A hybrid tree/finite difference approach for the Bates model

Maya Briani*
Lucia Caramellino†
Antonino Zanette‡

Abstract

We propose a hybrid approximation of functionals of the Bates jump model with stochastic interest rate that uses a tree method in the direction of the volatility and the interest rate, and a finite difference approach in order to handle the underlying asset price process. We also propose a hybrid Monte Carlo method to approximate the model, which uses a binomial tree in the direction of the volatility and the interest rate, and a space-continuous approximation for the underlying asset price process coming from a Euler-Maruyama type scheme. We show that our methods allow to obtain efficient and accurate European and American option prices. Numerical results are provided, showing the reliability and the efficiency of the algorithms.

Keywords: Bates model; Heston-Hull-White model; jump-diffusion process; stochastic volatility; American options; tree methods; finite difference.
2000 MSC: 91G10, 60H30, 65C20.

Premia 22

1 Introduction

The Black-Scholes [5] model assumes that the option underlying asset follows a geometric Brownian motion with drift and diffusion parameters with constant volatility. In practice, all option markets exhibit a volatility smile behaviour. In order to take into account this phenomenon in the foreign exchange option market Bates [3] introduces a stochastic-volatility with price jumps model where the dynamics of the underlying are driven by both stochastic volatility following the square root process of Heston [18], and by a Poisson jump process of the type originally introduced by Merton [24]. In the case of plain vanilla European options Fourier inversion methods [11] leads to closed-form formulas for option pricing under jump-diffusions with stochastic volatility. Nevertheless, in the American case the numerical literature is very poor. The finite difference methods for solving the parabolic integro partial differential equation associated to the option pricing problems can be based on implicit, explicit or alternating direction implicit schemes. The implicit scheme requires to solve a dense sparse system at each time step. Tovainen [30] propose a componentwise splitting method for pricing American

*Istituto per le Applicazioni del Calcolo, CNR Roma - m.briani@iac.cnr.it
†Dipartimento di Matematica, Università di Roma Tor Vergata - caramell@mat.uniromP2.it
‡Dipartimento di Scienze Economiche e Statistiche, Università di Udine - antonino.zanette@uniud.it

options in the Bates model. The linear complementarity problem associated to the American option problem is decomposed into a sequence of five one-dimensional LCP's problems at each time step. The advantage is that LCP's need the use of tridiagonal matrices. More recently Chiarella at al [12] develop a method of lines algorithm for pricing and hedging American option prices under stochastic volatility and jump-diffusion dynamics.
In this paper we propose the approach based both on tree and finite difference methods introduced in Briani, Caramellino and Zanette [6].

The paper is organized as follows. In Section 2, we introduce the Bates model with stochastic interest rate. In Section 3 we describe the tree procedure for the volatility and the interest rate pair (V, X). Then, in Section 4 we switch to the numerical approximation of the log-price process $Y = \log S$ and we describe both the Monte Carlo approach (Section 4.1) and the partial differential one (Section 4.2). To outline the latter procedure, we show in section 5 how to use our hybrid tree/finite difference scheme for the pricing of American options. Moreover, in Section 6 we give a schematic sketch of the major computational steps in order to use the two proposed approximations. Numerical results and comparisons are in Section 7.

2 The Bates model with stochastic interest rate

In this section we consider the case of Bates model associated with the Vasicek model for the stochastic interest rate. In the Bates model, as introduced in [3], the volatility is assumed to follow the Cox-Ingersoll-Ross (hereafter CIR) process and the underlying asset price process contains a noise from a jump process of the type originally introduced by Merton [24]. Moreover, we allow here the interest rate to follow a stochastic model and we assume to be described by a generalized Ornstein-Uhlenbeck (hereafter OU) process. More precisely, the dynamics under the risk neutral measure of the share price S, the volatility process V and the interest rate r, are given by the following jump-diffusion model:

$$\begin{aligned}
\frac{dS_t}{S_{t^-}} &= (r_t - \eta)dt + \sqrt{V_t}\, dZ_t^S + dH_t, \\
dV_t &= \kappa_V(\theta_V - V_t)dt + \sigma_V \sqrt{V_t}\, dZ_t^V, \\
dr_t &= \kappa_r(\theta_r(t) - r_t)dt + \sigma_r dZ_t^r,
\end{aligned}$$

with $S_0 > 0$, $V_0 > 0$ and $r_0 > 0$, Z^S, Z^V and Z^r are suitable and possibly correlated Brownian motions and H_t is a compound Poisson process with intensity λ and i.i.d. jumps $\{J_k\}_k$, that is

$$H_t = \sum_{k=1}^{K_t} J_k,$$

where K denotes a Poisson process with intensity λ. We assume that the Poisson process K, the jump amplitudes $\{J_k\}_k$ and the Brownian motions Z^S, Z^V and Z^r are independent. As suggested by Grzelak and Oosterlee in [17], the significant correlations are between the pairs (S, V) and (S, r). So, as done in [7], we assume that the couple (Z^V, Z^r) is a standard Brownian motion in $\mathbb{R}^2$ and Z^S is a Brownian motion in $\mathbb{R}$ which is correlated both with Z^V and Z^r:

$$d\langle Z^S, Z^V\rangle_t = \rho_1 dt \quad \text{and} \quad d\langle Z^S, Z^r\rangle_t = \rho_2 dt.$$

We recall that the volatility process V follows a CIR dynamics with mean reversion rate κ_V, long run variance θ_V and σ_V denotes the vol-vol (volatility of the volatility). We stress that we never require in this paper that the CIR process satisfies the Feller condition $2\kappa_V\theta_V \geq \sigma_V^2$, ensuring that the process V

never hits 0. So, we allow the volatility V to reach 0. The interest rate r_t is described by a generalized OU process, in particular θ_r is deterministic and fits the zero-coupon bond market values, for details see [10]. As already done in [19], we write the process r as follows:

$$r_t = \sigma_r X_t + \varphi_t \tag{2.1}$$

where

$$X_t = -\kappa_r \int_0^t X_s\, ds + Z_t^r \quad \text{and} \quad \varphi_t = r_0 e^{-\kappa_r t} + \kappa_r \int_0^t \theta_r(s) e^{-\kappa_r (t-s)} ds. \tag{2.2}$$

We recall that the process S can be written as follows. Let τ_k, $k \geq 1$, denote the jump times of the compound Poisson process H, and let us add $\tau_0 = 0$. Then for $k \geq 0$ and $t \in [\tau_k, \tau_{k+1})$ one has

$$S_t = S_{\tau_k} \exp\Big(\int_{\tau_k}^t \Big(r_s - \eta - \frac{1}{2} V_s \Big) ds + \int_{\tau_k}^t \sqrt{V_s} dZ_s \Big)$$

and at the jump time τ_{k+1},

$$S_{\tau_{k+1}} = S_{\tau_{k+1}^-} + S_{\tau_{k+1}^-} J_{k+1} = (1 + J_{k+1}) S_{\tau_k} \exp\Big(\int_{\tau_k}^{\tau_{k+1}} \Big(r_s - \eta - \frac{1}{2} V_s \Big) ds + \int_{\tau_k}^{\tau_{k+1}} \sqrt{V_s} dZ_s \Big).$$

From now on we set

$$Z^V = W_1, \quad Z^r = W_2, \quad Z^S = \rho_1 W_1 + \rho_2 W_2 + \rho_3 W_3,$$

where $W = (W_1, W_2, W_3)$ is a standard Brownian motion in $\mathbb{R}^3$ and the correlation parameter ρ_3 is given by

$$\rho_3 = \sqrt{1 - \rho_1^2 - \rho_2^2}, \quad \rho_1^2 + \rho_2^2 \leq 1.$$

By passing to the logarithm $Y = \ln S$ in the first component, taking into account the above mentioned correlations and (2.1)-(2.2), we reduce to the triple (Y, V, X) given by

$$\begin{aligned} dY_t &= \mu_Y(V_t, X_t, t)dt + \sqrt{V_t}\,(\rho_1 dW_t^1 + \rho_2 dW_t^2 + \rho_3 dW_t^3) + dN_t, \quad Y_0 = \ln S_0 \in \mathbb{R}, \\ dV_t &= \mu_V(V_t)dt + \sigma_V \sqrt{V_t}\, dW_t^1, \quad V_0 > 0, \\ dX_t &= \mu_X(X_t)dt + dW_t^2, \quad X_0 = 0, \end{aligned} \tag{2.3}$$

where

$$\mu_Y(v, x, t) = \sigma_r x + \varphi_t - \eta - \frac{1}{2} v, \tag{2.4}$$

$$\mu_V(v) = \kappa_V(\theta_V - v), \tag{2.5}$$

$$\mu_X(x) = -\kappa_r x, \tag{2.6}$$

and N_t is the compound Poisson process written through the Poisson process K and the i.i.d. jumps $\{\log(1 + J_k)\}_k$, that is

$$N_t = \sum_{k=1}^{K_t} \log(1 + J_k).$$

Recall that the Poisson process K has intensity λ and is such that K, the jump amplitudes $\{\log(1 + J_k)\}_k$ and the Brownian motions W_1, W_2 and W_3 are all independent. We also recall that the Lévy measure linked to the compound Poisson process N is given by

$$\nu(dx) = \lambda \mathbb{P}(\log(1 + J_1) \in dx),$$

and whenever $\log(1+J_1)$ is absolutely continuous then ν has a density as well:

$$\nu(dx) = \nu(x)dx = \lambda p_{\log(1+J_1)}(x)dx, \tag{2.7}$$

$p_{\log(1+J_1)}$ denoting the probability density function of $\log(1+J_1)$. For example, in the Merton model [24] it is assumed that $\log(1+J_1)$ has a normal distribution, that is

$$\log(1+J_1) \sim N(\mu, \delta^2).$$

This is the choice we will do in our numerical experiments and, for practical purposes, we shall take $\mu = \gamma - \frac{1}{2}\delta^2$ for a given suitable $\gamma \in \mathbb{R}$, as done in Chiarella *et al.* [12]. But others jump-amplitude measures can be selected. For example, in the Kou model [20] the law of $\log(1+J_1)$ is a mixture of exponential laws: the probability density functions is

$$p_{\log(1+J_1)}(x) = p\lambda_+ e^{-\lambda_+ x}\,\mathbf{1}_{x>0} + (1-p)\lambda_- e^{\lambda_- x}\,\mathbf{1}_{x<0},$$

$\mathbf{1}$ denoting the indicator function. Here, the parameters $\lambda_\pm > 0$ control the decrease of the distribution tails of negative and positive jumps respectively, and p is the probability of a positive jump.

3 The 2-dimensional tree for (V, X)

We consider an approximation for the pair (V, X) on the time-interval $[0, T]$ by means of a 2-dimensional computationally simple tree. This means that we construct a Markov chain running over a 2-dimensional recombining bivariate lattice and, at each time-step, both components of the Markov chain can jump only upwards or downwards. We consider the "multiple-jumps" approach by Nelson and Ramaswamy [25]. This technique has already been introduced and used in [7] for Heston-Hull-White types model. A detailed description of this procedure and of the benefits of its use, can be found in [7, 2, 6]. Here, we limit the argumentation to the essential ideas and to the main steps in order to apply the whole algorithm.

We start by considering a discretization of the time-interval $[0, T]$ in N subintervals $[nh, (n+1)h]$, $n = 0, 1, \ldots, N$, with $h = T/N$.

For the CIR volatility process V, we consider the binomial tree procedure firstly introduced in [2]. For $n = 0, 1, \ldots, N$, consider the lattice

$$\mathcal{V}_n^h = \{v_{n,k}\}_{k=0,1,\ldots,n} \quad \text{with} \quad v_{n,k} = \left(\sqrt{V_0} + \frac{\sigma}{2}(2k-n)\sqrt{h}\right)^2 \mathbf{1}_{\sqrt{V_0}+\frac{\sigma}{2}(2k-n)\sqrt{h}>0}. \tag{3.1}$$

Notice that $v_{0,0} = V_0$. For each fixed $v_{n,k} \in \mathcal{V}_n^h$, we define the "up" and "down" jump by means of

$$k_u^h(n,k) = \min\{k^* \,:\, k+1 \le k^* \le n+1 \text{ and } v_{n,k} + \mu_V(v_{n,k})h \le v_{n+1,k^*}\}, \tag{3.2}$$

$$k_d^h(n,k) = \max\{k^* \,:\, 0 \le k^* \le k \text{ and } v_{n,k} + \mu_V(v_{n,k})h \ge v_{n+1,k^*}\} \tag{3.3}$$

where the drift μ_V of V is defined in (2.6) and with the understanding $k_u^h(n,k) = n+1$ if the set in the r.h.s. of (3.2) is empty and $k_d^h(n,k) = 0$ if the set in the r.h.s. of (3.3) is empty. The transition probabilities are defined as follows: starting from the node (n,k) the probability that the process jumps to $k_u^h(n,k)$ and $k_d^h(n,k)$ at time-step $n+1$ are set as

$$p_u^{V,h}(n,k) = 0 \vee \frac{\mu_V(v_{n,k})h + v_{n,k} - v_{n+1,k_d^h(n,k)}}{v_{n+1,k_u^h(n,k)} - v_{n+1,k_d^h(n,k)}} \wedge 1 \quad \text{and} \quad p_d^{V,h}(n,k) = 1 - p_u^{V,h}(n,k) \tag{3.4}$$

respectively. This gives rise to a Markov chain $(\hat{V}_n^h)_{n=0,\dots,N}$ that weakly converges, as $h \to 0$, to the diffusion process $(V_t)_{t\in[0,T]}$ and turns out to be a robust tree approximation for the CIR process V. Remark that this holds independently of the validity of the Feller condition, see [2].

In Figure 1 we show an example of the lattice $\mathcal{V}_n^h$ together with some possible instances of the triple $(v_{n,k}, v_{n+1,k_d^h(n,j)}, v_{n+1,k_u^h(n,j)})$. Notice that, by construction, the lattice is never negative and reaches 0 (the actual possibility that the discretized process touches 0 and remains in 0 is linked to the transition probabilities and then to the parameter values).

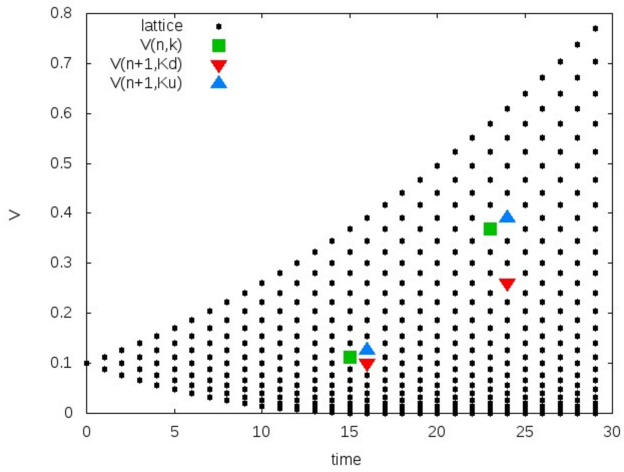

Figure 1: Example of a tree for the process V, showing as the tree may be visited.

Concerning the binomial tree for the process X, for $n = 0, 1, \dots, N$ consider the lattice

$$\mathcal{X}_n^h = \{x_{n,j}\}_{j=0,1,\dots,n} \quad \text{with} \quad x_{n,j} = (2j - n)\sqrt{h}. \tag{3.5}$$

Notice that $x_{0,0} = 0 = X_0$. For each fixed $x_{n,j} \in \mathcal{X}_n^h$, we define the "up" and "down" jump by means of $j_u^h(n,j)$ and $j_d^h(n,j)$ defined by

$$j_u^h(n,j) = \min\{j^* \,:\, j+1 \le j^* \le n+1 \text{ and } x_{n,j} + \mu_X(x_{n,j})h \le x_{n+1,j^*}\}, \tag{3.6}$$

$$j_d^h(n,j) = \max\{j^* \,:\, 0 \le j^* \le j \text{ and } x_{n,j} + \mu_X(x_{n,j})h \ge x_{n+1,j^*}\}, \tag{3.7}$$

μ_X being the drift of the process X, see (2.6). As usual, one sets $j_u^h(n,j) = n+1$ if the set in the r.h.s. of (3.6) is empty and $j_d^h(n,j) = 0$ if the set in the r.h.s. of (3.7) is empty. The transition probabilities are defined as follows: starting from the node (n,j), the probability that the process jumps to $j_u^h(n,j)$ and $j_d^h(n,j)$ at time-step $n+1$ are set as

$$p_u^{X,h}(n,j) = 0 \vee \frac{\mu_X(x_{n,j})h + x_{n,j} - x_{n+1,j_d^h(n,j)}}{x_{n+1,j_u^h(n,j)} - x_{n+1,j_d^h(n,j)}} \wedge 1 \quad \text{and} \quad p_d^{X,h}(n,j) = 1 - p_u^{X,h}(n,j) \tag{3.8}$$

respectively. This gives rise to a Markov chain $(\hat{X}_n^h)_{n=0,\dots,N}$ that weakly converges, as $h \to 0$, to the diffusion process $(X_t)_{t\in[0,T]}$ and turns out to be a robust tree approximation for the OU process X.

An example of such procedure is given in Figure 2, which draws the lattice $\mathcal{X}_n^h$ and possible instances of $x_{n,j}$, $x_{n+1,j_d^h(n,j)}$ and $x_{n+1,j_u^h(n,j)}$.

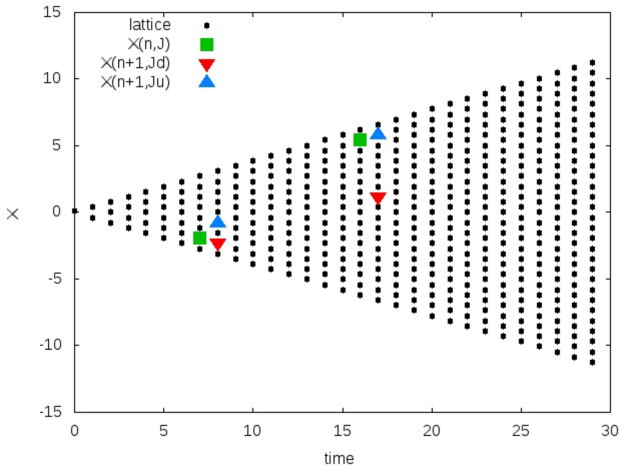

Figure 2: Example of a tree for the process X, showing as the tree may be visited.

The whole tree procedure for the pair (V, X) is obtained by joining the trees built for V and for X. Namely, for $n = 0, 1, \ldots, N$, consider the lattice

$$\mathcal{V}_n^h \times \mathcal{X}_n^h = \{(v_{n,k}, x_{n,j})\}_{k,j=0,1,\ldots,n}. \tag{3.9}$$

Starting from the node (n, k, j), which corresponds to the position $(v_{n,k}, x_{n,j}) \in \mathcal{V}_n^h \times \mathcal{X}_n^h$, we define the four possible jumps by means of the following four nodes at time $n + 1$:

$$\begin{aligned}
&(n+1, k_u^h(n,k), j_u^h(n,j)) \quad \text{with probability} \quad p_{uu}^h(n,k,j) = p_u^{V,h}(n,k)p_u^{X,h}(n,j),\\
&(n+1, k_u^h(n,k), j_d^h(n,j)) \quad \text{with probability} \quad p_{ud}^h(n,k,j) = p_u^{V,h}(n,k)p_d^{X,h}(n,j),\\
&(n+1, k_d^h(n,k), j_u^h(n,j)) \quad \text{with probability} \quad p_{du}^h(n,k,j) = p_d^{V,h}(n,k)p_u^{X,h}(n,j),\\
&(n+1, k_d^h(n,k), j_d^h(n,j)) \quad \text{with probability} \quad p_{dd}^h(n,k,j) = p_d^{V,h}(n,k)p_d^{X,h}(n,j),
\end{aligned} \tag{3.10}$$

where the above nodes $k_u^h(n,k)$, $k_d^h(n,k)$, $j_u^h(n,j)$ and $j_d^h(n,j)$ and the above probabilities $p_u^{V,h}(n,k)$, $p_d^{V,h}(n,k)$, $p_u^{X,h}(n,j)$ and $p_d^{X,h}(n,j)$ are defined in (3.2)-(3.3), (3.6)-(3.7), (3.4) and (3.8). The factorization of the jump probabilities in (3.10) follows the orthogonality property of the noises driving the two processes. As a quite immediate consequence of standard results (see e.g. the techniques in [25]), one gets the following: the associated bivariate Markov chain $(\hat{V}_n^h, \hat{X}_n^h)_{n=0,\ldots,N}$ weakly converges to the diffusion pair $(V_t, X_t)_{t\in[0,T]}$ solution to

$$\begin{aligned}
dV_t &= \mu_V(V_t)dt + \sigma_V\sqrt{V_t}\, dW_t^1, \quad V_0 > 0,\\
dX_t &= -\kappa_r X_t\, dt + \sigma_r\, dW_t^2, \quad X_0 = 0.
\end{aligned}$$

More details and remarks on the extension of this procedure to more general cases can be found in [7], see in particular Remark 3.1 therein in the case of non null correlation between V and X.

4 The approximation on the Y-component

In this section we describe how to manage the Y-component in (2.3) and how to discretize it by taking into account the tree procedure given for the pair (V, X). We will follow two different approaches: the first one is based on the standard Euler scheme for the Y-component and the second one is based on the

Feynman-Kac representation formula and on the finite difference approximation of partial differential problems.

We go now back to (2.3), that is

$$\begin{aligned} dY_t &= \mu_Y(V_t, X_t, t)dt + \sqrt{V_t}\,(\rho_1 dW_t^1 + \rho_2 dW_t^2 + \rho_3 dW_t^3) + dN_t, \quad Y_0 = \ln S_0 \in \mathbb{R}, \\ dV_t &= \mu_V(V_t)dt + \sigma_V\sqrt{V_t}\,dW_t^1, \quad V_0 > 0, \\ dX_t &= \mu_X(X_t)dt + dW_t^2, \quad X_0 = 0, \end{aligned} \tag{4.1}$$

with μ_Y, μ_V and μ_X given by

$$\mu_Y(v,x,t) = \sigma_r x + \varphi_t - \eta - \frac{1}{2}\,v, \quad \mu_V(v) = \kappa_V(\theta_V - v), \quad \mu_X(x) = -\kappa_r x.$$

By isolating $\sqrt{V_t}dW_t^1$ in the second line and dW_t^2 in the third one, we obtain

$$dY_t = \mu(V_t, X_t, t)dt + \rho_3\sqrt{V_t}\,dW_t^3 + \frac{\rho_1}{\sigma_V}dV_t + \rho_2\sqrt{V_t}dX_t + dN_t \tag{4.2}$$

with

$$\begin{aligned} \mu(v,x,t) &= \mu_Y(v,x,t) - \tfrac{\rho_1}{\sigma_V}\mu_V(v) - \rho_2\sqrt{v}\,\mu_X(x) \\ &= \sigma_r x + \varphi_t - \eta - \tfrac{1}{2}\,v - \tfrac{\rho_1}{\sigma_V}\,\kappa_V(\theta_V - v) + \rho_2\kappa_r x\sqrt{v}. \end{aligned} \tag{4.3}$$

To numerically solve (4.2), we mainly use the fact that the noises W^3 and N are independent of the processes V and X. We construct an approximating process $\bar{Y}_t^h$ as follows: we first take the approximating tree $(\hat{V}_n^h, \hat{X}_n)_{n=0,1,\dots,N-1}$ discussed in Section 3 and we call $(\bar{V}_t^h, \bar{X}_t^h)_{t\in[0,T]}$ the associated time-continuous càdlàg approximating process for (V,X), that is $\bar{V}_t^h = \hat{V}_{\lfloor t/h\rfloor}^h$ and $\bar{X}_t^h = \hat{X}_{\lfloor t/h\rfloor}^h$. Then, we insert the discretization $(\bar{V}^h, \bar{X}_t^h)$ for (V,X) in the approximating process for Y obtained by freezing the coefficients in (4.2). Therefore, the final approximating process $\bar{Y}_t^h$ is set as follows: $\bar{Y}_0^h = Y_0$ and for $t \in (nh, (n+1)h]$ with $n = 0, 1, \dots, N-1$

$$\begin{aligned} \bar{Y}_t^h = \;& \bar{Y}_{nh}^h + \mu(\bar{V}_{nh}^h, \bar{X}_{nh}^h, nh)(t - nh) + \rho_3\sqrt{\bar{V}_t^h}(W_t^3 - W_{nh}^3) \\ & + \frac{\rho_1}{\sigma_V}(\bar{V}_t^h - \bar{V}_{nh}^h) + \rho_2\sqrt{\bar{V}_t^h}(\bar{X}_t^h - \bar{X}_{nh}^h) + (N_t - N_{nh}). \end{aligned} \tag{4.4}$$

Now, to numerically compute the price of options written on the share price S, we pursue two different approaches.

4.1 The Monte Carlo approach

Let us show how one can simulate a single path by using the tree approximation (3.9) for the couple (V,X) and the standard Euler scheme (4.4) for the Y-component.

Consider the process (Y,V,X) as in (4.1). Let $(\hat{V}_n^h, \hat{X}_n^h)_{n=0,1,\dots,N}$ denote the Markov chain that approximates the pair (V,X), described in Section 3. We construct a sequence $(\hat{Y}_n)_{n=0,1,\dots,N}$ approximating Y at times nh, $n = 0, 1, \dots, N$, by means of the Euler scheme defined in (4.4): we set $\hat{Y}_0^h = Y_0$ and for $t \in [nh, (n+1)h]$ with $n = 0, 1, \dots, N-1$ then

$$\begin{aligned} \hat{Y}_{n+1}^h = \;& \hat{Y}_n^h + \mu(\hat{V}_n^h, \hat{X}_n^h, nh)h + \rho_3\sqrt{h\hat{V}_n^h}\Delta_{n+1} \\ & + \frac{\rho_1}{\sigma_V}(\hat{V}_{n+1}^h - \hat{V}_n^h) + \rho_2\sqrt{\hat{V}_n^h}(\hat{X}_{n+1}^h - \hat{X}_n^h) + (N_{(n+1)h} - N_{nh}), \end{aligned}$$

where μ is defined in (4.3) and $\Delta_1, \dots, \Delta_N$ denote i.i.d. standard normal r.v.'s, independent of the noise driving the chain $(\hat{V}, \hat{X})$. The simulation of $N_{(n+1)h} - N_{nh}$ is straightforward: one first generates a Poisson r.v. K_h of parameter λh and if $K_h > 0$ then also the log-amplitudes $\log(1 + J_k)$ for $k = 1, \dots, K_h$ are simulated. Then, the observed jump of the compound Poisson process is written as the sum of the logarithms of the simulated amplitudes, so that

$$\begin{aligned} \hat{Y}^h_{n+1} = \quad & \hat{Y}^h_n + \mu(\hat{V}^h_n, \hat{X}^h_n, nh)h + \rho_3\sqrt{h\hat{V}^h_n}\Delta_{n+1} \\ & + \frac{\rho_1}{\sigma_V}(\hat{V}^h_{n+1} - \hat{V}^h_n) + \rho_2\sqrt{\hat{V}^h_n}(\hat{X}^h_{n+1} - \hat{X}^h_n) + \sum_{k=1}^{K_h} \log(1 + J_k), \end{aligned} \tag{4.5}$$

in which the last sum is set equal to 0 if $K_h = 0$.

The above simulation scheme is plain: at each time step $n \geq 1$, one let the pair (V, X) evolve on the tree and simulate the process Y by using (4.5). We will refer to this procedure as *hybrid Monte Carlo algorithm*, the word "hybrid" being related to the fact that two different noise sources are considered: we simulate a continuous process in space (the component Y) starting from a discrete process in space (the tree for (V, X)).

4.2 The partial differential approach

We go back now to (4.4). If we set

$$\bar{Z}^h_t = \bar{Y}^h_t - \frac{\rho_1}{\sigma_V}(\bar{V}^h_t - \bar{V}^h_{nh}) - \rho_2\sqrt{\bar{V}^h_{nh}}(\bar{X}^h_t - \bar{X}_{nh}), \quad t \in [nh, (n+1)h] \tag{4.6}$$

then we have

$$\begin{aligned} & d\bar{Z}^h_t = \mu(\bar{V}^h_{nh}, \bar{X}^h_{nh}, nh)dt + \rho_3\sqrt{\bar{V}^h_{nh}}\, dW^3_t, + dN_t \quad t \in (nh, (n+1)h], \\ & \bar{Z}^h_{nh} = \bar{Y}^h_{nh}, \end{aligned} \tag{4.7}$$

that is $\bar{Z}^h$ solves a jump-diffusion stochastic equation with constant coefficients and at time nh it starts from $\bar{Y}^h_{nh}$. Take now a function f: we are interested in approximating

$$\mathbb{E}(f(Y_{(n+1)h}) \mid Y_{nh} = y, V_{nh} = v, X_{nh} = x).$$

By using our scheme and the process $\bar{Z}^h$ in (4.6), we approximate it with the expectation done on the approximating processes, that is

$$\begin{aligned} & \mathbb{E}(f(\bar{Y}^h_{(n+1)h}) \mid \bar{Y}^h_{nh} = y, \bar{V}^h_{nh} = v, \bar{X}^h_{nh} = x) \\ & \quad = \mathbb{E}\big(f(\bar{Z}^h_{(n+1)h} + \frac{\rho_1}{\sigma_V}(\bar{V}^h_{(n+1)h} - \bar{V}^h_{nh}) + \rho_2\sqrt{\bar{V}^h_{nh}}(\bar{X}^h_{(n+1)h} - \bar{X}^h_{nh})) \mid \bar{Z}^h_{nh} = y, \bar{V}^h_{nh} = v, \bar{X}^h_{nh} = x\big). \end{aligned}$$

Since $(\bar{V}^h, \bar{X}^h)$ is independent of the Brownian noise W^3 and on the Poisson process N driving $\bar{Z}^h$ in (4.7), we can write

$$\begin{aligned} & \mathbb{E}(f(\bar{Y}^h_{(n+1)h}) \mid \bar{Y}^h_{nh} = y, \bar{V}^h_{nh} = v, \bar{X}^h_{nh} = x) \\ & \quad = \mathbb{E}\Big(\Psi_f\Big(\tfrac{\rho_1}{\sigma_V}(\bar{V}^h_{(n+1)h} - \bar{V}^h_{nh}) + \rho_2\sqrt{v}(\bar{X}^h_{(n+1)h} - \bar{X}^h_{nh}); y, v, x\Big) \,\Big|\, \bar{V}^h_{nh} = v, \bar{X}^h_{nh} = x\Big), \end{aligned} \tag{4.8}$$

in which

$$\Psi_f(\zeta; y, v, x) = \mathbb{E}(f(\bar{Z}^h_{(n+1)h} + \zeta) \mid \bar{Z}^h_{nh} = y, \bar{V}^h_{nh} = v, \bar{X}^h_{nh} = x). \tag{4.9}$$

Now, in order to compute the above quantity $\Psi_f(\xi)$, we consider a generic function g and set

$$u(t,y;v,x) = \mathbb{E}(g(\bar{Z}^h_{(n+1)h}) \mid \bar{Z}^h_t = y, \bar{V}^h_t = v, \bar{X}^h_t = x), \quad t \in [nh,(n+1)h].$$

By (4.7) and the Feynman-Kac representation formula we can state that, for every fixed $x \in \mathbb{R}$ and $v \geq 0$, the function $(t,y) \mapsto u(t,y;v,x)$ is the solution to

$$\begin{cases} \partial_t u(t,y;v,x) + L^{(v,x)} u(t,y;v,x) = 0 & y \in \mathbb{R}, t \in [nh,(n+1)h), \\ u((n+1)h,y;v) = g(y) & y \in \mathbb{R}, \end{cases} \tag{4.10}$$

where $L^{(v,x)}$ is the integro-differential operator

$$\begin{aligned} L^{(v,x)} u(t,y;v,x) = \; & \mu(v,x)\partial_y u(t,y;v,x) + \tfrac{1}{2}\rho_3^2 v \partial^2_{yy} u(t,y;v,x) \\ & + \int_{-\infty}^{+\infty} \left[u(t,y+\xi;v,x) - u(t,y;v,x)\right] \nu(\xi) d\xi, \end{aligned} \tag{4.11}$$

where μ is given in (4.3) and ν is the Lévy measure associated to the compound Poisson process N, see (2.7). We are assuming here that the Lévy measure is absolutely continuous (in practice, we use a Gaussian density), but it is clear that the procedure we are going to describe can be straightforwardly extended to other cases.

FIN QUIIIIIIIIIIIIIIIIIIIIIIIIIIII

4.2.1 Finite differences and numerical quadrature

In order to numerically solve the partial integro differential equation (hereafter PIDE) (4.10), we generalize the approach already developed in [6] and [7]: at each time step n, we apply a finite difference algorithm to the differential part of the problem coupled now with a quadrature rule to approximate the integral term.

We start by fixing an infinite grid on the y-axis $\mathcal{Y} = \{y_i = Y_0 + i\Delta y\}_{i\in\mathbb{Z}}$, with $\Delta y = y_i - y_{i-1}$, $i \in \mathbb{Z}$. For fixed n and given $x \in \mathbb{R}$ and $v \geq 0$, we set $u^n_i = u(nh, y_i; v, x)$ the discrete solution of (4.10) at time nh on the point y_i of the grid $\mathcal{Y}$ - for simplicity of notations, in the sequel we do not stress in u^n_i the dependence on (v,x).

First of all, to numerically compute the integral term in (4.11) we need to truncate the infinite integral domain to a bounded interval $\mathcal{I}$, to be taken large enough in order that

$$\int_{\mathcal{I}} \nu(\xi) d\xi \approx 1. \tag{4.12}$$

In terms of the process, this corresponds to truncate the large jumps. Usually, the tails of ν decrease exponentially, so the probability of large jumps is very small. To simplify the following discussion we assume the Lèvy measure $\nu(\xi)d\xi$ to be symmetric with respect to a point ξ_0 and to go to zero outside $\mathcal{I}$ - this property is quite standard, however our procedure can be adapted to other selected properties for ν. Hence, we take $R \in \mathbb{N}$ large enough, set $\mathcal{I} = [\xi_0 - R\Delta y, \xi_0 + R\Delta y]$ and apply to (4.12) the trapezoidal rule on the grid $\mathcal{Y}$ with the same step Δy previously defined. Then, for $\xi_k = \xi_0 + k\Delta y$, $k = -R, \ldots, R$,

$$\int_{\xi_0 - R\Delta y}^{\xi_0 + R\Delta y} \left[u(s,y+\xi) - u(s,y)\right] \nu(\xi) d\xi \approx \Delta y \sum_{l=-R}^{R} \left(u(s,y+\xi_l) - u(s,y)\right) \nu(\xi_l). \tag{4.13}$$

We adjust the numerical grid such that $y_i + \xi_k = Y_0 + \xi_0 + (i+k)\Delta y \in \mathcal{Y}$ and then, for any i, k, the values $u(s, y_i + \xi_k)$ are well defined on the numerical grid $\mathcal{Y}$. These are technical settings and can be modified and calibrated for different Lévy measures ν.

But in practice one cannot solve the PIDE problem over the whole real line. So, we have to choose artificial bounds and impose numerical boundary conditions. We take a positive integer $M > 0$ and we define a finite grid $\mathcal{Y}_M = \{y_i = Y_0 + i\Delta y\}_{i\in\mathcal{J}_M}$, with $\mathcal{J}_M = \{-M, \dots, M\}$, and we assume that $M > R$. Fixing now a time $s = nh$ and a grid node $i \in \mathcal{J}_M$, for $y = y_i$ we have that the integral term in (4.13) splits into two parts: one part concerning nodes falling into the numerical domain $\mathcal{Y}_M$ and another part concerning nodes falling out of $\mathcal{Y}_M$. As an example,

$$\sum_{l=-R}^{R} u(nh, y_i + \xi_l)\nu(x_l) \approx \sum_{l=-R}^{R} u_{i+l}^n \nu(\xi_l) = \sum_{l\,:\,|l|\le R, |i+l|\le M} u_{i+l}^n \ \nu(\xi_l) + \sum_{l\,:\,|l|\le R, |i+l|> M} \tilde{u}_{i+l}^n \ \nu(\xi_l),$$

where $\tilde{u}_\cdot^n$ stands for *unknown* values that fall out of the finite numerical domain $\mathcal{Y}_M$. This implies that we must choose some suitable artificial boundary conditions. In [31] it has been shown that a good choice for the boundary conditions is the payoff function. Although this is the choice we will do in our numerical experiments, for the shake of generality we assume here the boundary values outside the $\mathcal{Y}_M$ domain to be settle by a known function $b = b(t, y)$ which is defined in all $\mathbb{R}^+ \times \mathbb{R}$.

Going back to the numerical scheme to solve the differential part of the equation (4.10), as already done in [7], we would apply an implicit in time approximation. However, to avoid to solve at each time step a linear system with a dense matrix, the non-local integral term needs anyway an explicit in time approximation. We then obtain an implicit-explicit (hereafter IMEX) scheme as proposed in [15] and [8]. Notice that, more sophisticated IMEX methods may be applied, see for instance [9, 27]. The use of these techniques however increases the computational costs and it is beyond the scope of this work.

As done in [7], to achieve greater precision we use the centered approximation for both first and second order derivatives in space. The discrete solution u^n at time nh is then computed in terms of the solution u^{n+1} at time $(n+1)h$ by solving the following discrete problem: for all $i \in \mathcal{J}_M$,

$$\frac{u_i^{n+1} - u_i^n}{h} + \tilde{\mu}_Y(v, x)\frac{u_{i+1}^n - u_{i-1}^n}{2\Delta y} + \frac{1}{2}\rho_3^2 \ v \ \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta y^2} + \Delta y \sum_{l=-R}^{R} \left(u_{i+l}^{n+1} - u_i^{n+1}\right)\nu(\xi_l) = 0. \tag{4.14}$$

We then get the solution $u^n = (u_{-M}^n, \dots, u_M^n)^T$ by solving the following linear system

$$A\, u^n = B u^{n+1} + d, \tag{4.15}$$

where $A = A(v, x)$ and B are $(2M+1) \times (2M+1)$ matrices and d is a $2M+1$ vector defined as follows. Concerning A, it is the tridiagonal real matrix given by

$$A = \begin{pmatrix} 1+2\beta & -\alpha-\beta & & & \\ \alpha-\beta & 1+2\beta & -\alpha-\beta & & \\ & \ddots & \ddots & \ddots & \\ & & \alpha-\beta & 1+2\beta & -\alpha-\beta \\ & & & \alpha-\beta & 1+2\beta \end{pmatrix}, \tag{4.16}$$

with

$$\alpha = \frac{h}{2\Delta y}\,\mu(v, x) \quad \text{and} \quad \beta = \frac{h}{2\Delta y^2}\,\rho_3^2 v, \tag{4.17}$$

μ being defined in (4.3). We stress on that at each time step n, the quantities v and x are constant and known values (defined by the tree procedure for (V, X)) and then α and β are constant parameters. B is the $(2M+1) \times (2M+1)$ real matrix given by

$$B = I + h\Delta y \begin{pmatrix} \nu(\xi_0) - \Lambda & \nu(\Delta y) & \dots & \nu(R\Delta y) & 0 \\ \nu(-\Delta y) & \nu(\xi_0) - \Lambda & \nu(\Delta y) & \dots & \nu(R\Delta y) \\ & \ddots & \ddots & \ddots & \\ 0 & \nu((-R+1)\Delta y) & \dots & \nu(-\Delta y) & \nu(\xi_0) - \Lambda \end{pmatrix}, \tag{4.18}$$

where I is the identity matrix and $\Lambda = \sum_{l=-R}^{R} \nu(\xi_l)$. The $2M+1$ real vector d contains the numerical boundary values:

$$d = a_b^n + a_b^{n+1}, \tag{4.19}$$

with

$$a_b^n = ((\beta - \alpha) b_{-M-1}^n, 0, \dots, 0, (\beta + \alpha) b_{M+1}^n)^T \in \mathbb{R}^{2M+1}$$

and $a_b^{n+1} \in \mathbb{R}^{2M+1}$ is such that

$$(a_b^{n+1})_i = \begin{cases} h\Delta y \sum_{l=-R}^{-M-i-1} \nu(x_l)\ b_{i+l}^{n+1}, & \text{for } i = -M, \dots, -M+R-1, \\ 0 & \text{for } i = -M+R, \dots, M-R, \\ h\Delta y \sum_{l=M-i+1}^{R} \nu(x_l)\ b_{i+l}^{n+1}, & \text{for } i = M-R+1, \dots, M-1, \end{cases} \tag{4.20}$$

where for standard notation $b_i^n = b(nh, y_i)$, $n \in \mathbb{N}$, $i \in \mathbb{Z}$.

We then obtain the following complete scheme. For $\beta \neq |\alpha|$, A is an invertible matrix and at each time nh, for each fixed $v \geq 0$ and $x \in \mathbb{R}$, the discrete solution $u^n = \{u_i^n\}_{i \in \mathcal{J}_M}$ of (4.10) is given in terms of the solution $u^{n+1} = \{u_i^{n+1}\}_{i \in \mathcal{J}_M}$ at time $(n+1)h$, by the following formula:

$$u(nh, y_i; v) \approx \sum_{j \in \mathcal{J}_{M_h}} \Pi_{ij}(v) g(y_j) + \tilde{d}_i(v), \quad i \in \mathcal{J}_M. \tag{4.21}$$

where $\Pi(v) = A^{-1}(v)B$ and $\tilde{d}(v) = A^{-1}(v)d$.

4.2.2 The final finite difference approximation

We can now come back to our original problem, that is the computation of the function $\Psi_f(\xi; y, v, x)$ in (4.9) allowing one to numerically compute the expectation in (4.8). This means that, at time step n, the pair (v, x) is chosen on the lattice $\mathcal{V}_n^h \times \mathcal{X}_n^h$: $v = v_{n,k}$, $x = x_{n,j}$ for $0 \leq k, j \leq n$. Then, (4.21) gives

$$\Psi_f(\zeta; y, v, x) \simeq \sum_{l \in \mathcal{J}_M} \Pi_{il}(v_{n,k}, x_{n,j}) f(y_l + \zeta) + \tilde{d}_i(v_{n,k}), \quad i \in \mathcal{J}_M. \tag{4.22}$$

Therefore, the expectation in (4.8) is finally computed on the approximating tree for (V, X) by means of the above approximation:

$$\begin{aligned} &\mathbb{E}(f(\bar{Y}_{(n+1)h}^h) \mid \bar{Y}_{nh}^h = y, \bar{V}_{nh}^h = v, \bar{X}_{nh}^h = x) \\ &\simeq \sum_{a,b \in \{u,d\}} \sum_{l \in \mathcal{J}_M} \Big(\Pi_{il}(v_{n,k}, x_{n,j}) f\big(y_l + \frac{\rho_1}{\sigma_V}(v_{(n+1),k_a(n,k)} - v) + \rho_2 \sqrt{v}(x_{(n+1),j_b(n,j)} - x)\big) \\ &\qquad + \tilde{d}_i(v_{n,k}) \Big) p_{ab}^h(n,k,j). \end{aligned} \tag{4.23}$$

5 The algorithm for the pricing of American options

In order to outline our numerical procedure, we show here how to use our hybrid tree/finite difference approximation (4.23) for the pricing of American options. Consider an American option with maturity T and payoff function $(\Phi(S_t))_{t\in[0,T]}$. We consider the log-price process, so the obstacle will be given by

$$\Psi(Y_t) = \Phi(e^{Y_t}), \quad t \in [0,T].$$

The price $P(t,y,v,x)$ of such an American option is then given by

$$P(t,y,v,x) = \sup_{\tau\in\mathcal{T}_{t,T}} \mathbb{E}\Big(e^{-\int_t^\tau (\sigma_r X_s^{t,x}+\varphi_s)ds}\Psi(Y_\tau^{t,y,v,x})\Big),$$

where $\mathcal{T}_{t,T}$ denotes the set of all stopping times taking values on $[t,T]$ and where we used the relation between the interest rate r and the process X: $r_t = \sigma_r X_t + \varphi_t$ (see (2.1) and (2.2)). Hereafter, $(Y^{t,y,v,x}, V^{t,v}, X^{t,x})$ denotes the solution of the jump-diffusion dynamic (2.3) starting at (y,v,x) at time t.

The price at time 0 of such an option is then approximated by a backward dynamic programming algorithm, working as follows. First, consider a discretization of the time interval $[0,T]$ into N subintervals of length $h = T/N$: $[0,T] = \cup_{n=0}^{N-1}[nh,(n+1)h]$. Then the price $P(0,Y_0,V_0,X_0)$ of such an American option is numerically approximated through the quantity $P_h(0,Y_0,V_0,X_0)$ which is iteratively defined as follows: for $(y,v,x)\in\mathbb{R}\times\mathbb{R}_+\times\mathbb{R}$, we have

$$\begin{cases} P_h(T,y,v,x) = \Psi(y) \\ \text{and as } n = N-1,\dots,0 \\ P_h(nh,y,v,x) = \max\Big\{\Psi(y), e^{-\int_{nh}^{(n+1)h}(\sigma_r X_t^{nh,x}+\varphi_t)dt}\mathbb{E}\Big(P_h\big((n+1)h, Y_{(n+1)h}^{nh,y,v,x}, V_{(n+1)h}^{nh,v}, X_{(n+1)h}^{nh,x}\big)\Big)\Big\}. \end{cases}$$

From the financial point of view, this means to allow the exercise at the fixed times nh, for $n = 0,\dots,N$.

We consider now the discretization scheme $(\bar{Y}^h, \bar{V}^h, \bar{X}^h)$ discussed in Section 4.2 and we use the approximation (4.23) for the conditional expectations that have to be computed at each time step n. So, for every point $(y_i, v_{n,k}, x_{n,j}) \in \mathcal{Y}_M \times \mathcal{V}_n^h \times \mathcal{X}_n^h$, by (4.23) we have

$$\begin{aligned} &\mathbb{E}\Big(P_h\big((n+1)h, Y_{(n+1)h}^{nh,y_{n,i,k,j},v_{n,k},x_{n,j}}, V_{(n+1)h}^{nh,v_{n,k}}, X_{(n+1)h}^{nh,x_{n,j}}\big)\Big) \\ &\simeq \sum_{a,b\in\{d,u\}}\sum_{\ell\in\mathcal{J}_M}\Big(\Pi_{i\ell}^h(v_{n,k},x_{n,j})\mathcal{S}_{n,k,j}P_h(\ell,a,b) + \tilde{d}_i(v_{n,k})\Big)p_{ab}^h(n,k,j) \end{aligned} \tag{5.1}$$

where $\mathcal{S}_{n,k,j}P_h$ denotes

$$\begin{aligned} &\mathcal{S}_{n,k,j}P_h(\ell,a,b) \\ &= P_h\Big((n+1)h, y_\ell + \frac{\rho_1}{\sigma_V}(v_{n+1,k_a(n,k)} - v_{n,k}) + \rho_2\sqrt{v_{n,k}}(x_{n+1,j_b(n,j)} - x_{n,j}), v_{n+1,k_a(n,k)}, x_{n+1,j_b(n,j)}\Big). \end{aligned} \tag{5.2}$$

Finally, we can summarize the backward induction giving our approximating algorithm as follows. For $n = 0,1,\dots,N$, we define $\tilde{P}_h(nh,y,v,x)$ for $(y,v,x)\in\mathcal{Y}_M\times\mathcal{V}_n^h\times\mathcal{X}_n^h$ as

$$\begin{cases} \tilde{P}_h(T,y_i,v_{N,k},x_{N,j}) = \Psi(y_i) \quad \text{for } (y_i,v_{N,k},x_{N,j})\in\mathcal{Y}_M\times\mathcal{V}_N^h\times\mathcal{X}_N^h \\ \text{and as } n = N-1,\dots,0: \\ \tilde{P}_h(nh,y_i,v_{n,k},x_{n,j}) = \max\Big\{\Psi(y_i), e^{-(\sigma_r x_{n,j}+\varphi_{nh})h}\times \\ \qquad\qquad \times\sum_{a,b\in\{d,u\}}\sum_{\ell\in\mathcal{J}_M}\Pi_{i\ell}^h(v_{n,k},x_{n,j})p_{ab}^h(n,k,j)\mathrm{T}_{n,k,j}P_h(\ell,a,b)\Big\}, \\ \qquad\qquad \text{for } (y_i,v_{n,k},x_{n,j})\in\mathcal{Y}_M\times\mathcal{V}_n^h\times\mathcal{X}_n^h. \end{cases} \tag{5.3}$$

6 A schematic sketch of the major computational steps in our algorithms

To summarize, we resume here the main computational steps of the two proposed algorithms.

Both hybrid Monte Carlo and hybrid tree/finite difference procedure have the same preprocessing steps:

(p1) define a discretization of the time-interval $[0,T]$ in N subintervals $[nh,(n+1)h]$, $n=0,\dots,N$, with $h=T/N$;

(p2) set the binomial tree $x_{n,j}$, $0\le j\le n\le N$ for the process X by using (3.5);

(p3) set the binomial tree $v_{n,k}$, $0\le k\le n\le N$ for the process V by using (3.1);

(p4) merge the binomial trees in the bivariate tree $(v_{n,k},x_{n,j})$, $0\le k,j\le n\le N$, by using (3.9);

(p5) compute the jump-nodes and the transitions probabilities p_{ab}, $(a,b)\in\{d,u\}$, using (3.10).

The bivariate tree for (V,X) has been now settled.

One way is now to pursue by applying the hybrid Monte Carlo method:

1. compute $\Delta_1,\dots,\Delta_N$ i.i.d. standard normal r.v. independent of the noise driving the chain in (V,X);
2. generate a positive Poisson r.v. K_h of parameter λh;
3. simulate the amplitudes J_k for $k=1,\dots,K_h$;
4. starting from $\hat{Y}_0^h=Y_0$, compute the approximates values $\hat{Y}_n^h$, $0\le n\le N-1$, using (4.5).
5. compute the option price ... COMPLETARE ...

On the other hand, after the preprocessing step (p5) one may pursue by applying the hybrid tree/finite difference approach by following the next steps:

(i) set a mesh grid y_i for the solution of all the PDE's;

(ii) for each node $(v_{N,k},x_{N,j})$, $0\le k,j\le N$, compute the option prices at maturity for each y_i, $i\in\mathcal{Y}_M$, by using the payoff function.

(iii) for $n=N-1,\dots 0$: for each $(v_{n,k},x_{n,j})$, $0\le k,j\le n$, compute the option prices for each y_i, $i\in\mathcal{Y}_M$, by solving PIDE (4.10) through (4.21), with terminal condition given by the weighted sum of the values at nodes $(a,b)\in\{u,d\}$ which have been computed in the previous step $n+1$ - weight by using the transition probabilities p_{ab}.

Remark 6.1. *We observe here that to compute the option price by the hybrid tree/finite difference procedure, in step (iii) we need to solve many times the tridiagonal system (4.15). It is solved by the LU-decomposition method in $O(M)$ operations, being M the total number of grid values $y_i\in\mathcal{Y}_M$. However, due to the approximation of the integral term (4.13), each time we have to compute the sum*

$$\sum u_{i+l}^n\nu(x_l), \tag{6.1}$$

which is the most computationally expensive step of this part of the algorithm: when applied directly it requires $O(M^2)$ operations. Following the Premia software implementation [26], we apply the Fast Fourier Transform (FFT) to compute the term (6.1) and the computational costs of this step reduce to $O(M\log M)$.

7 Numerical results

In this section we provide numerical results in order to asses the efficiency and the robustness of the hybrid tree-finite difference method and the hybrid Monte Carlo method in the case of plain vanilla options. We provide results for the (standard) Bates model and also under the case in which the interest rate process is assumed to be stochastic, the latter being a straightforward generalization.

In our numerical tests, following Chiarella *et al.* [12] we assume that

$$\log(1+J_1) \sim N\big(\gamma - \frac{1}{2}\delta^2, \delta^2\big). \tag{7.1}$$

7.1 The standard Bates model

In the European and American option contracts we are dealing with, we consider the following set of parameters used in the numerical results provided in Chiarella *et al.* [12]:

- initial prices $S_0 = 80, 90, 100, 110, 120$, strike price $K = 100$, maturity $T = 0.5$;
- interest rate $r = 0.03$, dividend rate $\eta = 0.05$;
- initial volatility $V_0 = 0.04$, long-mean $\theta = 0.04$, speed of mean-reversion $\kappa = 2$, volatility of volatility $\sigma = 0.4$, correlations $\rho = -0.5, 0.5$;
- intensity $\lambda = 5$, jump parameters $\gamma = 0$ and $\delta = 0.1$ (recall (7.1)).

The numerical study of the hybrid tree-finite difference method **HTFD** is split in two cases: **HTFD1** refers to the (fixed) number of time steps $N_t = 50$ and varying mesh grid $\Delta y_h = \Delta y = 0.01$, 0.005, 0.0025, 0.00125; we add the situation **HTFD2** where the number of time steps is equal to $N_t = 100$, with the same values of Δy as before.

Similarly, the numerical study of the hybrid Monte Carlo method **HMC** is split in two cases: **HMC1** refers to the (fixed) number of time steps $N_t = 50$ and varying Monte Carlo iterations $N_{MC} = 10000, 50000, 100000, 200000$; we add the situation **HMC2** where the number of time steps is equal to $N_t = 100$ and same N_{MC} as before. Concerning the American option case, we have implemented both **HMC1** and **HMC2** by means of the Longstaff-Schwartz method [23].

Table 1 reports European call option prices. Comparisons are given with a benchmark value obtained using the Carr-Madan pricing formula **CF** in [11] that applies Fast-Fourier transform methods (see the Premia software implementation [26]).

In Table 2 we provide results for American call option prices. In this case we compare with the values obtained by using the method of lines in [13] **GIUSTO?????**, called **MOL**, with mesh parameters $200, 250, 2995$ and the **PSOR** method with mesh parameters $1000, 3000, 6000$ that Chiarella *et al.* [12] used as true solution.

Table 3 refers to the computational time cost (in seconds) of the different algorithms for $\rho = -0.5$ in the European case.

The numerical results show that **HTFD** is accurate, reliable and efficient for pricing European and American options in the Bates model. Moreover, our hybrid Monte Carlo algorithm **HMC** provide reliable European prices. As a further evidence of the accuracy of our method, in Figure 3 and 4 we study the shapes of implied volatility smiles across moneyness $\frac{K}{S_0}$ and maturities T using **HTFD1** with $N_t = 50$ and $\Delta y = 0.005$, **HMC1** with $N_t = 50$ and $N_{MC} = 50000$ and we compare the graphs with the results from the benchmark values **CF** .

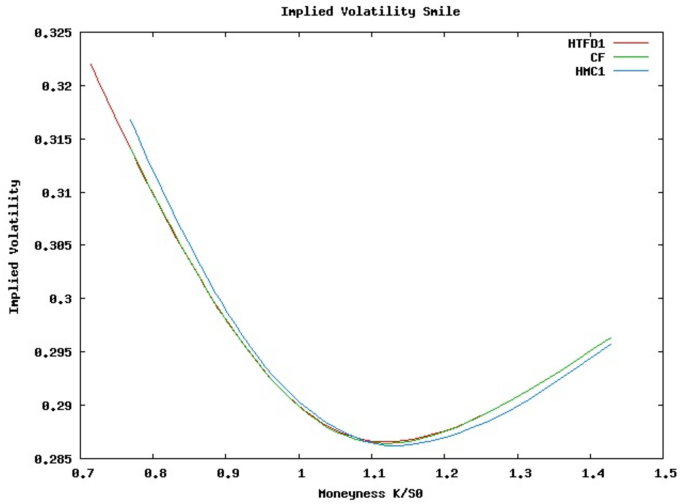

Figure 3: Moneyness vs implied volatility for European call options. $T = 0.5$, $r = 0.03$, $\eta = 0.05$, $V_0 = 0.04$, $\theta = 0.04$, $\kappa = 2$, $\sigma = 0.4$, $\lambda = 5$, $\gamma = 0$, $\delta = 0.1$, $\rho = -0.5$.

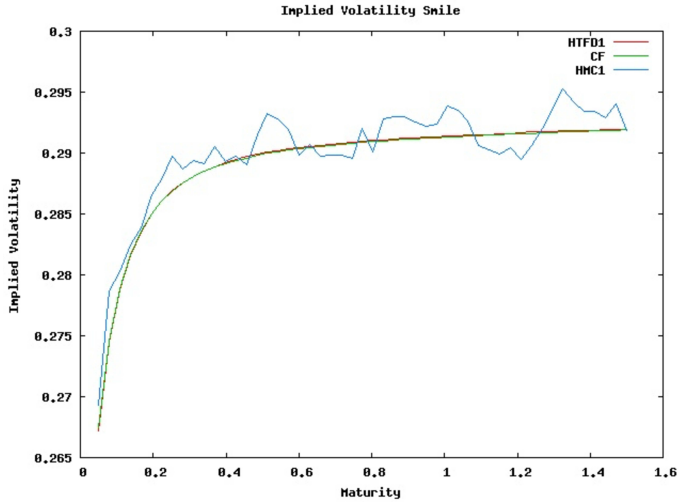

Figure 4: Maturity vs implied volatility for European call options. $S_0 = 100$, $K = 100$, $r = 0.03$, $\eta = 0.05$, $V_0 = 0.04$, $\theta = 0.04$, $\kappa = 2$, $\sigma = 0.4$, $\lambda = 5$, $\gamma = 0$, $\delta = 0.1$, $\rho = -0.5$.

(a)

$\rho = -0.5$	Δy	HTFD1	HTFD2	CF	N_{MC}	HMC1	HMC2
$S_0 = 80$	0.01	1.1302	1.1302		10000	1.08±0.09	1.11±0.09
	0.005	1.1293	1.1294		50000	1.12±0.04	1.17±0.04
	0.0025	1.1291	1.1292	1.1293	100000	1.14±0.03	1.14±0.03
	0.00125	1.1290	1.1291		200000	1.13±0.02	1.14±0.02
$S_0 = 90$	0.01	3.3330	3.3312		10000	3.27±0.17	3.27±0.17
	0.005	3.3316	3.3301		50000	3.32±0.08	3.40±0.08
	0.0025	3.3316	3.3298	3.3284	100000	3.34±0.05	3.34±0.05
	0.00125	3.3310	3.3297		200000	3.32±0.04	3.35±0.04
$S_0 = 100$	0.01	7.5245	7.5238		10000	7.46±0.25	7.46±0.25
	0.005	7.5236	7.5224		50000	7.53±0.11	7.62±0.11
	0.0025	7.5231	7.5221	7.5210	100000	7.54±0.08	7.52±0.08
	0.00125	7.5230	7.5220		200000	7.50±0.06	7.54±0.06
$S_0 = 110$	0.01	13.6943	13.6939		10000	13.69±0.34	13.69±0.34
	0.005	13.6923	13.6924		50000	13.71±0.15	13.81±0.15
	0.0025	13.6918	13.6921	13.6923	100000	13.72±0.11	13.69±0.11
	0.00125	13.6923	13.6920		200000	13.64±0.08	13.71±0.08
$S_0 = 120$	0.01	21.3172	21.3185		10000	21.40±0.41	21.40±0.41
	0.005	21.3156	21.3168		50000	21.35±0.18	21.46±0.19
	0.0025	21.3152	21.3164	21.3174	100000	21.36±0.13	21.32±0.13
	0.00125	21.3152	21.3163		200000	21.25±0.09	21.33±0.09

(b)

$\rho = 0.5$	Δy	HTFD1	HTFD2	CF	N_{MC}	HMC1	HMC2
$S_0 = 80$	0.01	1.4732	1.4744		10000	1.42±0.12	1.40±0.12
	0.005	1.4724	1.4744		50000	1.49±0.06	1.47±0.05
	0.0025	1.4723	1.4742	1.4760	100000	1.48±0.04	1.46±0.04
	0.00125	1.4722	1.4741		200000	1.47±0.03	1.48±0.03
$S_0 = 90$	0.01	3.6849	3.6859		10000	3.63±0.19	3.63±0.19
	0.005	3.6836	3.6849		50000	3.70±0.09	3.70±0.09
	0.0025	3.6832	3.6847	3.6862	100000	3.67±0.06	3.67±0.06
	0.00125	3.6832	3.6847		200000	3.66±0.04	3.70±0.04
$S_0 = 100$	0.01	7.6247	7.6246		10000	7.58±0.28	7.58±0.28
	0.005	7.6238	7.6232		50000	7.66±0.13	7.65±0.13
	0.0025	7.6234	7.6229	7.6223	100000	7.61±0.09	7.59±0.09
	0.00125	7.6233	7.6228		200000	7.58±0.06	7.64±0.06
$S_0 = 110$	0.01	13.4863	13.4835		10000	13.48±0.36	13.48±0.36
	0.005	13.4842	13.4818		50000	13.55±0.17	13.49±0.16
	0.0025	13.4837	13.4814	13.4791	100000	13.47±0.12	13.41±0.12
	0.00125	13.4836	13.4813		200000	13.42±0.08	13.49±0.08
$S0 = 120$	0.01	20.9678	20.9661		10000	21.04±0.44	21.04±0.44
	0.005	20.9659	20.9642		50000	21.05±0.20	20.98±0.20
	0.0025	20.9655	20.9636	20.9616	100000	20.96±0.14	20.87±0.14
	0.00125	20.9654	20.9635		200000	20.88±0.10	20.96±0.10

Table 1: *Prices of European call options.* $K = 100$, $T = 0.5$, $r = 0.03$, $\eta = 0.05$, $V_0 = 0.04$, $\theta = 0.04$, $\kappa = 2$, $\sigma = 0.4$, $\lambda = 5$, $\gamma = 0$, $\delta = 0.1$, $\rho = -0.5, 0.5$.

(a)

$\rho = -0.5$	Δy	HTFD1	HTFD2	PSOR	MOL
$S_0 = 80$	0.01	1.1365	1.1365		
	0.005	1.1356	1.1358		
	0.0025	1.1354	1.1356	1.1359	1.1363
	0.00125	1.1353	1.1355		
$S_0 = 90$	0.01	3.3579	3.3563		
	0.005	3.3564	3.3551		
	0.0025	3.3560	3.3548	3.3532	3.3530
	0.00125	3.3559	3.3547		
$S_0 = 100$	0.01	7.6010	7.6006		
	0.005	7.6001	7.5992		
	0.0025	7.5997	7.5989	7.5970	7.5959
	0.00125	7.5996	7.5989		
$S_0 = 110$	0.01	13.8853	13.8854		
	0.005	13.8836	13.8842		
	0.0025	13.8832	13.8839	13.8830	13.8827
	0.00125	13.8831	13.8838		
$S_0 = 120$	0.01	21.7180	21.7199		
	0.005	21.7168	21.7187		
	0.0025	21.7166	21.7184	21.7186	21.7191
	0.00125	21.7165	21.7183		

(b)

$\rho = 0.5$	Δy	HTFD1	HTFD2	PSOR	MOL
$S_0 = 80$	0.01	1.4817	1.4837		
	0.005	1.4809	1.4830		
	0.0025	1.4807	1.4828	1.4843	1.4848
	0.00125	1.4807	1.4828		
$S_0 = 90$	0.01	3.7134	3.7148		
	0.005	3.7121	3.7139		
	0.0025	3.7118	3.7137	3.7145	3.7146
	0.00125	3.7118	3.7137		
$S_0 = 100$	0.01	7.7044	7.7051		
	0.005	7.7036	7.7039		
	0.0025	7.7033	7.7036	7.7027	7.7018
	0.00125	7.7032	7.7036		
$S_0 = 110$	0.01	13.6770	13.6756		
	0.005	13.6752	13.6742		
	0.0025	13.6747	13.6739	13.6722	13.6715
	0.00125	13.6747	13.6738		
$S_0 = 120$	0.01	21.3668	21.3671		
	0.005	21.3655	21.3658		
	0.0025	21.3653	21.3655	21.3653	21.3657
	0.00125	21.3652	21.3653		

Table 2: *Prices of American call options.* $K = 100$, $T = 0.5$, $r = 0.03$, $\eta = 0.05$, $V_0 = 0.04$, $\theta = 0.04$, $\kappa = 2$, $\sigma = 0.4$, $\lambda = 5$, $\gamma = 0$, $\delta = 0.1$, $\rho = -0.5, 0.5$.

Δy	HTFD1	HTDF2	N_{MC}	HMC1	HMC2
0.01	0.25	0.97	10000	0.18	0.35
0.005	0.49	1.96	50000	0.88	1.73
0.0025	1.09	4.37	100000	1.74	3.45
0.00125	2.35	9.20	200000	3.49	6.88

Table 3: *Computational times (in seconds) for European Call options for* $S_0 = 100$, $\rho = -0.5$.

7.2 Bates model with stochastic interest rate

We consider now the case of Bates model associated with the Hull-White model for the stochasticinterest rate. We consider in particular the initial interest rate $r_0 = 0.03$, the speed of mean-reversion $\kappa_r = 1$, the interest rate volatility $\sigma_r = 0.2$; The time-varying long-term mean $\theta_r(t)$ fit the theoretical bond prices to the yield curve observed on the market. We have chosen for this purpose the following interest rate curve $P_r(0,T) = e^{-0.03T}$. We study the cases $\rho_1 = \rho_{SV} = -0.5$, $\rho_2 = \rho_{Sr} = -0.5, 0, 0.5$ and no correlation is assumed to exist between r and V. We consider now mesh grid $\Delta y_h = \Delta y = 0.02, 0.01, 0.005, 0.0025$ because the cases $\Delta y = 0.00125$ requires huge computational times. The benchmark value **B-MC** is obtained using a Monte Carlo method **MC** by simulating paths through the accurate third-order Alfonsi [1] discretization scheme for the CIR stochastic volatility process and an exact scheme for the interest rate with a huge number of Monte Carlo simulations (1 million iterations) and $N_t = 300$ discretization time steps. In the American case, we consider the Longstaff-Schwartz [23] algorithm **MC-LS** with 20 exercise dates. All Monte Carlo results report the 95% confidence intervals.

Tables 4 and 5 report European and American call option prices. Table 6 refers to the computational time cost (in seconds) of the different algorithms in the call European case. The numerical results confirm the good numerical behavior of **HTFD** and **HMC** in the Bates-Hull-White model as well.

8 Conclusions

In this paper we have extended the hybrid tree/finite difference method already introduced in [6] to the Bates model, which can be seen as the Heston model with jumps. We have also developed a Monte Carlo simulation scheme, which works as usual in the direction of the underlying asset price process but uses a discrete approximation in space (Markov chain) to approximate the volatility process. We use our schemes to numerically evaluate American options. The results turn out to be good and reliable, the comparison with existing pricing methods shows that our numerical methods are efficient also in terms of computing time costs.

Acknowledgements. The authors wish to thank Andrea Molent for having implemented the Alfonsi Monte Carlo scheme and the Longstaff-Schwarz algorithm.

(a)

$\rho_{Sr} = -0.5$	Δy	HTFD1	HTFD2	B-MC	N_{MC}	HMC1	HMC2
$S_0 = 80$	0.02	1.0169	1.0079		10000	1.00±0.09	0.96±0.09
	0.01	1.0201	1.0188		50000	1.02±0.04	0.97±0.04
	0.0050	1.0199	1.0194	1.0153±0.01	100000	1.00±0.03	1.00±0.03
	0.0025	1.0197	1.0193		200000	1.01±0.02	1.01±0.02
$S_0 = 90$	0.01	3.1172	3.1032		10000	3.05±0.16	3.05±0.16
	0.01	3.1186	3.1137		50000	3.10±0.07	3.03±0.07
	0.0050	3.1174	3.1135	3.1008±0.02	100000	3.07±0.05	3.08±0.05
	0.0025	3.1174	3.1136		200000	3.09±0.04	3.10±0.04
$S_0 = 100$	0.02	7.2528	7.2472		10000	7.17±0.24	7.17±0.24
	0.01	7.2528	7.2479		50000	7.21±0.11	7.18±0.11
	0.0050	7.2528	7.2480	7.2315±0.02	100000	7.18±0.08	7.24±0.08
	0.0025	7.2528	7.2480		200000	7.22±0.05	7.25±0.05
$S_0 = 110$	0.02	13.4553	13.4565		10000	13.30±0.32	13.30±0.32
	0.01	13.4465	13.4440		50000	13.37±0.15	13.40±0.15
	0.0050	13.4435	13.4407	13.4256±0.03	100000	13.35±0.10	13.46±0.10
	0.0025	13.4432	13.4404		200000	13.40±0.07	13.47±0.07
$S_0 = 120$	0.02	21.1320	21.1356		10000	20.89±0.40	20.89±0.40
	0.01	21.1243	21.1239		50000	21.03±0.18	21.09±0.18
	0.0050	21.1222	21.1214	21.1070±0.04	100000	21.01±0.13	21.17±0.13
	0.0025	21.1215	21.1207		200000	21.06±0.09	21.16±0.09

(b)

$\rho_{Sr} = 0.5$	Δy	HTFD1	HTFD2	B-MC	N_{MC}	HMC1	HMC2
$S_0 = 80$	0.02	1.3459	1.3379		10000	1.29±0.11	1.28±0.11
	0.01	1.3482	1.3471		50000	1.34±0.05	1.30±0.05
	0.0050	1.3479	1.3475	1.3446±0.01	100000	1.32±0.03	1.31±0.03
	0.0025	1.3477	1.3473		200000	1.33±0.02	1.34±0.02
$S_0 = 90$	0.01	3.7320	3.7233		10000	3.62±0.18	3.62±0.18
	0.01	3.7323	3.7304		50000	3.69±0.08	3.65±0.08
	0.0050	3.7311	3.7298	3.7263±0.02	100000	3.66±0.06	3.68±0.06
	0.0025	3.7311	3.7299		200000	3.69±0.04	3.72±0.04
$S_0 = 100$	0.02	8.0100	8.0073		10000	7.83±0.26	7.83±0.26
	0.01	8.0112	8.0102		50000	7.92±0.12	7.93±0.12
	0.0050	8.0114	8.0107	8.0069±0.03	100000	7.91±0.08	7.97±0.08
	0.0025	8.0114	8.0107		200000	7.95±0.06	8.02±0.06
$S_0 = 110$	0.02	14.1482	14.1505		10000	13.89±0.35	13.89±0.35
	0.01	14.1413	14.1414		50000	14.01±0.16	14.05±0.16
	0.0050	14.1388	14.1388	14.1323±0.03	100000	14.01±0.11	14.10±0.11
	0.0025	14.1386	14.1386		200000	14.06±0.08	14.17±0.08
$S_0 = 120$	0.02	21.6737	21.6772		10000	21.37±0.42	21.37±0.42
	0.01	21.6670	21.6674		50000	21.50±0.19	21.55±0.19
	0.0050	21.6651	21.6653	21.6501±0.04	100000	21.52±0.13	21.63±0.13
	0.0025	21.6645	21.6646		200000	21.57±0.10	21.71±0.10

Table 4: *Prices of European call options.* $K = 100$, $T = 0.5$, $\eta = 0.05$, , $r_0 = 0.03$, $\kappa_r = 1$, $\sigma_r = 0.2$, $V_0 = 0.04$, $\theta = 0.04$, $\kappa = 2$, $\sigma = 0.4$, $\lambda = 5$, $\gamma = 0$, $\delta = 0.1$, $\rho_{SV} = -0.5$,$\rho_{Sr} = -0.5, 0.5$.

(a)

$\rho_{Sr} = -0.5$	Δy	HTFD1	HTFD2	MC-LS
$S_0 = 80$	0.02	1.0561	1.0470	
	0.01	1.0598	1.0588	
	0.0050	1.0597	1.0596	1.0544±0.01
	0.0025	1.0596	1.0595	
$S_0 = 90$	0.01	3.2511	3.2364	
	0.01	3.2537	3.2493	
	0.0050	3.2528	3.2494	3.2273±0.01
	0.0025	3.2528	3.2495	
$S_0 = 100$	0.02	7.6012	7.5952	
	0.01	7.6020	7.5976	
	0.0050	7.6022	7.5980	7.5589±0.02
	0.0025	7.6022	7.5980	
$S_0 = 110$	0.02	14.1510	14.1524	
	0.01	14.1443	14.1425	
	0.0050	14.1420	14.1401	14.0909±0.03
	0.0025	14.1419	14.1399	
$S_0 = 120$	0.02	22.2466	22.2505	
	0.01	22.2412	22.2419	
	0.0050	22.2398	22.2402	22.1736±0.03
	0.0025	22.2394	22.2397	

(b)

$\rho_{Sr} = 0.5$	Δy	HTFD1	HTFD2	MC-LS
$S_0 = 80$	0.02	1.3551	1.3470	
	0.01	1.3576	1.3566	
	0.0050	1.3573	1.3570	1.3559±0.01
	0.0025	1.3571	1.3569	
$S_0 = 90$	0.01	3.7696	3.7606	
	0.01	3.7705	3.7688	
	0.0050	3.7694	3.7685	3.7633±0.02
	0.0025	3.7694	3.7686	
$S_0 = 100$	0.02	8.1285	8.1249	
	0.01	8.1308	8.1301	
	0.0050	8.1311	8.1308	8.1122±0.03
	0.0025	8.1312	8.1309	
$S_0 = 110$	0.02	14.4455	14.4468	
	0.01	14.4409	14.4414	
	0.0050	14.4389	14.4395	14.3884±0.03
	0.0025	14.4388	14.4394	
$S_0 = 120$	0.02	22.2859	22.2893	
	0.01	22.2815	22.2827	
	0.0050	22.2802	22.2813	
	0.0025	22.2798	22.2808	22.2039±0.04

Table 5: *Prices of American call options.* $K = 100$, $T = 0.5$, $\eta = 0.05$, $r_0 = 0.03$, $\kappa_r = 1$, $\sigma_r = 0.2$, $V_0 = 0.04$, $\theta = 0.04$, $\kappa = 2$, $\sigma = 0.4$, $\lambda = 5$, $\gamma = 0$, $\delta = 0.1$, $\rho_{SV} = -0.5$,$\rho_{Sr} = -0.5, 0.5$.

Δy	HTFD1	HTDF2
0.02	9.79	78.2
0.01	22.2	167.6
0.005	46.4	330.9
0.0025	91.2	725.4

Table 6: *Computational times (in seconds) for European Call options for* $S_0 = 100$, $\rho_{Sr} = -0.5$.

References

[1] A. Alfonsi (2010): High order discretization schemes for the CIR process: application to affine term structure and Heston models, *Mathematics of Computation*, **79**, 209-237. 18

[2] E. Appolloni, L. Caramellino, A. Zanette (2015): A robust tree method for pricing American options with CIR stochastic interest rate. *IMA Journal of Management Mathematics*, **26**, 345-375. 4, 5

[3] Bates, D.S. (1996): Jumps and stochastic volatility: exchange rate processes implicit in Deutsch mark options. *Rev Fin*, **9**, 69-107 1, 2

[4] Berman, A., Plemmons, R.J. (1994): Nonnegative matrices in the mathematical sciences, *Society for Industrial and Applied Mathematics (SIAM)*, Philadelphia, PA.

[5] Black, F., Scholes, M. (1973): The Pricing of Options and Corporate Liabilities. *Journal of Political Economy* **81**, 637-654. 1

[6] Briani, M., Caramellino, L., Zanette, A. (2015): A hybrid tree-finite difference approach for the Heston model. *IMA Journal of Managment Mathematics*, to appear. ArXiv:1307.7178. 2, 4, 9, 18

[7] Briani, M., Caramellino, L., Zanette, A. (2015): A hybrid tree/finite-difference approach for Heston-Hull-White type models. `ArXiv:1503.03705`. 2, 4, 6, 9, 10

[8] Briani, M., La Chioma, C., Natalini, R. (2004): Convergence of numerical schemes for viscosity solutions to integro-differential degenerate parabolic problems arising in financial theory. *Numer. Math.*, **98**(4), 607–646. 10

[9] Briani, M., Natalini, R., Russo, G. (2007): Implicit-Explicit Numerical Schemes for Jump-Diffusion Processes. *Calcolo*, **44**, 33-57. 10

[10] D. Brigo, F. Mercurio (2006): *Interest Rate Models-Theory and Practice*. Springer, Berlin. 3

[11] Carr, P., Madan, D. (1999): Option valuation using the Fast Fourier Transform. *Journal of Computational Finance*, **3**, 463-520. 1, 14

[12] Chiarella, C., Kang, B., Meyer, G., Ziogas, A. (2009): The evaluation of American option prices under stochastic volatility and jump-diffusion dynamics using the method of lines. *Int. J. Theor. Appl. Finan.*, **12**, 393. 2, 4, 14

[13] C. Chiarella, B.Kang, G.H. Meyer (2012): The evaluation of barrier option prices under stochastic volatility. *Computers and Mathematics with Applications*, **64**, 2034-2048. 14

[14] Cox, J.C., Ingersoll, J., Ross, S. (1985): A theory of the term structure of interest rates, *Econometrica*, **53**, 385-407.

[15] Cont, R., Voltchkova, E. (2005): A finite difference scheme for option pricing in jump-diffusion and exponential Lévy models. *SIAM Journal on Numerical Analysis*, **43**(4), 1596–1626. 10

[16] V. D'Halluin, P.A. Forsyth, G. Labahn (2005): A semi-Lagrangian Approach for American Asian options under jump-diffusion, *Siam J.Sci.Comp.* **27**, 315-345.

[17] A.L. Grzelak, C.W. Oosterlee (2011): On the Heston model with stochastic interest rates. *SIAM J. Fin. Math.* **2**, 255-286. 2

[18] S.L. Heston (1993): A Closed-Form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options, *Review of Financial Studies*, **6**, 327-3q43. 1

[19] J. Hull, A. White (1994): Numerical procedures for implementing term structure models I. *Journal of Derivatives* **2**(1), 7-16. 3

[20] Kou, S.G. (2002): A Jump-Diffusion Model for Option Pricing. *Management Science*, **48**, 1086-1101. 4

[21] Kushner, H., Dupuis, P.G. (1992): *Numerical methods for stochastic control problems in continous time.* Springer Verlag.

[22] Ikonen, S., Toivanen, J. (2009): Operator splitting methods for pricing American options under stochastic volatility, *Numer. Math.*, **113**, 299-324.

[23] Longstaff, F.A., Schwartz, E.S. (2001): Valuing American options by simulations: a simple least squares approach. *The Review of Financial Studies*, **14**, 113-148. 14, 18

[24] Merton, R. C. (1976): Option pricing when underlying stock returns are discontinuous. *J. Financial Econom.*, **3**, 125–144. 1, 2, 4

[25] Nelson, D.B., Ramaswamy, K. (1990): Simple binomial processes as diffusion approximations in financial models. *The Review of Financial Studies*, **3**, 393-430. 4, 6

[26] PREMIA: An Option Pricer. `http://www.premia.fr` 13, 14

[27] Salmi, S., Toivanen, J. (2014), IMEX schemes for pricing options under jump-diffusion models. *Applied Numerical Mathematics*, **84**, 33-45. 10

[28] Strikwerda, J.C. (2004): Finite difference schemes and partial differential equations, *Society for Industrial and Applied Mathematics (SIAM)*, Philadelphia, PA.

[29] Thomas, L.H. (1949): Elliptic Problems in Linear Differential Equations over a Network, *Watson Sci. Comput. Lab Report*, Columbia University, New York.

[30] Toivanen J. (2010), A Componentwise Splitting Method for Pricing American Options Under the Bates Model. *Applied and Numerical Partial Differential Equations, Computational Methods in Applied Sciences*, **15**, 213-227. 1

[31] E. Voltchkova (2005), Integro-differential evolution equations: numerical methods and applications in finance. *PhD thesis*, Ecole Polytechnique Paris. 10