Crank-Nicolson method for vanilla TRS

I – Fully implicit method/CK method

Contrary to previously highlighted explicit method, implicit method discretization leads to following formulas for TRS:

$$\frac{\partial \mathrm{TRS}}{\partial \mathrm{t}} + (\mathrm{r} - \mathrm{C}_{\mathrm{yield}})\mathrm{S} - \mathrm{rTRS} = 0$$

$$\xRightarrow[\text{discrete}]{} \frac{\mathrm{TRS}_{\mathrm{t+1}}^{\mathrm{k}} - \mathrm{TRS}_{\mathrm{t}}^{\mathrm{k}}}{\delta \mathrm{t}} + (\mathrm{r} - \mathrm{C}_{\mathrm{yield}})\mathrm{S}^{\mathrm{k}} - \mathrm{rTRS}_{\mathrm{t+1}}^{\mathrm{k}} = 0$$

But contrary to option's equation and due to delta one characteristics, the target discretization is highly simplified without the usual linear equations system solver:

$$\mathrm{TRS}_{\mathrm{t+1}}^{\mathrm{k}} = \frac{\delta \mathrm{t}(\mathrm{r} - \mathrm{C}_{\mathrm{yield}})}{1 + \mathrm{r}\delta \mathrm{t}} \mathrm{S}^{\mathrm{k}} + \frac{1}{1 + \mathrm{r}\delta \mathrm{t}} \mathrm{TRS}_{\mathrm{t}}^{\mathrm{k}}$$

Then this methodology in vanilla TRS context can be seen as an "exercise" or POC, in order to show that algorithm/methodology also works for TRS.

Still du to simplified formula and hence simplified discretization, fully implicit and Crank-Nicolson method are equal.

II – Program of study – numerical results

- The model will be developed through python and C++ through finite difference method using CK scheme.
- Output value should be like explicit scheme.