ESD 40.317 Financial Systems Design
Homework 6

Due: 5 Aug before midnight

Consider the Constant Elasticity of Variance (CEV) model for risk-neutral stock price evolution. It has the following SDE:

$$dX = rXdt + \sigma X^{\beta} dW$$

This SDE's corresponding PDE for pricing puts and calls is:

$$\frac{1}{2}\sigma^2 x^{2\beta} f_{xx} + rxf_x - rf + f_t = 0$$

Which after the t → τ change of variables we discussed becomes:

$$\frac{1}{2}\sigma^2 x^{2\beta} f_{xx} + rxf_x - rf = f_\tau \quad \text{where we use } \beta = 0.75$$

In this assignment you will study alternative approaches to compute the price of a European up-and-in call option with the following terms: expiration in 0.5 years; strike price of \$100; and knock-in barrier at \$115. The current stock price x_0 is \$110; r is 2%; and σ is 25%. The option is only activated if the barrier has been knocked in meaning if the spot price hits or breaches the barrier. If not, the option is not activated and will not pay out. The barrier "monitoring style" for this option is also European, meaning that the check for knocking in is only performed once, at expiration. This is called European Knock-In or EKI in the industry.

1. Solve this PDE using the Explicit FDM with the number of steps for the asset price as an input and determine the necessary number of time steps to maintain the stability condition within the algorithm. Experiment with the number of asset steps required to get the option value at x_0 as a percentage of x_0 rounded to 2 decimal places to be ~4.35%.
2. Use Richardson's extrapolation on the Explicit FDM by computing two solution vectors where the second vector doubles the number of asset price steps. Then use the values from the second vector to adjust the values in the first solution vector. Finally, find the number of asset price steps required to get to the same accuracy as in 1.
3. Solve the PDE using the Crank Nicolson method with the number of time steps as an input. Use 500 time steps and vary the asset price steps to get to the same accuracy as 1.
4. Plot the option prices from all 3 methods on top of each other to confirm you have implemented them correctly.
5. Measure the running time of the different methods using the magic command %%time in jupyter notebook. You can also use %%timeit which will measure multiple runs of the algorithm to reduce noise. How do the running times compare?

Additional notes

1) Let the upper limit of the stock dimension (i.e. your approximation of an infinite stock price) be 2 times the strike price. This will be sufficient here as the volatility is not high and time to expiry is short in this example.

2) The formulas in the lecture slides and accompanying Jupyter notebooks are for the Geometric Brownian Motion (GBM) model, so you will need to modify them for the CEV model.