$\tilde{I}_{j+1,i}$: this part must remain <u>explicit</u>, otherwise it is very hard to deal with.

$j+1$ KNOWN ; j UNKNOWN.

Now, let's implement this : we'll start from the θ-method that we implement for B&S & we'll modify it for Levy.

- We need new market params : λ, $\underbrace{mu, delta}_{\text{jump size} \sim N(\mu, \delta^2)}$
  jump times → λ

- Maybe enlarge the grid bounds : since B&S underestimates large events... You can enlarge it by starting from the B&S bounds $(r - \frac{\sigma^2}{2})T \pm 6 \times \sqrt{T}\sigma$; or by rule of thumbs, like

$$\begin{cases} x_{MIN} = \log\left(\frac{0.2\, S_0}{S_0}\right) = \log(0.2) ; \\ x_{MAX} = \log\left(\frac{3S_0}{S_0}\right) = \log(3) . \end{cases}$$

- Before dealing with matrices, we must compute α & λ:

1) code a ν fct:

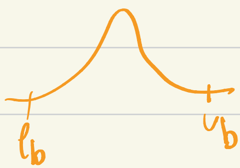
nu@(y) $\longrightarrow$ Lévy Measure.

2) code a fct:

[alpha, lambda_num, lb, ub] = Levy_integral(nu)

where we compute α & λ.

- <u>Integral domain truncation</u>:



we set our lb as the y s.t $\nu(y) < 10^{-10}$ on the left (so any point smaller than lb is $< 10^{-10}$). Same for the upper bound.

- <u>Quadrature</u>: $N_q = 2N$;

we can plot the "kernel", i.e, the Lévy measure between l_b & u_b.

$\leadsto$ use for example "trapz()" for trapezoidal quadrature formula.

3) change the matrices :

everywhere we have $r - \frac{\sigma^2}{2}$ we must add $(-\alpha)$ & everywhere we have r we must add λ.

4) In backward in time solution, we must add the integral I at time t_j. Like this :

Let's write what is known on the right hand side & what is unknown on the lhs :

$$-\frac{1}{\Delta t} V_{j,i} + (1-\theta)\left[\left(r - \frac{\sigma^2}{2} - \alpha\right)\frac{V_{j,i+1} - V_{j,i-1}}{2\Delta x} + \frac{\sigma^2}{2}\frac{V_{j,i+1} - 2V_{j,i} + V_{j,i-1}}{\Delta x^2} - (r+\lambda)V_{j,i}\right]$$

$$= -\frac{1}{\Delta t} V_{j+1,i} - \theta\left[\left(r - \frac{\sigma^2}{2} - \alpha\right)\frac{V_{j+1,i+1} - V_{j+1,i-1}}{2\Delta x} + \frac{\sigma^2}{2}\frac{V_{j+1,i+1} - 2V_{j+1,i} + V_{j+1,i-1}}{\Delta x^2} - (r+\lambda)V_{j+1,i}\right] - \tilde{I}_{j+1,i}$$

So this becomes :

$$M_1 v_j = M_2 v_{j+1} + BC - I$$

At each time step, we need to re-compute the integral I : levy_Integral2(lb, ub, x, V, nu).

Another fct with unique output I & computes the I integral.

We need an interpolation function to compute $v(t, x+y)$, knowing V and x ...

⚠ when we change the BCs, we also have to change BCs in the integral's definition.
↑ This is the MAIN changing point wrt B&S PDE.

→ For Barrier options, the barrier gives one of the bounds x_{min} or x_{max}. For example, for Down & Out, $x_{min} = \log\left(\frac{D}{S_0}\right)$.

→ For AM Put Option : don't use the discount factor on the BC.
And solve the linear system by PSOR to discriminate -step by step- whether we

are in Early Exercise or Continuation.

PS: we can do the same for Kou.

So what is missing here is to know how to deal with the Lévy processes that DON'T belong to F.A. Lévy $\longrightarrow$ I.A Lévy.

RN γ i.e $\psi_X(-i) = 0$ (γ fixed).

Recap: PIDE (γ, σ^2, ν)

(EU) Call option

$S_t = S_0 e^x$; $c(t,x) \longleftrightarrow c(T,x) = (S_0 e^x - K)^+$

$$\underbrace{\frac{\partial c}{\partial t} + \left(r - \frac{\sigma^2}{2}\right)\frac{\partial c}{\partial x} + \frac{\sigma^2}{2}\frac{\partial^2 c}{\partial x^2} - rc}_{B\&S} + \int_{\mathbb{R}} \left(c(t,x+y) - c(t,x) - (e^y - 1)\frac{\partial c}{\partial x} \right) \nu(y)\,dy.$$

where $c := c(t,x)$.

1st IDEA: F.A so we can split the integral.

$\alpha = \int_{\mathbb{R}} (e^y - 1)\nu(y)\,dy$, $\lambda = \int_{\mathbb{R}} \nu(y)\,dy$ & then:

The PIDE becomes:

$$\frac{\partial c}{\partial t} + \left(r - \frac{\sigma^2}{2} - \alpha\right)\frac{\partial c}{\partial x} + \frac{\sigma^2}{2}\frac{\partial^2 c}{\partial x^2} - (r+\lambda)c + \int_{\mathbb{R}} c(t,x+y)\nu(y)\,dy.$$

This part is discretized as we want:

- EE ;
- I.E ; $\longrightarrow$ unconditionally stable.
- Crank-Nicholson ...

only EXPLICITELY. ⚠

We know how to do it since it's exactly like B&S, but w/ α, λ.

• Let's discretize in time:

$c(t,x)$ $\qquad$ $t_j = j\Delta t$ $\qquad$ so we have:

$\frac{1}{2}$-discretization: discretized only on time.

$$\frac{c(t_{j+1},x) - c(t_j,x)}{\Delta t} + \left(r - \frac{\sigma^2}{2} - \alpha\right)\frac{\partial c}{\partial x}(t_j,x) + \frac{\sigma^2}{2}\frac{\partial^2 c}{\partial x^2}(t_j,x) - (r+\lambda)\,c(t_j,x) + \boxed{\int_{\mathbb{R}} c(t_{j+1},x+y)\,\nu(y)\,dy} = 0$$

Now let's put the unknown on the left, the known on the rhs: $J(t_{j+1},x)$

$$-\frac{1}{\Delta t}c(t_j,x) + \left(r - \frac{\sigma^2}{2} - \alpha\right)\frac{\partial c}{\partial x}(t_j,x) + \frac{\sigma^2}{2}\frac{\partial^2 c}{\partial x^2}(t_j,x) - (r+\lambda)\,c(t_j,x)$$

$$= -\frac{1}{\Delta t}c(t_{j+1},x) - J(t_{j+1},x)$$

- **Let's discretize also in x (log price):**

x_i, $i \in [\![0, N]\!]$. $\begin{cases} x_i = x_{MIN} + i\Delta x \\ \text{where } \Delta x = \dfrac{x_{MAX} - x_{MIN}}{N} \end{cases}$

$$\text{BCs.} \begin{cases} c(t_j, x_0) = 0 \quad \forall j \\ c(t_j, x_N) = S_0 e^{x_N} - K e^{-r(T - t_j)} \quad \forall j \end{cases}$$

And inside the domain ($i \in [\![1, N-1]\!]$):

$$-\frac{1}{\Delta t} c_{j,i} + \left(r - \frac{\sigma^2}{2} - \alpha\right) \frac{c_{j,i+1} - c_{j,i-1}}{2\Delta x} + \frac{\sigma^2}{2} \frac{c_{j,i+1} - 2c_{j,i} + c_{j,i-1}}{\Delta x^2}$$

$$-(r + \lambda) c_{j,i} = -\frac{1}{\Delta t} c_{j+1,i} - J(\underbrace{t_{j+1}}_{\text{Explicit.}}, x_i).$$

For a Put, in the code we need to change the payoff, the BCs BUT ALSO the Lévy Integral. The difference is in the BC, when $z <= x(1)$.

↳ Possible to check with CM using P-C parity.