Computational Finance
FIN-472
Take-Home Exam 4

November 18, 2022

- **Please upload your solutions on moodle by Friday 02.12.2022 at 8h15.**
- **Please do not forget to upload in Moodle your Matlab codes. You should submit exactly one Matlab file for each part.**
- **You should also upload one PDF file containing the Matlab codes and the plots.**

Exercise 1: Consider the following PDE for the time-dependent Black & Scholes equation in asset-prices on bounded domain:

$$\begin{cases} \partial_t V(S,t) - \dfrac{\sigma^2}{2} S^2 \partial_{SS} V(S,t) - rS\partial_S V(S,t) + rV(S,t) = 0 \quad \text{in } [0, S_{max}) \times (0,T], \\ V(S_{max}, t) = 0, \\ V(S,0) = G(S) = \max\{K - S, 0\}. \end{cases} \tag{1}$$

Observe that in (1) we assume $S_{min} = 0$.

a) Consider the sensitivity parameter *Vega* that measures the sensitivity of the option price to volatility, i.e. the derivative of the option value with respect to the volatility of the underlying asset $\nu = \partial_\sigma V$. *Derive* (detailing the steps of your procedure) the PDE for ν by differentiating equation (1). *Set* the initial and boundary conditions equal to zero. *What kind* of equation did you obtain?

b) Consider now the parameters $\sigma = 0.2$, $r = 0.015$, $K = 1$, $T = 1$, and *fix* S_{max} so that the domain truncation error is less than 10^{-6}. Starting from the solver `bs_timestepping.m` written in Exercise 1 of Homework 8, *write* a Matlab code that numerically solves the equation for ν. In particular,

   i) *Compute* the numerical solution for problem (1) using a $\theta-$method with $\theta = 0.5$. Use a discretization with $N_x = 60$ intervals in the physical space and $N_t = 100$ time steps.

ii) *Use* the numerical result of the part (i) to approximate by finite differences the right hand side of the PDE solved by ν (the one you obtained in part a), see also slide 5 of Lecture 8-b). Finally, numerically *solve* the PDE for ν obtaining an approximate solution ν_h. Use again $\theta = 0.5$.

iii) *Compare* ν_h with the exact solution ν that can be obtained by using the matlab function `blsvega` (see the **moodle** page). In particular, *plot* $\nu_h(T, S_i)$ and $\nu(T, S_i)$ for all values S_i that belong to `FD_grid(2:end)`.

Remark: The comparison is asked to be performed in the discretized interval $(0, S_{max}]$ because the command `blsvega` takes only values of the underlying asset strictly larger than zero.

c) *Verify* the convergence properties of the finite difference solver: *solve* numerically the equation for *Vega* on a sequence of increasingly finer computational grids, generated by choosing N_x (the number of intervals in space) as $N_x = (10 \times 2^i)$ for $i = 0, 1, \ldots, 5$ and N_t (the number of time-steps) as $N_t = 0.6 \times N_x$, and compute for each discretization level the error of the finite difference solution

$$e(h, T) = \max_i |\nu_h(S_i, T) - \nu(S_i, T)|, \quad h = \frac{S_{max}}{N_x}$$

where the exact solution $\nu(S_i, T)$ can be again computed with the command `blsvega`. *Plot* $e(h, T)$ against all the values of h. What order of convergence do you observe?

d) i) Use the central finite difference scheme to *directly approximate* ν as

$$\nu_{\delta\sigma} = \frac{V(S, t, \sigma + \delta\sigma) - V(S, t, \sigma - \delta\sigma)}{2\delta\sigma}$$

where $V(S, t, \sigma + \delta\sigma)$ is the numerical solution of (1) with volatility equal to $\sigma + \delta\sigma$ and $V(S, t, \sigma - \delta\sigma)$ is the numerical solution of (1) with volatility $\sigma - \delta\sigma$. *Compute* $\nu_{\delta\sigma}$ for $\delta\sigma = [0, 0005; 0, 001; 0, 01; 0, 02; 0, 03; 0, 05]$.

ii) *Compare* the two methods (the one of part b) and the one of part d)i)!) to approximate ν. In particular, on the same plot of part c), *plot* the approximation errors

$$e_{\delta\sigma}(h, T) = \max_i |\nu_{\delta\sigma}(S_i, T) - \nu(S_i, T)|, \quad h = \frac{S_{max}}{N_x}$$

for all values of $\delta\sigma$. As done in part b), consider a sequence of increasingly finer computational grids generated by choosing N_x as $N_x = (10 \times 2^i)$ for $i = 0, 1, \ldots, 5$ and N_t as $N_t = 0.6 \times N_x$. Plot $e_{\delta\sigma}(h, T)$ against all the values of h. *Describe* the obtained results. Which method is more accurate?