FORWARD :

$$\underbrace{A V_{j,i-1}}_{\text{UNKNOWN}} + \underbrace{B V_{j,i}}_{\text{UNKNOWN}} + \underbrace{C_{j,i+1}}_{\text{UNKNOWN}} = -\frac{1}{\Delta t} \underbrace{V_{j+1,i}}_{\text{KNOWN}}$$

$$\begin{cases} A = -\dfrac{(r - \sigma^2/2)}{2\Delta x} + \dfrac{\sigma^2}{2} \dfrac{1}{\Delta x^2} \\ B = -\dfrac{1}{\Delta t} - \dfrac{\sigma^2}{\Delta x^2} - r \\ C = \dfrac{(r - \sigma^2/2)}{2\Delta x} + \dfrac{\sigma^2}{2\Delta x^2} \end{cases}$$

Implicit Euler Scheme.

↓

Because the equation has 3 unknown : not enough with one equation to give a solution. But with the boundary conditions we can do it.

Now, let's implement these two methods on MATLAB.

- Price a EU Call option, under B&S, using Finite difference by EXPLICIT EULER:

2022/2023: EU_CALL_ExplicitEuler.m

When we plot the call price vs spot: we obtain very different graphs: indeed, we make an error at each iteration. And, the Explicit Euler Scheme is instable (the errors cumulate). And we do not know exactly how large M should be. "M >> N - conditionally stable". TRY & ERROR.

Very simple but conditionally stable. So let's write implicit Euler scheme.

- Same but with IMPLICIT EULER:

2022/2023 : EUCALL_ImplicitEuler.m

$$V_{j,0} = 0 \quad \text{Boundary condition.}$$

$$\begin{cases} A V_{j,0} + B V_{j,1} + C V_{j,2} = -\frac{1}{\Delta t} V_{j+1,1} \\ A V_{j,1} + B V_{j,2} + C V_{j,3} = -\frac{1}{\Delta t} V_{j+1,2} \\ \vdots \\ A V_{j,N-2} + B V_{j,N-1} + C V_{j,N} = -\frac{1}{\Delta t} V_{j+1,N-1} \end{cases}$$

$$V_{j,N} = S_{max} - K e^{-r(T - j\Delta t)} \quad \text{B. condition.}$$

This corresponds to:

$$\underbrace{\begin{bmatrix} 1 & 0 & \cdots & & 0 \\ A & B & C & 0 \cdots & 0 \\ 0 & A & B & C \cdots & 0 \\ \vdots & & & & \\ 0 & \cdots & 0 & A & B\ C \\ 0 & \cdots & & 0 & 1 \end{bmatrix}}_{\text{"mat A"}} \underbrace{\begin{bmatrix} V_{j,0} \\ \vdots \\ V_{j,N} \end{bmatrix}}_{\text{"V"}} = \underbrace{\begin{bmatrix} 0 \\ -\frac{1}{\Delta t} V_{j+1,1} \\ \vdots \\ -\frac{1}{\Delta t} V_{j+1,N-1} \\ S_{max} - K e^{-r(T-j\Delta t)} \end{bmatrix}}_{\text{"Rhs"}}$$

→ we can easily solve this via MATLAB.

```
V = matA\rhs.
```

"mat A" in the code.

matA can be defined as a sparse matrix.

04/11/2024

Theta method :

Imp. Euler $\theta = 0$

Exp. Euler $\theta = 1$

$\Big\} \longrightarrow$ THETA METHOD.

$$\frac{V_{j+1,i} - V_{j,i}}{\Delta t} + \left(r - \frac{\sigma^2}{2}\right)\frac{V_{j,i+1} - V_{j,i-1}}{2\Delta x} + \frac{\sigma^2}{2}\frac{V_{j,i+1} - 2V_{j,i} + V_{j,i-1}}{\Delta x^2} - rV_{j,i} = 0$$

~ : UNKNOWN

Implicit Euler

$$\frac{V_{j+1,i} - V_{j,i}}{\Delta t} + \left(r - \frac{\sigma^2}{2}\right)\frac{V_{j+1,i+1} - V_{j+1,i-1}}{2\Delta x} + \frac{\sigma^2}{2}\frac{V_{j+1,i+1} - 2V_{j+1,i} + V_{j+1,i-1}}{\Delta x^2} - rV_{j+1,i} = 0$$

~ : UNKNOWN

$\hookrightarrow$ Explicit Euler (but $j \leftarrow j+1$.)

Let's compute : $(1-\theta)\,\mathrm{IE} + \theta\,\mathrm{EE} = 0 + 0 = 0$

it gives us : for $\theta \in (0,1)$,

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

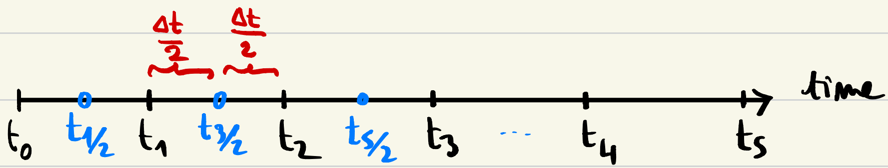
$$+\,\theta\left[\left(r - \frac{\sigma^2}{2}\right)\frac{V_{j+1,i+1} - V_{j+1,i-1}}{2\Delta x} + \frac{\sigma^2}{2}\frac{V_{j+1,i+1} - 2V_{j+1,i} + V_{j+1,i-1}}{\Delta x^2} - rV_{j+1,i}\right] = 0.$$

THETA METHOD :

- The ERROR is $O(\Delta t + \Delta x^2)$.
- Unconditionnally stable if $\theta \leq 1/2$. (when we give a little bit more IE than EE)
- But, if $\theta = 1/2$, ERROR $= O(\Delta t^2 + \Delta x^2)$, known as CRANK-NICHOLSON SCHEME.

→ **But, we can also derive directly the "CN" scheme:**

Time: t_j, $j \in [\![0, M]\!]$.



Let's consider the formula:

$$\frac{\partial V}{\partial t}\left(t_{j+1/2}, x_i\right) = \frac{V_{j+1,i} - V_{j,i}}{2 \times \left(\frac{\Delta t}{2}\right)} + O\left(\left(\frac{\Delta t}{2}\right)^2\right)$$

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

So we have :

$$\underbrace{\overbrace{\frac{V_{j+1,i}-V_{j,i}}{\Delta t}}^{\text{center scheme}}}_{O(\Delta t^2)} + \underbrace{\left(r-\frac{\sigma^2}{2}\right)\overbrace{\frac{V_{j+\frac{1}{2},i+1}-V_{j+\frac{1}{2},i-1}}{2\Delta x}}^{\text{center scheme}} + \frac{\sigma^2}{2}\overbrace{\frac{V_{j+\frac{1}{2},i+1}-2V_{j+\frac{1}{2},i}+V_{j+\frac{1}{2},i-1}}{\Delta x^2}}^{\text{center scheme}}}_{O(\Delta x^2)} - rV_{j+\frac{1}{2},i}=0$$

$$O(\Delta t^2+\Delta x^2)$$

Now, we just need to be able to write :

$$V_{j+1/2,i} = \frac{1}{2}V_{j+1,i} + \frac{1}{2}V_{j,i} + O(\Delta t^2) \qquad (*)$$

and to replace in the equation above to get the Crank-Nicholson Scheme.

<u>Proof of (*)</u> : By Taylor-Expansion,

$$\cdot\; V_{j,i} = V(t_j,x_i) = V\left(t_{j+1/2}-\frac{\Delta t}{2},x_i\right) = V\left(t_{j+\frac{1}{2}},x_i\right)-\frac{\Delta t}{2}\times$$

$$V'\left(t_{j+1/2},x_i\right)+\frac{1}{2}\frac{\Delta t^2}{4}V''\left(t_{j+1/2},x_i\right)+\dots$$

• $V_{j+1,i} = V_{j+\frac{1}{2},i} + \frac{\Delta t}{2} V'(t_{j+1/2}, x_i) + \frac{1}{2}\frac{\Delta t^2}{4} V''(t_{j+1/2}, x_i) + \dots$

Let's sum the 2, & $\times \frac{1}{2}$:

$$\frac{1}{2}\left(V_{j,i} + V_{j+1,i} \right) = V_{j+\frac{1}{2},i} + \Theta(\Delta t^2) \quad \blacksquare$$

• So, the θ-method scheme is :

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

$$+ \theta\left[\left(r - \frac{\sigma^2}{2}\right)\frac{V_{j+1,i+1} - V_{j+1,i-1}}{2\Delta x} + \frac{\sigma^2}{2}\frac{V_{j+1,i+1} - 2V_{j+1,i} + V_{j+1,i-1}}{\Delta x^2} - rV_{j+1,i}\right] = 0$$

Let's write it in VECTOR FORM :

$$\begin{bmatrix} A & B & C \end{bmatrix} \begin{bmatrix} V_{j,i-1} \\ V_{j,i} \\ V_{j,i+1} \end{bmatrix} = \begin{bmatrix} \tilde{A} & \tilde{B} & \tilde{C} \end{bmatrix} \begin{bmatrix} V_{j+1,i-1} \\ V_{j+1,i} \\ V_{j+1,i+1} \end{bmatrix}$$

Now let's implement this on MATLAB.