Computational Finance
FIN-472
Homework 9

November 17, 2017

Exercise 1: Consider the following PDE (steady Black & Scholes eq. in log-price on bounded domain).

$$\begin{cases} -\dfrac{\sigma^2}{2}\partial_{xx}V(x) - \left(r - \dfrac{\sigma^2}{2}\right)\partial_x V(x) + rV(x) = f(x), \quad x \in (x_{min}, x_{max}) \\ V(x_{min}) = a, V(x_{max}) = b. \end{cases} \tag{1}$$

a) Compute $f(x), a, b$ such that (1) is solved by $V(x) = 3x^2 + 1$.

b) Write a finite differences solver for (1), starting from the draft file `draft_bs_logform_steady.m`.

c) Let $\sigma = 1, r = 1, x_{min} = 0, x_{max} = 4$. Use the finite differences solver to compute a numerical solution u_h for problem (1), and verify that the result is correct by comparing graphically the exact solution and the approximated one (use $Nx = 251$ intervals). Moreover, verify that the finite differences error $e = \max_i |V(x_i) - V_h(x_i)|$ is 0 (up to machine precision).

Exercise 2: Consider the following PDE (time-dependent Black & Scholes in log-prices on bounded domain).

$$\begin{cases} \partial_t V(x,t) - \dfrac{\sigma^2}{2}\partial_{xx}V(x,t) - \left(r - \dfrac{\sigma^2}{2}\right)\partial_x V(x,t) + rV(x,t) = f(x,t), \; x \in (x_{min}, x_{max}), t \in (0, T] \\ V(x_{min}) = a(t), \quad V(x_{max}) = b(t) \\ V(x, 0) = G(x) \end{cases} \tag{2}$$

a) Compute $f(x,t), G(x), a(t), b(t)$ such that (2) is solved by $V(x,t) = 4t(3x^2 + 1)$.

b) Write the Forward Euler finite differences solver for (2), starting from the draft file `draft_bs_logform_forward_euler.m`.

c) Let $\sigma = 1, r = 1, x_{min} = 0, x_{max} = 1, T = 1$. Use the finite differences solver to compute a numerical solution u_h for problem (2), and verify that the result is correct by comparing graphically the real solution and the approximated one at final time (use $Nx = 21$ intervals in space and $N_t = 512$ intervals in time). Moreover, verify that the finite differences error $e = \max_i |V(x_i) - V_h(x_i)|$ is 0 (up to machine precision).

d) Repeat the previous point with $N_t = 256$. What happens? Explain the results obtained.

Exercise 3: Consider again the PDE of the previous exercise (2).

a) Write a θ-method time-stepping finite differences solver for (2), starting from the draft file `draft_bs_logform_timestepping.m`.

b) Let $f(x,t), a(t), b(t), G(x), \sigma, r, x_{min}, x_{max}, T$ as in the previous exercise. Use the θ-method finite differences solver with $\theta = 1$ and $\theta = 0.5$ to compute a numerical solution u_h for problem (2), and verify that the result is correct by comparing graphically the real solution and the approximated one (use $Nx = 21$ intervals in space and $N_t = 100$ time-intervals). Moreover, verify that the finite differences error $e = \max_i |V(x_i) - V_h(x_i)|$ is 0 up to machine precision.

Exercise 4: Consider again equation (2). Set $S_{min} = 0.6$ and $S_{max} = 1.65$, $x_{min} = \log(S_{min})$, $x_{max} = \log(S_{max})$, $f(x,t) = 0$, $a(t) = Ke^{-rt} - S_{min}$, $b(t) = 0$, $\sigma = 0.21$, $r = 0.015$, $T = 1$, and $G(x) = \max\{K - e^x, 0\}$, in order to solve the Black & Scholes model in log-prices for an European Put option with strike price $K = 1$. Use the Crank-Nicholson time-stepping method with 500 intervals in time and 925 intervals in space ($N_t = 500$, $Nx = 925$).

a) Plot the final value of the option versus the price of the underlying asset (<u>*not*</u> the log-price), and compare it with the Matlab built-in command `blsprice` (contained in the Matlab financial toolbox). Are the two solutions comparable?

b) Compute the finite differences approximation of the value of the option for $S = 0.9$, using the Matlab command `interp1` with `'linear'` option.

c) Given any tolerance *tol*, use the result stated in Proposition 4 of the class slides to obtain S_{min} and S_{max} such that the truncation error committed when solving the Black & Scholes model in $[S_{min}, S_{max}]$ instead of $(0, \infty)$ is smaller than *tol*. *Hint: derive S_{min}, S_{max} s.t.*

$$\frac{tol}{K} = \phi(\alpha_1(S_{min})), \quad \frac{tol}{K} = \phi(\alpha_2(S_{max}))$$

*and implement it in Matlab (you will need the quantiles of the standard normal distribution, which can be computed by the Matlab command **`norminv`**, contained in the Statistics Matlab toolbox).*

d) Consider a decreasing sequence of tolerances, $tol = [10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}]$. For each tolerance, compute the associated interval, solve numerically the equation, compute the error $e = \max_i |V(x_i, T) - V_h(x_i, T)|$ and verify that such error is smaller than the required tolerance (use a log-log plot to this end). Note that for a fair comparison, the spatial discretization should be kept constant as S_{max} increases (i.e. same Nx and same N_t).