But before we need to write everything in MATRIX FORM:

$$
\underbrace{\begin{bmatrix} 1 & 0 & 0 & \cdots & & 0 \\ A & B & C & 0 & \cdots & 0 \\ 0 & A & B & C & \cdots & 0 \\ 0 & 0 & A & B & \cdots & 0 \\ \vdots & & & & \ddots & \vdots \\ 0 & 0 & \cdots & & \cdots & C \\ 0 & 0 & \cdots & & 0 & 1 \end{bmatrix}}_{M1}
\begin{bmatrix} V_{j,0} \\ V_{j,1} \\ \vdots \\ V_{j,N} \end{bmatrix}
=
\underbrace{\begin{bmatrix} 0 & \cdots & & & & 0 \\ \tilde{A} & \tilde{B} & \tilde{C} & 0 & \cdots & 0 \\ 0 & \tilde{A} & \tilde{B} & & \cdots & 0 \\ \vdots & & & \ddots & & \vdots \\ 0 & \cdots & & \tilde{A} & \tilde{B} & \tilde{C} \\ 0 & \cdots & & & 0 & 0 \end{bmatrix}}_{M2}
\begin{bmatrix} V_{j+1,0} \\ V_{j+1,1} \\ \vdots \\ V_{j+1,N} \end{bmatrix}
+
\underbrace{\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ S_0 e^{x_{MAX}} - K e^{-r(T-t_j)} \end{bmatrix}}_{BC_j}
$$

i.e : $$\underline{V_j} = (M1)^{-1}\left(M2\,\underline{V_{j+1}} + BC_j\right).$$

- We did theta-Method for EU CALL.
Now we want price EU PUT.

1st change: Payoff is now: $(K - S_0 e^{x})^+$

2nd change: BC(t) is now: $x_{MIN} \longrightarrow Ke^{-r(T-t)} - S_0 e^{x_{min}}$
$x_{MAX} \longrightarrow V = 0$

So to go from Eu Call → Eu Put we just need to change the payoff & the BC in the code.

- **B&S "classical" PDE:** NO LOG PRICE

$$\frac{\partial V}{\partial t} + rS\frac{\partial V}{\partial S} + \frac{\sigma^2}{2}S^2\frac{\partial^2 V}{\partial S^2} - rV = 0$$

$S \in [0, +\infty)$

Two possible truncations

$[0, S_{MIN}]$: PRO = no lower bound truncat° error. TR. error $= \Delta S^*$.

$\underline{\text{Given } N,}$

Smaller Finite Difference (FD) $\Leftarrow$ $\left\{ \begin{array}{l} [S_{MIN}, S_{MAX}] : \text{PRO} = \\ \Delta S = \dfrac{S_{MAX} - S_{MIN}}{N} < \dfrac{S_{MAX}}{N} = \Delta S^* \end{array} \right.$

error: $\mathcal{O}(\Delta S^2) < \mathcal{O}(\Delta S^{*2})$.

$[0, S_{MIN}]$: $S_i = i \Delta S^*$ $i \in [\![0, N]\!]$ $\Delta S^* = \dfrac{S_{MAX}}{N}$.

$[S_{MIN}, S_{MAX}]$: $S_i = i \Delta S$ $i \in [\![0, N]\!]$ $\Delta S = \dfrac{S_{MAX} - S_{MIN}}{N}$

So we choose the smallest FD error:

- $\Delta t = \dfrac{T}{M}$, $t_j = j \Delta t$, $j \in [\![0, M]\!]$.

- $S_i = S_{MIN} + i \Delta S$, $\Delta S = \dfrac{S_{MAX} - S_{MIN}}{N}$, $i \in [\![0, N]\!]$.

We denote $V_{j,i} = V(t_j, S_i)$: price of the derivative at t_j, and with asset price being S_i.

We want to solve this problem:

$$\begin{cases} \dfrac{\partial V}{\partial t} + rS\dfrac{\partial V}{\partial S} + \dfrac{\sigma^2}{2}S^2\dfrac{\partial^2 V}{\partial S^2} - rV = 0 & \forall t\in[0,T],\ \forall S\in [S_{MIN}, S_{MAX}]; \\ V(T,S) = (S-K)^+; & \\ V(t,S_{MIN}) = 0; & \\ V(t,S_{MAX}) = S_{MAX} - Ke^{-r(T-t)}. & \end{cases}$$

The truncated version of the problem.
This is what we want to solve.

We can proceed as in the following:

t_j, S_i, $V_{j,i} = V(t_j, S_i)$, for $i \in [\![1, N-1]\!]$:

$$\frac{\partial V}{\partial t}(t_j,S_i) + \underbrace{rS_i\frac{V_{j,i+1}-V_{j,i-1}}{2\Delta S} + \frac{\sigma^2}{2}S_i^2\frac{V_{j,i+1}-2V_{j,i}+V_{j,i-1}}{\Delta S^2} - rV_{j,i}} \overset{O(\Delta S^2)}{=} 0$$

Terminal condition: $j+1$ known $\longrightarrow$ j unknown.

<u>Implicit Euler:</u>

$$\frac{V_{j+1,i} - V_{j,i}}{\Delta t} + (\;---\;) \overset{O(\Delta t + \Delta S^2)}{=} 0$$

$$\begin{bmatrix} A_i & B_i & C_i \end{bmatrix} \begin{bmatrix} V_{j,i-1} \\ V_{j,i} \\ V_{j,i+1} \end{bmatrix} = -\frac{1}{\Delta t} V_{j+1,i} \qquad \text{with :}$$

$$\left.\begin{aligned} A_i &= \frac{-rS_i}{2\Delta S} + \frac{\sigma^2}{2\Delta S^2} S_i^2 \\ B_i &= -\frac{1}{\Delta t} - \frac{\sigma^2 S_i^2}{\Delta S^2} - r \\ C_i &= \frac{rS_i}{2\Delta S} + \frac{\sigma^2 S_i^2}{2\Delta S^2} \end{aligned}\right\} \text{They depend on } i\,!$$

$$\begin{bmatrix} 1 & 0 & \cdots & & & & 0 \\ A_1 & B_1 & C_1 & 0 & \cdots & & 0 \\ 0 & A_2 & B_2 & C_2 & 0 & \cdots & 0 \\ & & \cdots & & & & \\ 0 & \cdots & & & A_{N-1} & B_{N-1} & C_{N-1} \\ 0 & \cdots & & & & 0 & 1 \end{bmatrix}$$

We can implement it on MATLAB:
cf Implicit_Euler_ClassicPDE.m

Look at code, S_0 is used only @ the end. So:

What about the greeks ?

$$\Delta = \frac{\partial V}{\partial S} \simeq \frac{V_{0,i+1} - V_{0,i-1}}{2\Delta S}$$

$$\Gamma = \frac{\partial^2 V}{\partial S^2} \simeq \frac{V_{0,i+1} - 2V_{0,i} + V_{0,i-1}}{\Delta S^2}$$

see the code where we compute greeks.

3rd position

1st position

For example: $$\Delta(S_1) = \frac{V_{0,2} - V_{0,0}}{2\Delta S}$$

because:

$$V_0 = \begin{bmatrix} V_{0,0} \\ V_{0,1} \\ V_{0,2} \\ \vdots \\ V_{0,N} \end{bmatrix}$$

Similarly: $$\Delta(S_{N-1}) = \frac{V_{0,N} - V_{0,N-2}}{2\Delta S}$$

→ So we cannot compute $\cancel{\Delta(S_0)}$, $\cancel{\Delta(S_N)}$.

In the code, we also plot Δ & Γ vs "node".

For the greeks, how can we do when we deal with the log price PDE?

$$V(t,S) \rightsquigarrow \hat{V}(t,x) = V(t, S_0 e^x)$$

$$\frac{\partial V}{\partial S}(t,S) = \frac{\partial V}{\partial x}\frac{\partial x}{\partial S} = \frac{\partial \hat{V}}{\partial x}\frac{\partial \log(S/S_0)}{\partial S} = \frac{1}{S}\frac{\partial \hat{V}}{\partial x} = \frac{1}{S_0 e^x}\frac{\partial \hat{V}}{\partial x}.$$

We can do it for Call & for Put:

only the payoff (@ maturity) & the BCs change.