is a lognormal random variable (see (35)), hence its conditional density is

$$g(x) = \frac{1}{x\sigma_i \sqrt{\Delta t_i}} \phi(d_i(x)), \quad (45)$$

where $\sigma_i$ is defined in (33), $\Delta t_i := t_i - t_{i-1}$, $\phi(\cdot)$ is the standard normal density function and

$$d_i(x) = \log \left( \frac{x}{S(t_{i-1})\sigma_i} \right) - (r - \frac{1}{2}\sigma_i^2)\Delta t_i. \quad (46)$$

Now, to find the estimator of delta and gamma, i.e. the first and the second derivative with respect to the initial price of the underlying asset price, respectively, we take $p = S$ in (44) and we compute the derivative of $g$ with respect to $S(0)$. After some algebra, we have

$$\frac{\partial g(x)}{\partial S(0)} = \frac{d_i(x) \phi(d_i(x))}{xS(0)\sigma_i^2 \Delta t_i}. \quad (47)$$

Dividing this latter by $g(x)$ and evaluating the expression at $x = S(t_1)$, we have the following score function for LR delta estimator

$$\frac{d_1}{S(0)\sigma_1 \sqrt{\Delta t_1}}. \quad (48)$$

Analogously for LR gamma estimator. So we have

**Delta:**

$$e^{-rT}(S(T) - K)^+ I\{\max_{1 \leq i \leq M} S(t_i) < H\} \left( \frac{d_1}{S(0)\sigma_1 \sqrt{\Delta t_1}} \right), \quad (49)$$

**Gamma:**

$$e^{-rT}(S(T) - K)^+ I\{\max_{1 \leq i \leq M} S(t_i) < H\} \left( \frac{d_1^2 - d_1 \sigma_1 \sqrt{\Delta t_1} - 1}{S^2(0) \sigma_1^2 \Delta t_1} \right), \quad (50)$$

where $d_i$ is defined in (46), $\sigma_i$ in (33) and $\Delta t_i := t_i - t_{i-1}$.

To compute the estimator of rho, it is sufficient to compute the derivative of $g$ with respect to $r$.

**Rho:**

$$e^{-rT}(S(T) - K)^+ I\{\max_{1 \leq i \leq M} S(t_i) < H\} \left( -T + \sum_{i=1}^{M} \frac{d_i \sqrt{\Delta t_i}}{\sigma_i} \right), \quad (51)$$

where $d_i, \sigma_i$ and $\Delta t_i$ are as above.
6.2 American-style options

Whereas a European option can be exercised only at maturity, an American option can be exercised any time up to its expiration date. The value of American option is the value achieved by exercising it optimally. Finding this value entails finding the optimal exercise rule and computing the expected discounted payoff of the option under this rule. The embedded optimization problem makes this a difficult problem for simulation.

Specifically, the American option pricing problem is to find

\[ C = \max_{\tau} \mathbb{E} \left[ e^{-r\tau} (S(\tau) - K)^+ \right], \]  

over all stopping times \( \tau \leq T \), where \( \mathbb{Q} \) is the risk-neutral measure, \( r \) denotes the riskless rate of interest, \( T \) is the option maturity, \( K \) and \( S \) are the strike and the stock price, respectively.

Most methods for computing American option prices rely on a dynamical programming representation of the problem, but it is also convenient to view it through stopping rules or, equivalently, exercise regions. This reduces the optimal stopping problem to a much more tractable finite-dimensional optimization problem.

A possible approach is the random tree method of Broadie and Glasserman [BG97]. Through this method, it is possible to solve the full optimal stopping problem and estimate the value of an American option by generating two estimates of the asset price based on random samples of future state trajectories, via Monte Carlo simulation, and increasingly refined approximations to optimal exercise decisions. One estimate is biased high and one is biased low; both estimates are asymptotically unbiased and converge to the true price. The two estimates are combined to give a valid, conservative confidence interval for the asset price.

For more on the use of Monte Carlo simulation for derivatives pricing see Boyle et al. [BBG97] and for a review of all numerical methods see Broadie and Detemple [BD96].

**Random Tree Method.**

The afore-mentioned estimators are based on simulated trees. The simulated trees are parameterized by \( b \), the number of branches per node. It is important to keep in mind that the nodes at a fixed time appear according to the order in which they are generated, not according to their node values, as would be the case in a lattice method.

Let \( C \) denote the price of an American call option with \( M + 1 \) exercise opportunities at times \( t_i, \ i = 0, \ldots, M \). We denote the high estimator by \( \Theta \). It is defined as the call value estimate obtained by a dynamic programming (DP) algorithm applied to the simulated tree. At the terminal date, the option value is known. At each prior date, the option value is defined to be the maximum of the immediate exercise value and the expectation of the succeeding discounted option values. Finally, \( \Theta \) is the estimated option value at the initial node. The \( \Theta \) estimator gives
an estimate of the true option price which is biased upward, that is \( \mathbb{E}^Q[\Theta] \geq C \), but is consistent and converges to \( C \) as \( b \) increases (about the type of convergence, see Section 3, Broadie and Glasserman [BBG97]).

The low estimator is obtained by separating the branches at each node into two sets. The first set of branches is used to decide whether or not to exercise, and the second set is used to estimate the continuation value, if necessary. Also this estimator is biased, in particular it is biased downward, that is \( \mathbb{E}^Q[\Theta] \leq C \) and converges to \( C \).

### 7 Numerical results

In this section, to compare the efficiency of the exact algorithm described in Section 4, we give some numerical results about the prices obtained with the closed formula (29) and those obtained with the simulation method. To do this, we apply our model to the DJ Euro Stoxx 50 market\(^1\) using the set of parameters reported in Table 1.

The Dow Jones Euro Stoxx 50 (DJ50) ‘blue-chip’ index covers the fifty EuroZone largest sector leaders whose stocks belong to the Dow Jones Euro Stoxx Index. DJ50’s option market is very liquid and ranges widely in both maturities and strike prices. The shortest maturity is one month being ten years the longest with ten overall maturities. Strike prices cover moneyness from 85% up to 115%. It is worth reminding that indexes carry dividends paid by companies so that data present a dividend yield. This feature has been properly considered in all above valuation formulae by subtracting the dividend yield \( d \) from the drift term of the dynamics of the asset price.

![Table 1: Values of parameters of the model (4)-(5).](https://example.com/table1.png)

<table>
<thead>
<tr>
<th>( \theta^* )</th>
<th>( \kappa^* )</th>
<th>( \sigma_S )</th>
<th>( \sigma_v )</th>
<th>( \xi )</th>
<th>( \lambda )</th>
<th>( j_S )</th>
<th>( \delta_S )</th>
<th>( j_v )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.175</td>
<td>0.25</td>
<td>0.08</td>
<td>0.2</td>
<td>-0.4</td>
<td>0.05</td>
<td>0.025</td>
<td>0.02</td>
<td>0.03</td>
</tr>
</tbody>
</table>

Volatilities in Table 2 represent the term \( \sqrt{\sigma_S^2 + \xi v(0)} \) (the instantaneous variance of spot return at \( t = 0 \), and not simply \( \sigma_S \) as in Black and Scholes model), where \( v(0) \) is the initial value of the stochastic volatility dynamics. It follows that we can obtain \( v(0) \) from

\[
v(0) = \frac{\sigma^2_{MKT} - \sigma^2_S}{\xi^2},
\]

\(^1\)Data provided by Banca IMI, Milano.
where $\sigma_{MKT}$ is the market volatility.

Table 2 gives simulation results for European-style options using different numbers of paths for the variance and for the price. In particular we simulate 100 000 variance paths and 1 000 price paths conditional on each variance path and jumps. The results refer to call options on November 23, 2006 with spot price $S(0) = 4116.40$, time to maturity 1 year, riskless rate $\bar{r} = 3.78\%$, dividend yield $d = 3.37\%$ and volatilities in Table 2. The time needed for prices obtained with exact algorithm is 548.807 seconds with an AMD Athlon MP 2800+, 2.25 GHz processor. The codes were written in FORTRAN programming language. As one can see, the prices obtained with the exact method are very similar to those obtained with the closed formula (29). This is an encouraging result for pricing options that do not have closed-form formulae such as barrier and American options.

<table>
<thead>
<tr>
<th>moneyness</th>
<th>strike $K$</th>
<th>market volatility $\sigma_{MKT}$</th>
<th>closed formula price</th>
<th>exact algorithm price</th>
</tr>
</thead>
<tbody>
<tr>
<td>85.00%</td>
<td>3498.94</td>
<td>0.1890</td>
<td>711.11304</td>
<td>711.974487</td>
</tr>
<tr>
<td>90.00%</td>
<td>3704.76</td>
<td>0.1780</td>
<td>545.30995</td>
<td>545.248901</td>
</tr>
<tr>
<td>95.00%</td>
<td>3910.58</td>
<td>0.1660</td>
<td>393.24333</td>
<td>393.767395</td>
</tr>
<tr>
<td>97.50%</td>
<td>4013.49</td>
<td>0.1600</td>
<td>324.19082</td>
<td>324.333435</td>
</tr>
<tr>
<td>100.00%</td>
<td>4116.40</td>
<td>0.1550</td>
<td>262.12052</td>
<td>262.717041</td>
</tr>
<tr>
<td>102.50%</td>
<td>4219.31</td>
<td>0.1500</td>
<td>206.35584</td>
<td>206.269440</td>
</tr>
<tr>
<td>105.00%</td>
<td>4322.22</td>
<td>0.1450</td>
<td>157.47292</td>
<td>157.688583</td>
</tr>
<tr>
<td>110.00%</td>
<td>4528.04</td>
<td>0.1380</td>
<td>85.18119</td>
<td>85.346504</td>
</tr>
<tr>
<td>115.00%</td>
<td>4733.86</td>
<td>0.1320</td>
<td>40.68553</td>
<td>40.149242</td>
</tr>
</tbody>
</table>

Table 2: *Comparison among prices of European options with spot price $S(0) = 4116.40$, time to maturity $T - t = 1$ (year), riskless rate $\bar{r} = 3.78\%$ and dividend yield $d = 3.37\%$ on November 23, 2006.*
A Appendix

This appendix derives the characteristic function of the probabilities $\tilde{P}_j$ given in (13).

Following Rockinger and Semenova [RS05], we define the characteristic functions of the process $(Y, v)$, which evolves according to (21) and (22), as

$$\varphi_j(Y, v, t; u_1, u_2) = \mathbb{E}^Q[\exp iu_1Y(T) + iu_2v(T)|Y(t) = Y, v(t) = v].$$

By applying the Itô's lemma, we have

$$\frac{\partial \varphi_j}{\partial t} + \frac{\partial \varphi_j}{\partial Y} [r - \lambda jS] + \frac{1}{2} a_j(\sigma_S^2 + \xi^2 v) + \frac{\partial \varphi_j}{\partial v} [k^* (\theta^* - v) + b_j \xi v]$$

$$+ \frac{1}{2} \frac{\partial^2 \varphi_j}{\partial Y^2} (\sigma^2 + \xi^2 v) + \frac{\partial^2 \varphi_j}{\partial Y \partial v} \sigma v + \frac{1}{2} \frac{\partial^2 \varphi_j}{\partial v^2} \sigma_v^2 v$$

$$+ \lambda \mathbb{E}^Q[J_{\varphi_j}(Y + J_Y, v + J_v) - \varphi_j(Y, v)] = 0.$$  

To go further, we have to reformulate the expectation term, so we introduce the following jump transform

$$\zeta(c_1, c_2) = \mathbb{E}^Q[\exp \{c_1 J_Y + c_2 J_v\} |Y(t) = Y, v(t) = v], \quad (c_1, c_2) \in \mathbb{R}^2,$$

where $Q$ is the risk-neutral measure and $J_Y$, $J_v$ are the jump sizes of $Y$ and $v$, respectively. The expectation term is so given by

$$\varphi_j(Y, v, t; u_1, u_2)[\zeta(iu_1, D_j) - 1],$$

where the functions $D_j = D_j(\tau; u_1, u_2)$ are specified later.

Now, we guess that $\varphi_j$ is exponential-affine

$$\varphi_j(Y, v, t; u_1, u_2) := \exp [C_j(\tau; u_1, u_2) + J_j(\tau; u_1, u_2) + D(\tau; u_1, u_2) v + iu_1 Y],$$

where $\tau = T - t$. We want to write (55) with this latter form of $\varphi_j$, so we need the following derivatives

$$\frac{\partial \varphi_j}{\partial \tau} = \left( - \frac{\partial C_j}{\partial \tau} - \frac{\partial J_j}{\partial \tau} - \frac{\partial D_j}{\partial \tau} v \right) \varphi_j,$$

$$\frac{\partial \varphi_j}{\partial Y} = iu_1 \varphi_j,$$

$$\frac{\partial \varphi_j}{\partial v} = D_j \varphi_j,$$

$$\frac{\partial^2 \varphi_j}{\partial Y^2} = -u_1^2 \varphi_j,$$

$$\frac{\partial^2 \varphi_j}{\partial Y \partial v} = iu_1 D_j \varphi_j,$$

$$\frac{\partial^2 \varphi_j}{\partial v^2} = D_j^2 \varphi_j.$$  

The PDE (55) becomes

$$- \frac{\partial C_j}{\partial \tau} - \frac{\partial J_j}{\partial \tau} - \frac{\partial D_j}{\partial \tau} v + iu_1 \left[ (r - \lambda jS) + \frac{1}{2} a_j (\sigma_S^2 + \xi^2 v) \right]$$

$$+ D_j [k^* (\theta^* - v) + b_j \xi v] - \frac{1}{2} u_1^2 (\sigma_S^2 + \xi^2 v) + iu_1 D_j \sigma_v \xi v + \frac{1}{2} D_j^2 \sigma_v^2 v$$

$$+ \lambda \left[ \zeta(iu_1, D_j) - 1 + j_S c_j \right] = 0.$$  

Now, the terms in $v$, the ones related to the diffusion part $D$ and the ones related to jumps are grouped together to obtain the following ordinary differential equations (ODEs)

$$- \frac{\partial C_j}{\partial \tau} + iu_1 r + \frac{1}{2} a_j iu_1 \sigma_S^2 + D_j k^* \theta^* - \frac{1}{2} u_1^2 \sigma_S^2 = 0,$$

$$\frac{\partial J_j}{\partial \tau} + iu_1 \theta^* + \frac{1}{2} \sigma_v^2 = 0,$$

$$\frac{\partial D_j}{\partial \tau} = D_j \left( - r - \lambda jS + \frac{1}{2} a_j (\sigma_S^2 + \xi^2 v) \right) + \frac{1}{2} \sigma_v^2.$$  

$$16$$
It is well known that a solution to (61) is of the form
\[ \frac{-\partial D_j}{\partial \tau} + \frac{1}{2} a_j \xi^2 i u_1 - D_j k^* + b_j \xi \sigma_v D_j - \frac{1}{2} u_1^2 \xi^2 + i u_1 D_j \sigma_v \xi + \frac{1}{2} D_j \sigma_v^2 = 0, \]
and
\[ -\frac{\partial J_j}{\partial \tau} - i u_1 \lambda j s + \lambda [\xi (i u_1, D_j) - 1 + j s c_j] = 0. \]
boundary conditions are \( D_j(0; u_1, u_2) = C_j(0; u_1, u_2) = J_j(0; u_1, u_2) = 0 \) so that (25) is satisfied.
It is well known that a solution to (61) is of the form \( D_j = \overline{D}_j + \frac{1}{2} \overline{z} \), where \( \overline{D}_j \) satisfies
\[ \frac{1}{2} \sigma_v^2 \overline{D}_j + (i u_1 \sigma_v \xi - k^* + b_j \xi \sigma_v) \overline{D}_j + \frac{1}{2} i u_1 \xi^2 (i u_1 + a_j) = 0, \]
and
\[ \overline{D}_j = \frac{-B_j - \sqrt{\Delta_j}}{\sigma_v^2}, \]
where
\[ B_j = i u_1 \sigma_v \xi - k^* + b_j \xi \sigma_v, \]
\[ \Delta_j = B_j^2 - \sigma_v^2 \xi^2 (a_j + i u_1), \]
and \( z \) satisfies
\[ \frac{d z}{d \tau} - \sqrt{\Delta_j} z = -\frac{1}{2} \sigma_v^2, \]
namely,
\[ z(\tau) = \frac{\sigma_v^2}{2 \sqrt{\Delta_j}} + c e^{\sqrt{\Delta_j} \tau}, \]
where \( c \) is a constant. The terminal condition \( D_j(0; u_1, u_2) = 0 \) gives us the value of the constant, that is \( c = \frac{\sigma_v^2 (\sqrt{\Delta_j} - B_j)}{2 \sqrt{\Delta_j} (\sqrt{\Delta_j} + B_j)} \).
The functions \( D_j \) are so given by (28). Replacing \( D_j \) in (60) we have
\[ \frac{-\partial C_j}{\partial \tau} + i u_1 r + \frac{1}{2} a_j i u_1 \xi S - \frac{1}{2} u_1^2 \xi^2 \]
\[ + k^* \theta^* \left( \frac{-B_j - \sqrt{\Delta_j}}{\sigma_v^2} + \frac{2 \sqrt{\Delta_j} (B_j + \Delta_j)}{\sigma_v^2 (B_j + \sqrt{\Delta_j} + e^{2 \sqrt{\Delta_j}} (-B_j + \sqrt{\Delta_j}))} \right) = 0. \]
It follows that
\[ C_j(\tau; u_1, u_2) = \left( i u_1 r + \frac{1}{2} a_j i u_1 \xi S - \frac{1}{2} u_1^2 \xi^2 - k^* \theta^* \frac{B_j + \sqrt{\Delta_j}}{\sigma_v^2} \right) \tau \]
\[ + \int_0^\tau \frac{2 \sqrt{\Delta_j} (B_j + \Delta_j)}{\sigma_v^2 (B_j + \sqrt{\Delta_j} + e^{2 \sqrt{\Delta_j}} (-B_j + \sqrt{\Delta_j}))} dq, \]
and solving the integral part, the functions \( C_j \) are given by (26). At last, replacing (28) in (62), one gets
\[ -\frac{\partial J_j}{\partial \tau} = -i u_1 \lambda j s + \lambda [\xi (i u_1, D_j) - 1 + j s c_j], \]
or, equivalently,
\[ J_j(\tau; u_1, u_2) = -\lambda \tau (i u_1 j s + 1 - j s c_j) \]
\[ + \frac{1}{2} \lambda \int_0^\tau \left( \exp \{ (log(1 + j s) - \frac{1}{2} \delta^2) i u_1 - \frac{1}{2} \delta^2 u_1^2 \} + \frac{1}{1 - j s, D_j(u_1, u_2)} \right) d q. \]
Using the explicit form of $D_j$ and computing the latter integral, we obtain (27). Note that the terminal conditions $C_j(0; u_1, u_2) = J_j(0; u_1, u_2) = 0$ are satisfied.
References


