Deterministic methods for option pricing in exponential Lévy models

Ekaterina Voltchkova and Peter Tankov

Premia 18

1 Exponential Lévy models

We consider the following model for the stock price:

$$S_t = S_0 e^{rt + X_t},$$

where X_t is a *Lévy process*. The characteristic function $\phi_t(z) = \mathbb{E}[\exp(izX_t)]$ of a Lévy process has the form $\phi_t(z) = \exp\{t\psi(z)\}$ with

$$\psi(z) = \frac{\sigma^2 z^2}{2} + i\gamma z + \int_{-\infty}^{\infty} (e^{izx} - 1 - izx1_{|x|\leq 1})\nu(\mathrm{d}x).$$

It is determined by three parameters called the *Lévy triplet* of X:

- $\sigma \geq 0$: volatility of the diffusion part
- $\gamma \in \mathbb{R}$: drift
- $\nu(dx)$: positive measure on $\mathbb{R} \setminus \{0\}$ (*Lévy measure*)

The Lévy measure has to satisfy the following integrability conditions:

$$\int_{-1}^{+1} x^2 \nu(\mathrm{d}x) < \infty, \qquad \int_{|x|>1} \nu(\mathrm{d}x) < \infty.$$

In the context of option pricing, due to the martingale condition on the discounted price $e^{-rt}S_t$, ν satisfies in addition

$$\int_{|x|>1} e^x \nu(\mathrm{d}x) < \infty,$$

and the drift is determined by the other parameters:

$$\gamma = -\frac{\sigma^2}{2} - \int_{-\infty}^{\infty} (e^x - 1 - x1_{|x|\leq 1})\nu(\mathrm{d}x).$$

One distinguishes two types of exponential Lévy models: the so-called *jump-diffusion models*, with $\sigma > 0$ and $\nu(\mathbb{R}) < \infty$, and *pure jump models*, where $\sigma = 0$ and $\nu(\mathbb{R}) = \infty$. We give below examples of such models used in financial literature.

1.1 Jump-diffusion models

Merton model
The Lévy measure in this model has a gaussian density:

$$\nu(x) = \lambda \frac{e^{-(x-\mu)^2/2\delta^2}}{\sqrt{2\pi\delta^2}}.$$

Here λ is the jump intensity, μ the average jump size, and δ the standard variation of jump sizes.

Double exponential model (Kou)
In this model, the jumps have an asymmetric exponential distribution:

$$\nu_0(x) = p\lambda\lambda_+ e^{-\lambda_+ x} 1_{x>0} + (1-p)\lambda\lambda_- e^{-\lambda_- |x|} 1_{x<0}.$$

Here λ is the jump intensity, parameters $\lambda_- > 0$ and $\lambda_+ > 1$ control the decrease of the distribution tails of, respectively, negative and positive jumps, and p is the probability of a positive jump.

1.2 Pure jump models

Variance Gamma model
The Lévy measure of a Variance Gamma process X_t has a density given by:

$$\nu(x) = \frac{1}{\kappa|x|} e^{Ax - B|x|} \qquad \text{with} \quad A = \frac{\theta}{\sigma^2} \quad \text{and} \quad B = \frac{\sqrt{\theta^2 + 2\sigma^2/\kappa}}{\sigma^2}.$$

The characteristic function of X_t is equal to

$$\phi_t(u) = e^{itu\gamma}(1 + \frac{u^2\sigma^2\kappa}{2} - i\theta\kappa u)^{-\frac{t}{\kappa}}, \qquad \text{with} \quad \gamma = \frac{1}{\kappa}\log(1 - \frac{\sigma^2\kappa}{2} - \theta\kappa).$$

Normal inverse gaussian models (NIG)
The Lévy density in this model is given by

$$\nu(x) = \frac{C}{|x|} e^{Ax} K_1(B|x|)$$

with

$$C = \frac{\sqrt{\theta^2 + \sigma^2/\kappa}}{\pi\sigma\sqrt{\kappa}}, \qquad A = \frac{\theta}{\sigma^2}, \qquad \text{and} \quad B = \frac{\sqrt{\theta^2 + \sigma^2/\kappa}}{\sigma^2},$$

where K_1 is the modified Bessel function of second kind. Note that the asymptotic behaviour of K_1 in zero implies that

$$\nu(x) \sim \frac{1}{|x|^2} \qquad \text{as} \quad x \to 0.$$

Let us make a computational remark: if Ax is large, the exponential in the expression of ν may lead to an overflow. To avoid this, we use the asymptotic behaviour of K_1 for large arguments:

$$K_1(x) \approx \frac{\pi}{\sqrt{2\pi x}} e^{-x} \qquad \text{as} \quad x >> 1.$$

We then obtain the following approximation[1] of the Lévy density:

$$\nu(x) \approx \frac{C}{|x|} \frac{\pi}{\sqrt{2\pi B|x|}} e^{Ax - B|x|}.$$

Tempered stable models
These models are also known as CGMY or KoBoL. The Lévy density of a tempered stable process has the following expression:

$$\nu(x) = \frac{c_-}{|x|^{1+\alpha_-}} e^{-\lambda_- |x|} 1_{x<0} + \frac{c_+}{|x|^{1+\alpha_+}} e^{-\lambda_+ x} 1_{x>0},$$

with $c_\pm > 0$, $\lambda_- > 0$, $\lambda_+ > 1$, and $0 < \alpha_\pm < 2$.

A detailed presentation of exponential Lévy models and their properties can be found in [1].

2 Standard European options

We consider a European option with maturity T and payoff $H(S_T)$. If $C_t = C(t, S_t)$ is its price at time t, let us introduce the following notations:

- $\tau = T - t$ (time to maturity),
- $x = \log(S/S_0)$ (logarithmic return),
- $h(x) = H(S_0 e^x)$ (payoff function after change of variables),
- $u(\tau, x) = e^{r\tau} C(T - \tau, S_0 e^x)$ (option's forward price in new variables).

The function $u(\tau, x)$ may be characterized as the solution of the following partial integro-differential equation (PIDE)[2]:

$$\frac{\partial u}{\partial \tau}(\tau, x) = Lu(\tau, x), \qquad \text{on } (0, T] \times \mathbb{R}, \tag{1}$$

$$u(0, x) = h(x), \qquad \text{on } \mathbb{R}, \tag{2}$$

[1] In practice, we use it when $Ax > 600$.

[2] If the option price u is not a smooth function, which is possible in the pure jump case ($\sigma = 0$), then it must be considered as a weak, *viscosity* solution to the PIDE (see [2, 3]).

where L is an integro-differential operator:

$$Lf(x) = \frac{\sigma^2}{2}\frac{\partial^2 f}{\partial x^2}(x) - \left(\frac{\sigma^2}{2} - r\right)\frac{\partial f}{\partial x}(x) + \int_{-\infty}^{\infty}[f(x+y) - f(x) - (e^y - 1)\frac{\partial f}{\partial x}(x)]\nu(dy). \quad (3)$$

Equation (1)–(2) plays the same role in exponential Lévy models as the Black-Scholes equation in the Black-Scholes model. In fact, the BS model is a particular case of exponential Lévy models: we obtain it if $\nu \equiv 0$.

3 Barrier "out" options

We also consider *up-and-out*, *down-and-out*, and *double barrier* options. We have, respectively, an upper barrier $U > S_0$, a lower barrier $L < S_0$, or both of them. If the stock price S_t has not cross the barriers before maturity T, then the payoff of the option is $H(S_T)$; otherwise, the payoff is zero or equal to a rebate $G(\tau^*, S_{\tau^*})$ where τ^* is the first exit time.

As in the European vanilla case, the price $C_t = C_b(t, S_t)1_{t<\tau^*}$ of such a barrier option can be found by solving the same integro-differential equation with additional boundary conditions.[3] Let us denote:

- $l = \log(L/S_0)$,
- $u = \log(U/S_0)$,
- $g(\tau, x) = e^{r\tau}G(T - \tau, S_0e^x)$ (rebate after the change of variables),
- $v(\tau, x) = e^{r\tau}C_b(T - \tau, S_0e^x)$ (forward price).

Then,

$$\frac{\partial v}{\partial \tau}(\tau, x) = Lv(\tau, x), \qquad (\tau, x) \in (0, T] \times (l, u), \quad (4)$$

$$v(0, x) = h(x), \qquad x \in (l, u), \quad (5)$$

$$v(\tau, x) = g(\tau, x), \qquad \tau \in [0, T], \quad x \notin (l, u). \quad (6)$$

We mean that $l = -\infty$ if there is no lower barrier and $u = \infty$ if there is no upper barrier. So, (4)–(6) covers all types of barrier options above, as well as the European vanilla case.

Note, that (6) is different from usual boundary conditions for differential equations: it gives the values of the solution not only *at* the barriers but also *beyond* the barriers. It is an important consequence of the *non-local* character of the operator L.

[3] It is proven rigorously only in the case of a Lipschitz payoff H and some futher conditions on the Lévy triplet [3, 2]. Nevertheless, the relationship between option prices and PIDEs, relying on some empirical arguments, is also used in more general situations.

4 Numerical solution of the integro-differential equation

To solve numerically the problem (4)–(6), we do the following steps:

- *Approximation of small jumps in the pure jump case.* In the pure jump case, the Lévy measure ν has a non-integrable singularity at the origin. In other terms, there are too many small jumps: their sum is infinite. To avoid this singularity, we replace small jumps by a suitable brownian motion.
- *Truncation of large jumps.* This corresponds to truncate the integration domain in (3).
- *Localization.* It means that we chose a bounded computational domain if the problem was initially stated on an unbounded interval (as in the European or one-barrier cases). This implies that we have to chose some artificial boundary conditions.
- *Discretization.* The derivatives of the solution are replaced by usual finite differences and the integral terms are approximated using the trapezoidal rule. The problem is then solved using an explicit-implicit scheme.

Let us now consider these steps in more details. A complete presentation with proofs can be found in [2, 4].

4.1 Approximation of small jumps

In the case $\nu(\mathbb{R}) = \infty$, the small jumps of a Lévy process generate a behavior similar to that of a brownian motion. This led to the idea to replace the jumps smaller than some $\varepsilon > 0$ by a Wiener process with the same variance:

$$\sigma^2(\varepsilon) = \int_{-\varepsilon}^{\varepsilon} x^2 \nu(\mathrm{d}x). \tag{7}$$

Remark that in models with jumps of infinite activity ($\nu(\mathbb{R}) = \infty$) we have $\sigma = 0$. By approximating the small jumps, we get a non-zero diffusion component $\sigma^2(\varepsilon)$ which has a regularizing effect on the solution. It makes the numerical solution easier.

We obtain the following PIDE:

$$\frac{\partial u}{\partial \tau}(\tau, x) = \frac{\sigma^2(\varepsilon)}{2} \frac{\partial^2 u}{\partial x^2}(\tau, x) - \left(\frac{\sigma^2(\varepsilon)}{2} - r \right) \frac{\partial u}{\partial x}(\tau, x) + \int_{|x|>\varepsilon} [u(\tau, x+y) - u(\tau, x) - (e^y - 1) \frac{\partial u}{\partial x}(\tau, x)] \nu(dy). \tag{8}$$

The integral in (8) is no more singular. In [2, 4], it is proven that the solution $u^\varepsilon(\tau, x)$ of this approximated equation converges to the solution $u(\tau, x)$ of the original problem as $\varepsilon \to 0$.

Clearly, if the model is of jump-diffusion type (e.g. Merton or Kou), this approximation is not needed.

4.2 Truncation of large jumps

We cannot calculate numerically an integral on the infinite range $(-\infty, \infty)$, so we have to truncate this domain to a bounded interval (B_l, B_r). 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. Therefore, we don't change much the solution by truncating the tails of ν. The rigorous proof of the validity of such approximation is given in [2, 4].

The PIDE becomes

$$\frac{\partial u}{\partial \tau}(\tau, x) = \frac{\sigma^2}{2}\frac{\partial^2 u}{\partial x^2}(\tau, x) + \left(r - \frac{\sigma^2}{2}\right)\frac{\partial u}{\partial x}(\tau, x) + \int_{B_l}^{B_r} [u(\tau, x+y) - u(\tau, x) - (e^y - 1)\frac{\partial u}{\partial x}(\tau, x)]\nu(dy). \quad (9)$$

In practice, we fix some level of tolerance (e.g. 10^{-5}) and truncate the values of ν which are smaller than this level ($\nu(x) < 10^{-5}$).

4.3 Localization

Similarly, for the computational purposes, the domain of definition of the equation has to be bounded. For barrier options, the barriers are the natural limits for this domain and the rebate is the natural boundary condition. In absence of barriers, we have to choose artificial bounds $(-A_l, A_r)$ and impose artificial boundary conditions. Recall that "boundary" conditions in this case mean the extension of the solution beyond the bounds: $u(\tau, x) = g(\tau, x)$ for all $x \notin (-A_l, A_r)$, $\tau \in [0, T]$.

In [2], it is shown that a good choice for the boundary conditions is $g(\tau, x) = h(x + r\tau)$ where h is the payoff function. For example, for a put option, we have $h(x) = (K - S_0 e^x)^+$ and thus $g(\tau, x) = (K - S_0 e^{x+r\tau})^+$.

In the case of one barrier, we need this boundary condition only on one side of the domain: the other is zero or given by the rebate. For example, for an up-and-out option without rebate, we obtain the following integro-differential problem:

$$\begin{aligned}
&\frac{\partial v}{\partial \tau}(\tau, x) = Lv(\tau, x), && (\tau, x) \in (0, T] \times (-A_l, u),\\
&v(0, x) = h(x), && x \in (-A_l, u),\\
&v(\tau, x) = 0, && \tau \in [0, T], \quad x \geq u,\\
&v(\tau, x) = h(x + r\tau), && \tau \in [0, T], \quad x \leq -A_l.
\end{aligned}$$

Practically, to chose the bounds A_l and A_r we proceed as follows. We first fix the domain of interest where we want to obtain a good precision on the solution:

$I = [\log(2/3), \log(2)]$ (this range corresponds to the values of moneyness between 0.5 and 1.5). Then, we chose the computational domain in such a way that the process $rt + X_t$ has a small probability to exit this domain starting from I. A heuristic technique for doing this consists in choosing $(-A_l, A_r)$ depending on the average and the variance of $rt + X_t$. We presume that, starting on the point x, the process will essentially stay in

$$\bigcup_{t\in[0,T]} (x + rt + t\mathbb{E}X_1 - kt\sqrt{VarX_1}, x + rt + t\mathbb{E}X_1 + kt\sqrt{VarX_1}) \subseteq (x + T\mathbb{E}X_1 - kT\sqrt{VarX_1}, x + rT + kT\sqrt{VarX_1}),$$

for some $k > 0$.[4] Taking into account all points of I, we then put

$$(A_l, A_r) = (\log(2/3) + T\mathbb{E}X_1 - kT\sqrt{VarX_1}, \log(2) + rT + kT\sqrt{VarX_1}). \quad (10)$$

Numerical tests showed that $k = 3$ is sufficient. The moments of X_1 can be easily calculated in all models mentioned in Section 1 (see Table 1).

Table 1: The average and the variance of X_1 in some exponential Lévy models.

model	$\mathbb{E}X_1$	$VarX_1$
Merton	$-\sigma^2/2 - \lambda(\exp(\mu + \delta^2/2) - 1 - \mu)$	$\sigma^2 + \lambda(\delta^2 + \mu^2)$
Kou	$-\sigma^2/2 - \lambda\left(\frac{p}{\lambda_+(\lambda_+-1)} + \frac{1-p}{\lambda_-(\lambda_-+1)}\right)$	$\sigma^2 + \lambda\left(\frac{p}{\lambda_+^2} + \frac{1-p}{\lambda_-^2}\right)$
VG	$\theta + \log(1 - \sigma^2\kappa/2 - \theta\kappa)/\kappa$	$\sigma^2 + \theta^2\kappa$
NIG	$\theta + (\sqrt{1 - \sigma^2\kappa - 2\theta\kappa} - 1)/\kappa$	$\sigma^2 + \theta^2\kappa$
Temp. Stable	$-E_+ - E_-$, see (11)–(12) below	$\frac{c_+\Gamma(2-\alpha_+)}{\lambda_+^{2-\alpha_+}} + \frac{c_-\Gamma(2-\alpha_-)}{\lambda_-^{2-\alpha_-}}$

$$E_+ = \begin{cases} \Gamma(-\alpha_+)\lambda_+^{\alpha_+} c_+ \left(\left(1 - \frac{1}{\lambda_+}\right)^{\alpha_+} - 1 + \frac{\alpha_+}{\lambda_+}\right), & \text{si } \alpha_+ \neq 1, \\ c_+\left[(\lambda_+ - 1)\log\left(1 - \frac{1}{\lambda_+}\right) + 1\right], & \text{si } \alpha_+ = 1, \end{cases} \quad (11)$$

$$E_- = \begin{cases} \Gamma(-\alpha_-)\lambda_-^{\alpha_-} c_- \left(\left(1 + \frac{1}{\lambda_-}\right)^{\alpha_-} - 1 - \frac{\alpha_-}{\lambda_-}\right), & \text{si } \alpha_- \neq 1, \\ c_-\left[(\lambda_- + 1)\log\left(1 + \frac{1}{\lambda_-}\right) - 1\right], & \text{si } \alpha_- = 1. \end{cases} \quad (12)$$

4.4 Discretization

We can always assume that $\sigma > 0$ and $\nu(\mathbb{R}) = \lambda < +\infty$, possibly after the approximation of small jumps (Section 4.1). In this case, we can separate the

[4] To obtain the second line, we have used the fact that $\mathbb{E}X_1 \leq 0$ due to the martingale condition.

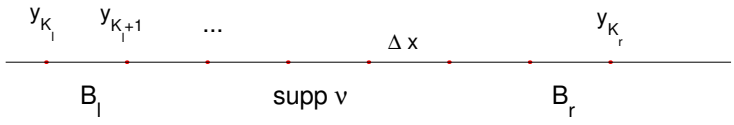

Figure 1: The support of ν is discretized with the same step Δx as the interval $[-A_l, A_r]$.

three terms under the integral in (3) and write the operator in this form:[5]

$$Lu = \frac{\sigma^2}{2}\frac{\partial^2 u}{\partial x^2} - \left(\frac{\sigma^2}{2} - r\right)\frac{\partial u}{\partial x} + \int_{B_l}^{B_r} \nu(dy)u(\tau, x+y) - \lambda u - \alpha\frac{\partial u}{\partial x}, \qquad (13)$$

where $\alpha = \int_{B_l}^{B_r}(e^y - 1)\nu(\mathrm{d}y)$. We consider the localized problem on $(-A_l, A_r)$, $A_l, A_r > 0$:

$$\frac{\partial u}{\partial \tau} = Lu, \qquad \text{on} \quad (0,T]\times(-A_l, A_r) \qquad (14)$$
$$u(0,x) = h(x), \qquad x \in (-A_l, A_r), \qquad (15)$$
$$u(\tau,x) = g(\tau,x), \qquad x \notin (-A_l, A_r). \qquad (16)$$

Let us introduce a uniform grid on $[0,T]\times\mathbb{R}$:

$$\tau_n = n\Delta t, \quad n = 0\ldots M, \qquad x_i = -A_l + i\Delta x, \quad i \in \mathbb{Z},$$

with $\Delta t = T/M$, $\Delta x = (A_r + A_l)/N$. The values of u on this grid are denoted by $\{u_i^n\}$.

To approximate the integral term, we use the trapezoidal rule with the same discretization step Δx. Choose K_l, $K_r \in \mathbb{Z}$ such that $[B_l, B_r] \subset [(K_l - 1/2)\Delta x, (K_r + 1/2)\Delta x]$ (Fig. 1). Then,

$$\int_{B_l}^{B_r} \nu(dy)u(\tau, x_i + y) \approx \sum_{j=K_l}^{K_r} \nu_j u_{i+j}, \quad \text{where} \quad \nu_j = \int_{(j-1/2)\Delta x}^{(j+1/2)\Delta x} \nu(\mathrm{d}y). \qquad (17)$$

In all models considered above, ν_j, λ, and α can be calculated analytically or numerically with very high precision.[6] However, if the exact calculation is not possible, we can use the following approximations:

$$\nu_j \approx \frac{\nu(y_{j-1/2}) + \nu(y_{j+1/2})}{2}\Delta x, \quad \lambda \approx \hat\lambda = \sum_{j=K_l}^{K_r}\nu_j, \quad \alpha \approx \hat\alpha = \sum_{j=K_l}^{K_r}(e^{y_j} - 1)\nu_j,$$

where $y_j = j\Delta x$.

[5] If $\nu(\mathbb{R}) = \infty$, each term diverges if taken separately while their combination is integrable.

[6] For example, in tempered stable models, ν_j can be expressed using the incomplete gamma function which can be efficiently evaluated by standard routines.

The space derivatives of u are approximated by finite differences:

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_i \approx \frac{u_{i+1} - 2u_i + u_{i-1}}{(\Delta x)^2}, \tag{18}$$

$$\left(\frac{\partial u}{\partial x}\right)_i \approx \frac{u_{i+1} - u_i}{\Delta x}, \quad \text{or} \quad \left(\frac{\partial u}{\partial x}\right)_i \approx \frac{u_i - u_{i-1}}{\Delta x}. \tag{19}$$

The choice of the approximation of the first order derivative — forward or backward difference — depends on the parameters σ, r, and α (see below).

Using (17)–(19) we obtain an approximation for $Lu \approx D_\Delta u + J_\Delta u$, where $D_\Delta u$ and $J_\Delta u$ are chosen in the following way:

Explicit-Implicit Scheme. Without loss of generality, suppose that $\sigma^2/2 - r < 0$. Then

$$(D_\Delta u)_i = \frac{\sigma^2}{2}\frac{u_{i+1} - 2u_i + u_{i-1}}{(\Delta x)^2} - \left(\frac{\sigma^2}{2} - r\right)\frac{u_{i+1} - u_i}{\Delta x}. \tag{20}$$

If $\sigma^2/2 - r > 0$, we change the discretization of $\partial u/\partial x$ by choosing the backward difference instead of the forward one. It is necessary for the stability of the algorithm.

Similarly, if $\alpha < 0$ we discretize J as follows:

$$(J_\Delta u)_i = \sum_{j=K_l}^{K_r} \nu_j u_{i+j} - \lambda u_i - \alpha\frac{u_{i+1} - u_i}{\Delta x}. \tag{21}$$

Otherwise, we change the approximation of the first derivative.

Finally, we replace the problem (14)–(16) by the following *explicit-implicit scheme*:

Initialization :

$$u_i^0 = h(x_i), \quad \text{if } i \in \{0, \dots, N-1\}, \tag{22}$$

$$u_i^0 = g(0, x_i), \quad \text{otherwise.} \tag{23}$$

For $\mathbf{n = 0, \dots, M - 1}$:

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = (D_\Delta u^{n+1})_i + (J_\Delta u^n)_i, \quad \text{if } i \in \{0, \dots, N-1\} \tag{24}$$

$$u_i^{n+1} = g(n\Delta t, x_i), \quad \text{otherwise.} \tag{25}$$

Here the non-local operator J is treated explicitly to avoid the inversion of the dense matrix J_Δ, while the differential part D is treated implicitly. At each time iteration, we first evaluate vector $J_\Delta u^n$ where u^n is known, and then solve the tridiagonal system (24) for u^{n+1} (see next two sections for details).

4.5 Computation of the integral using FFT

At each time iteration, we have to calculate the sum

$$(Cu)_i := \sum_{j=K_l}^{K_r} \nu_j u_{i+j}, \qquad i = 0, \ldots, N-1 \tag{26}$$

which is the most computationally expensive step of the algorithm when applied directly. Indeed, it requires $O(N^2)$ operations. Fortunately, the particular form of this sum allows to reduce considerably the computational time by using the Fast Fourier Transform. First of all, recall which quantities may be calculated efficiently by the FFT.

The *discrete Fourier transform* of $x = (x_1, \ldots, x_n)$ is given by the following vector:

$$\langle x \rangle_k \quad = \quad \sum_{j=1}^{n} x_j \omega_n^{(j-1)(k-1)}, \qquad k = 1, \ldots, n,$$

where $\omega_n = \exp(-2\pi i/n)$. Conversely, given $\langle x \rangle$, we can reconstruct x by the *inverse discrete Fourier transform*:

$$x_j \quad = \quad \langle\langle x \rangle\rangle_j^{-1} = \frac{1}{n} \sum_{k=1}^{n} \langle x \rangle_k \omega_n^{-(j-1)(k-1)}, \qquad j = 1, \ldots, n.$$

To calculate (Cu), we will use the discrete analogue of the convolution theorem. For $y = (y_1, \ldots, y_n)$, let us define

$$c_j = \sum_{k=1}^{n} x_k y_{j+1-k}, \qquad j = 1, \ldots, n, \tag{27}$$

where indices of y are taken modulo n: for example, $y_0 \equiv y_n$, $y_{-1} \equiv y_{n-1}$ and so on. Vector c represents the *discrete convolution* of x and y. The discrete convolution theorem states the following:

$$\langle c \rangle_k = \langle x \rangle_k \langle y \rangle_k, \qquad k = 1, \ldots, n$$

which implies

$$c_j = \sum_{k=1}^{n} x_k y_{j+1-k} = \langle\langle x \rangle \langle y \rangle\rangle_j^{-1}, \qquad j = 1, \ldots, n. \tag{28}$$

The FFT algorithm allows to calculate simultaneously all values of c_j, $j = 1, \ldots, n$, by $O(n \ln(n))$ operations. This is true for all n, even prime, and not only for $n = 2^p$.

Let us now explain how to apply the property (28) to the calculation of the sum (26). We define $\tilde{N} = N + K_r - K_l$ and construct two vectors of size $\tilde{N}$:

$$\begin{aligned} \mu \quad &= \quad (\underbrace{\nu_{K_r}, \ldots, \nu_{K_l}}_{K_r - K_l + 1}, \underbrace{0, \ldots, 0}_{N-1}), \\ v \quad &= \quad (\underbrace{u_{K_r+1}, \ldots, u_{K_r+N-1}}_{N-1}, \underbrace{u_{K_l}, \ldots, u_{K_r}}_{K_r - K_l + 1}). \end{aligned}$$

Vector μ is composed of ν_k's ranged in the inverse order and it is completed with zeros. Vector v contains all values of u (from u_{K_l} to u_{N-1+K_r}) used in the calculation of $(Cu)_i$, for $i = 0, \ldots, N-1$, but the first $K_r - K_l + 1$ values are placed on the back of the vector. One can check that

$$(Cu)_i = \sum_{k=1}^{\tilde{N}} \mu_k v_{i+1-k}, \qquad i = 0, \ldots, N-1, \tag{29}$$

where v is interpreted as a periodic vector of period $\tilde{N}$. The equality (29) can also be written in a vector form:

$$\langle\langle\mu\rangle\langle v\rangle\rangle^{-1} = ((Cu)_1, \ldots, (Cu)_{N-1}, \underbrace{*, \ldots, *}_{K_r - K_l}, (Cu)_0).$$

In practice, we calculate $\langle\mu\rangle$ once and then find $(Cu^n)_i$, $i = 0, \ldots, N-1$, at each iteration by computing one discrete Fourier transform for $\langle v\rangle$ and one inverse transform for $\langle\langle\mu\rangle\langle v\rangle\rangle^{-1}$ (values marked by $*$ are an artefact of the computation: we don't use them). FFT algorithm realizes this procedure with $O(N\ln(N))$ operations, which is much better than $O(N^2)$.

As we noted, there exist FFT algorithms for arbitrary N and not only for powers of 2. However, if a good implementation of such algorithms is not available, it is possible to use standard FFT routines designed for $N = 2^p$ by completing μ and v with zeros. More precisely, we define N_z such that $\tilde{N} + N_z = 2^p$, for some $p \in \mathbb{N}$, and put

$$\begin{aligned} \mu &= (\underbrace{\nu_{K_r}, \ldots, \nu_{K_l}}_{K_r - K_l + 1}, \underbrace{0, \ldots, 0}_{N_z}, \underbrace{0, \ldots, 0}_{N-1}), \\ v &= (\underbrace{u_{K_r+1}, \ldots, u_{K_r+N-1}}_{N-1}, \underbrace{0, \ldots, 0}_{N_z}, \underbrace{u_{K_l}, \ldots, u_{K_r}}_{K_r - K_l + 1}). \end{aligned}$$

The values of $(J_\Delta u)_i$ are then obtained in the same way:

$$\langle\langle\mu\rangle\langle v\rangle\rangle^{-1} = ((Cu)_1, \ldots, (Cu)_{N-1}, \underbrace{*, \ldots, *}_{K_r - K_l + N_z}, (Cu)_0).$$

4.6 Implicit part: solving the tridiagonal system

Without loss of generality, suppose as previously that $\sigma^2/2 - r < 0$ and $\alpha < 0$. One can rewrite (24) as follows:

$$(1 + a\Delta t)u_0^{n+1} - b\Delta t u_1^{n+1} = v_0^n + c\Delta t u_{-1}^{n+1}, \tag{30}$$

$$-c\Delta t u_{i-1}^{n+1} + (1 + a\Delta t)u_i^{n+1} - b\Delta t u_{i+1}^{n+1} = v_i^n, \qquad i = 1, \ldots, N-2, \tag{31}$$

$$-c\Delta t u_{N-2}^{n+1} + (1 + a\Delta t)u_{N-1}^{n+1} = v_i^n + b\Delta t u_N^{n+1}, \tag{32}$$

where

$$\begin{aligned} a &= \frac{\sigma^2}{(\Delta x)^2} - \left(\frac{\sigma^2}{2} - r\right)\frac{1}{\Delta x}, \\ b &= \frac{\sigma^2}{2(\Delta x)^2} - \left(\frac{\sigma^2}{2} - r\right)\frac{1}{\Delta x}, \\ c &= \frac{\sigma^2}{2(\Delta x)^2}, \\ v_i^n &= u_i^n + \Delta t\left\{(Cu^n)_i - \lambda u_i^n - \alpha\frac{u_{i+1}^n - u_i^n}{\Delta x}\right\}. \end{aligned}$$

Values of u_{-1}^{n+1}, u_N^{n+1}, and u_N^n are determined by the boundary conditions. The tridiagonal system of equations (30)–(32) on $u^{n+1} = (u_0^{n+1}, \ldots, u_{N-1}^{n+1})$ is solved by the LU-decomposition method in $O(N)$ operations.

4.7 Properties of the scheme

For discrete functions on the grid, we define the uniform norm:

$$\|u\|_{\Delta,\{0,\ldots,M\}} := \sup_{n=0,\ldots,M,\ i\in\mathbb{Z}} |u_i^n|. \tag{33}$$

If $\Delta t \le \Delta x/(|\alpha| + \lambda\Delta x)$, the following properties are satisfied:

Consistency: the scheme is locally consistent with the PIDE with order $O(\Delta t + \Delta x)$.

Stability: the scheme is stable in the uniform norm (33).

Monotonicity: the scheme satisfies the comparison principle: let $h^1 \le h^2$ and $g^1 \le g^2$, and let u_Δ^1 and u_Δ^2 be the solutions of the scheme corresponding respectively to the initial conditions h^1 and h^2 and boundary conditions g^1 and g^2; then $u_\Delta^1 \le u_\Delta^2$.

Convergence: the solution of the scheme converges in the uniform norm (33) to the viscosity solution of the PIDE.

Rate of convergence: in the European vanilla case, we have the following estimate on the rate of convergence: if $\Delta t \sim \Delta x^2$, then

$$\|u - u_\Delta\|_{\Delta,\{0,\ldots,M\}} \le C\Delta x.$$

Complexity: $O(N\ln(N))$ operations.

For the proofs, see [2, 4].

4.8 Choice of ε

As we noted, in the case of pure-jump models with jumps of infinite intensity, we first approximate the small jumps and then solve the approximated problem by finite-difference scheme. Therefore, we have two sources of error:

$$\|u - u_\Delta^\varepsilon\| \leq \|u - u^\varepsilon\| + \|u^\varepsilon - u_\Delta^\varepsilon\|.$$

In the European vanilla case, we have shown that the approximation error is proportional to ε, and the discretization error is of order Δx but with a constant depending on ε:

$$\|u - u_\Delta^\varepsilon\| \leq C_1\varepsilon + C_2(\varepsilon)\Delta x. \tag{34}$$

The constant $C_2(\varepsilon)$ typically blows up as $\varepsilon \to 0$, so it is not a good idea to chose ε as small as possible. There exists an optimal $\varepsilon(\Delta x)$ which minimizes the global error (34).

In [2], we search $\varepsilon(\Delta x)$ in the form $C(\Delta x)^a$ and give the following estimate on the parameter a:

$$a = \begin{cases} \frac{2}{4-\alpha}, & 0 \leq \alpha \leq 2/3 \\ \frac{1}{1+\alpha}, & 2/3 < \alpha < 2, \end{cases} \tag{35}$$

where α describes the degree of singularity of ν at the origin: $\nu(x) \sim 1/|x|^{1+\alpha}$. For instance, in Variance Gamma model $\alpha = 0$, in NIG model $\alpha = 1$, and in Tempered Stable models $\alpha \in (0, 2)$.

We don't know if the estimate (35) is optimal but, in absence of better choice, we use it in our calculations. In TS models with $\alpha_- \neq \alpha_+$, we approximate the small jumps also in an asymmetric way: the negative jumps with parameter ε_- and the positive ones with ε_+.

4.9 Centered version of the scheme

In the above scheme, we use one-sided (forward or backward) differences to approximate the first order derivative. We can also try to use the centered approximation to achieve greater precision. The problem with this scheme is that it may be unstable: there is now condition on Δt and Δx which guarantee the stability. However, in some cases this scheme works and then gives better results. We implemented it for comparison.

We first check the condition

$$\Delta x \leq \frac{\sigma^2}{|\sigma^2/2 - r|}$$

which is the stability condition for the implicit part. If it is not satisfied, the scheme will blow up for sure, so we *must* use a one-sided approximation. Otherwise, we use

$$\left(\frac{\partial u}{\partial x}\right)_i \approx \frac{u_{i+1} - u_{i-1}}{2\Delta x}. \tag{36}$$

For the explicit part, we also use the centered approximation (36) but we cannot guarantee that the scheme will be stable. Roughly speaking, the stability of the scheme depends on the smoothness of the solution. In particular, it is probably not a good idea to use this scheme for barrier options because of the irregularities at the barriers.

Here are the discretized operators for the centered scheme:

$$\begin{aligned}(D_\Delta u)_i &= \frac{\sigma^2}{2}\frac{u_{i+1}-2u_i+u_{i-1}}{(\Delta x)^2} - \left(\frac{\sigma^2}{2}-r\right)\frac{u_{i+1}-u_{i-1}}{2\Delta x},\\ (J_\Delta u)_i &= \sum_{j=K_l}^{K_r}\nu_j u_{i+j} - \lambda u_i - \alpha\frac{u_{i+1}-u_{i-1}}{2\Delta x}.\end{aligned}$$

The scheme is obtained by substituting these expressions into (24). It is then treated exactly in the same way as explained in previous sections.

5 Pricing European options via Fourier transform

Another deterministic approach to pricing European options in exponential Lévy models was proposed by Carr and Madan [5]. They use Fourier transform and, in particular, the Fast Fourier transform algorithm. We present here a slightly improved version of their method proposed in [6, 1].

Let $\{X_t\}_{t\geq 0}$ be a Lévy process. To compute the price of a call option

$$C(k) = e^{-rT}E[(e^{rT+X_T}-e^k)^+],$$

we would like to express its Fourier transform in log strike in terms of the characteristic function $\Phi_T(v)$ of X_T and then find the prices for a range of strikes by Fourier inversion. However we cannot do this directly because $C(k)$ is not integrable (it tends to 1 as k goes to $-\infty$). The key idea is to instead compute the Fourier transform of the (modified) time value of the option, that is, the function

$$z_T(k) = e^{-rT}E[(e^{rT+X_T}-e^k)^+] - (1-e^{k-rT})^+. \tag{37}$$

Proposition 1 (Carr and Madan [5]). *Let $\{X_t\}_{t\geq 0}$ be a real-valued Lévy process satisfying the martingale condition, such that*

$$E[e^{(1+\alpha)X_t}] < \infty \tag{38}$$

for some $\alpha > 0$. Then the Fourier transform in log-strike k of the time value of a call option is given by:

$$\zeta_T(v) := \int_{-\infty}^{+\infty} e^{ivk} z_T(k)dk = e^{ivrT}\frac{\Phi_T(v-i)-1}{iv(1+iv)}. \tag{39}$$

Remark 1. Since typically $\Phi_T(z) \to 0$ as $\Re z \to \infty$, $\zeta_T(v)$ will behave like $|v|^{-2}$ at infinity which means that the truncation error in the numerical evaluation of the inverse Fourier transform will be large. The reason of such a slow convergence is that the time value (37) is not smooth; therefore its Fourier transform does not decay sufficiently fast at infinity. For most models the convergence can be improved by replacing the time value with a smooth function of strike. Instead of subtracting the (non-differentiable) intrinsic value of the option from its price, we suggest to subtract the Black-Scholes call price with a non-zero volatility (which is a smooth function). The resulting function will be both integrable and smooth. Suppose that hypothesis (38) is satisfied and denote

$$\tilde{z}_T(k) = e^{-rT} E[(e^{rT+X_T} - e^k)^+] - C_{BS}^{\Sigma}(k),$$

where $C_{BS}^{\Sigma}(k)$ is the Black-Scholes price of a call option with volatility Σ and log-strike k for the same underlying value and the same interest rate. Proposition 1 then implies that the Fourier transform of $\tilde{z}_T(k)$, denoted by $\tilde{\zeta}_T(v)$, satisfies

$$\tilde{\zeta}_T(v) = e^{ivrT} \frac{\Phi_T(v-i) - \Phi_T^{\Sigma}(v-i)}{iv(1+iv)}, \tag{40}$$

where $\Phi_T^{\Sigma}(v) = \exp(-\frac{\Sigma^2 T}{2}(v^2 + iv))$. Since for most exp-Lévy models found in the literature (except Variance Gamma) the characteristic function decays faster than every power of its argument at infinity, this means that the expression (40) will also decay faster than every power of v as $\Re v \to \infty$, and the truncation error in the numerical evaluation of the inverse Fourier transform will be very small for *every* $\Sigma > 0$.[7]

Numerical Fourier inversion. Option prices can be computed by evaluating numerically the inverse Fourier transform of $\tilde{\zeta}_T$:

$$\tilde{z}_T(k) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} e^{-ivk} \tilde{\zeta}_T(v) dv. \tag{41}$$

This integral can be efficiently computed for a range of strikes using the Fast Fourier Transform. Recall that this algorithm allows to calculate the discrete Fourier transform $\mathrm{DFT}[f]_{n=0}^{N-1}$, defined by,

$$\mathrm{DFT}[f]_n := \sum_{k=0}^{N-1} f_k e^{-2\pi i n k/N}, \quad n = 0 \dots N-1, \tag{42}$$

using only $O(N \log N)$ operations.

[7] The convergence of $\tilde{\zeta}_T$ to zero is faster than exponential for all values of Σ and it is particularly good for the value of Σ for which $\tilde{\zeta}(0) = 0$.

To approximate option prices, we truncate and discretize the integral (41) as follows:

$$\frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-ivk}\tilde{\zeta}_T(v)dv = \frac{1}{2\pi}\int_{-L/2}^{L/2} e^{-ivk}\tilde{\zeta}_T(v)dv + \varepsilon_T$$
$$= \frac{L}{2\pi(N-1)}\sum_{m=0}^{N-1} w_m\tilde{\zeta}_T(v_m)e^{-ikv_m} + \varepsilon_T + \varepsilon_D, \quad (43)$$

where ε_T is the truncation error, ε_D is the discretization error, $v_m = -L/2+m\Delta$, $\Delta = L/(N-1)$ is the discretization step and w_m are weights, corresponding to the chosen integration rule (for instance, for the Simpson's rule $w_0 = 1/3$, and for $k = 1, \ldots, N/2$, $w_{2k-1} = 4/3$ and $w_{2k} = 2/3$).[8] Now, choosing $k_n = k_0 + \frac{2\pi n}{N\Delta}$ we see that the sum in the last term becomes a discrete Fourier transform:

$$\frac{L}{2\pi(N-1)}e^{ik_nL/2}\sum_{m=0}^{N-1} w_m\tilde{\zeta}_T(k_m)e^{-ik_0m\Delta}e^{-2\pi inm/N}$$
$$= \frac{L}{2\pi(N-1)}e^{ik_nL/2}\mathrm{DFT}_n[w_m\tilde{\zeta}_T(k_m)e^{-ik_0m\Delta}]$$

Therefore, the FFT algorithm allows to compute $\tilde{z}_T$ and option prices for the log strikes $k_n = k_0 + \frac{2\pi n}{N\Delta}$. The log strikes are thus equidistant with the step d satisfying

$$d\Delta = \frac{2\pi}{N}.$$

This relationship implies that if we want to compute option prices on a fine grid of strikes, and at the same time keep the discretization error low, we must use a large number of points.

References

[1] R. Cont and P. Tankov. Financial modelling with jump processes. Chapman & Hall / CRC Press, 2003. 3, 14

[2] E. Voltchkova, *Integro-differential evolution equations: numerical methods and applications in finance*, PhD thesis, Ecole Polytechnique Paris, 2005. Available from `http://tel.ccsd.cnrs.fr/tel-00010842` (in French). 3, 4, 5, 6, 12, 13

[3] R. Cont and E. Voltchkova. Integro-differential equations for option prices in exponential Lévy models. *Finance & Stochastics*, 2005, Vol. 3, pp. 299–325. 3, 4

[8] We use the FFT with $N = 2^p$, so N is even.

[4] R. Cont and E. Voltchkova, *A finite difference scheme for option pricing in jump diffusion and exponential Lévy models*, SIAM Journal on Numerical Analysis, 2005, Vol. 43, No. 4, pp. 1596–1626. 5, 6, 12

[5] P. Carr and D. Madan. Option Valuation Using the Fast Fourier Transform. Journal of Computational Finance, 1998, Vol. 2, No. 2, pp. 61–73. 14

[6] P. Tankov, *Processus de Lévy en finance: problèmes inverses et modélisation de dépendance*. PhD thesis, Ecole Polytechnique Paris, 2004. Available from `http://tel.ccsd.cnrs.fr/tel-00007944` 14