M.Phil. in Mathematical Finance

Numerical Methods in Finance II 2020

Tutorial 2 (Finite-difference Methods Continued)

Problems

1. As per the notes, implement the theta method finite-difference scheme for the Black-Scholes PDE. Price a European down-and-out barrier call option, with strike K and lower barrier l, using the following option related parameters

$$\sigma = 40\%, \quad r = 6\%, \quad K = 40, \quad l = 20, \quad T = 1$$

and the following mesh related parameters

$$S_{\max} = 180, \quad S_{\min} = l, \quad M = N = 80 \quad \text{and} \quad \theta = 0.75.$$

Hint: The initial condition and the upper boundary condition are the same as those for a standard call option. The lower boundary is 0 for all τ, corresponding to the knock-out condition. For debugging purposes define your initial condition as an inline function named `Uinit`, and name the boundary functions `U0` and `Uinf`, respectively.

Provide a surface plot of the result (in terms of S and $t = T - \tau$). In a second graph, plot the difference between the finite difference solution at $t = 0$ and the closed form value given by

$$C = S\Phi(d_1) - Ke^{-rT}\Phi(d_1 - \sigma\sqrt{T}) - S\Phi(d_2)\left(\frac{l}{S}\right)^{2(\mu+1)} + Ke^{-rT}\Phi(d_2 - \sigma\sqrt{T})\left(\frac{l}{S}\right)^{2\mu}$$

where

$$\begin{aligned} \mu &= \frac{r - \sigma^2/2}{\sigma^2} \\ d_1 &= \frac{\log(S/K)}{\sigma\sqrt{T}} + (1+\mu)\sigma\sqrt{T} \\ d_2 &= \frac{\log(l^2/(SK))}{\sigma\sqrt{T}} + (1+\mu)\sigma\sqrt{T}. \end{aligned}$$

Using logical indexing on the finite difference solution, display the at-the-money price of the option at inception.

2. Modify the code in Question 1 by replacing the matrix inversion with an implementation of the SOR algorithm. Use an over-relaxation parameter of $\omega = 1.4$ and a convergence tolerance of `1e-5`.

Hint: You will need to implement the matrix inversion as a `while` loop nested in the `for` loop that iterates through the time indices.

NB: Please ensure that this problem is working correctly — you will reuse this code in a subsequent tutorial.

3. (Homework) Change the boundary conditions and parameters in Question 1 to implement the double boundary call option in the lecture notes. Check that you are able to reconstruct the graphs in the example.

4. (Homework) Implement the Thomas algorithm for decomposing a tridiagonal system into corresponding lower and upper triangular matrices. Now implement the code for forward and backward substitution and use it to replace the Matlab matrix inversion in Question 1.