Norwegian University of Science and Technology
Department of Mathematical Sciences

Project 1

Practical information

- *Deadline and hand-in:* Sunday February 27 (before midnight). Hand in the project in ovsys (pdf file + Jupyter Notebook).
- *Report:* The report can be written as a pdf-document together with your python code as Jupyter file. More details are present on the wiki page under "Exercises and projects". Write the report as a scientific report, not as a solution to an exercise. Meaning: Describe the problem you want to solve, describe the method you are using, write mathematical results as mathematical statements, and make sure there is a consistency between theoretical and numerical results etc. Use plots whenever appropriate, make sure they are readable, explain clearly what you observe and if that is what was expected.
- *Grading:* Max 4 points for the report, max 12 for problem 1, max 4 for problem 2 a) and b), and max 2 for problem 2 (c). Total max score 20 points (min(sum subproblem scores, 20)).
- *Learning objectives:* When completed this project you should demonstrate that you are able to:
  - develop and implement a finite difference scheme for variable coefficients parabolic problems.
  - perform an error analysis.
  - identify and solve potential deficiencies of a scheme.
  - communicate the results in a scientific manner.

Some advice:

- *Implementation:* Do not implement everything at once, split the work into smaller pieces, and make sure each of them works before you continue.
- *Writing:* Imagine you are writing for fellow students, think about how you make them understand and be interested in what you have learned during the project without having seen the project description. Writing takes a lot of time, so start early and accept that you may want to rewrite parts, that is part of the process.
- *Time organization:* Think about how much time you are willing to use on this project. If you are completely stucked at one point, maybe it is better to skip it and concentrate on writing a good report instead.

Problem background

In this project we will solve boundary value problems for the different version of the Black-Scholes (BS) equation. These are PDEs from mathematical finance describing the price of (European type) options.

Black-Scholes PDEs:

$$(1) \qquad u_t - \frac{1}{2}\sigma^2 x^2\, u_{xx} - rx\, u_x + c\, u = 0, \quad x \in \mathbb{R}^+,\ t \in (0,T), \qquad \text{(1-D } \textbf{linear} \text{ BS)}$$

$$(2) \qquad u_t - \frac{1}{2}x^2 \varphi(u_{xx}) u_{xx} = 0, \quad x \in \mathbb{R}^+,\ t \in (0,T), \qquad \text{(1-D } \textbf{nonlinear} \text{ BS)}$$

where $\sigma,\ r,\ c > 0$, are volatility, interest rate, dividends and correlation, and

$$\varphi(x) = \sigma_1^2 + \frac{\sigma_2^2 - \sigma_1^2}{2}\Big(1 + \frac{2}{\pi}\arctan x\Big),$$

and hence for $\sigma_2 > \sigma_1$, $\varphi(-\infty) = \sigma_1^2$, $\varphi(\infty) = \sigma_2^2$, $\varphi' > 0$, and $|\varphi'| < \frac{\sigma_2^2 - \sigma_1^2}{\pi}$ (check it!). Obs: $\varphi(r) > \sigma_1^2 > 0$ for all $x \in \mathbb{R}$.

Initial conditions: $K, H > 0$ strike prices.

$$u(x,0) = \max(K - x, 0) =: \big(K - x\big)^+ \qquad \text{(European put)}$$

$$u(x,0) = \big(x - K\big)^+ - 2\big(x - (K+H)\big)^+ + \big(x - (K+2H)\big)^+, \qquad \text{(butterfly spread)}$$

$$u(x,0) = \operatorname{sgn}^+(x - K), \quad \operatorname{sgn}^+(r) = \begin{cases} 0, & r < 0, \\ 1, & r \geq 0, \end{cases} \qquad \text{(binary call)}$$

Boundary conditions:

The domain is $\mathbb{R}^+$ and we need boundary conditions at $x = 0$ and $x = \infty$.

(i) At $x = 0$, the PDE becomes an ODE that can be solved explicitly from the initial condition. The PDE is invariant in $\mathbb{R}^+$ and no boundary condition is needed at $x = 0$ for the problem to be well-posed. Numerically, though, when we approximate, we make some errors and then a boundary condition is needed anyway.

(ii) $x = \infty$. Here we need artificial B.C.'s to have a finite computational domain. Let $x = R > 0$ denote right boundary. R needs to be large enough so that the B.C. imposed does not disturb the solution too much in the domain of interest (e.g. reltively near $x = 0$). The following artificial B.C.'s have been used in the literature:

$$\begin{aligned} u(R,t) &= 0 \qquad &&\text{(or, better} \qquad u(R,t) = u(R,0)), \\ u_x(R,t) &= 0 \qquad &&\text{(or, better} \qquad u_x(R,t) = u_x(R,0)), \\ u_{xx}(R,t) &= 0. \end{aligned}$$

Problems

$\boxed{1}$ For the 1-D **linear** Black-Scholes:

a) Solve it with artificial B.C. with a central difference in space and Forward Euler, Backward Euler, Crank-Nicolson in time. Write the program experimenting with some of the I.C.'s and B.C.'s given above.

b) Show that with Dirichlet B.C.'s, the Forward Euler and Crank-Nicolson schemes in part **a)** are monotone if $\sigma^2 > r$ and a CFL condition holds. Write explicitly what these CFL conditions are. Show that Backward Euler scheme is monotone.

c) Use part b) to show L^∞ stability in one or more cases where Dirichlet B.C.'s are used. Argue as we did in class, for implicit schemes you will need the discrete maximum principle.

d) Show consistency and L^∞ convergence for one of the schemes in part a). Can you get an error bound in L^∞?

e) Test the schemes, check the CFL condition in Forward Euler case numerically, check convergence rates, plot the solution and log-log plots of the errors in some cases. Which scheme is faster, and which is slower for a given accuracy? (Can you compute the computational time of your code in each case?)

Hint: To find an exact solution of the problem, you may add a right hand side f and change the initial data. Then as explained in the lectures, any nice function will be a solution for a suitable choice of f and the initial data.

f) For one specific problem and scheme, test different choices of the artificial B.C.'s and comment on the results. Check the effect of changing $R(> 2K)$ on the error in the spatial interval $[0, 2K]$.

$\boxed{2}$ For the 1-D **nonlinear** Black-Scholes:

(IMEX) $$\frac{1}{k}\nabla_t U_m^{n+1} = \frac{1}{2}x_m^2 \varphi\Big(\frac{1}{h^2}\delta_x^2 U_m^n\Big)\frac{1}{h^2}\delta_x^2 U_m^{n+1}$$

(BE) $$\frac{1}{k}\nabla_t U_m^{n+1} = \frac{1}{2}x_m^2 \varphi\Big(\frac{1}{h^2}\delta_x^2 U_m^{n+1}\Big)\frac{1}{h^2}\delta_x^2 U_m^{n+1}.$$

a) Implement and test the first scheme. Use a non-convex exact solution for testing. Plot the solution for the (butterfly spread) initial condition.

b) Compute the truncation error of the schemes and show they are both monotone in the sense that they can be written as

$$\alpha_m^{n+1}U_m^{n+1} - \sum_{i\neq 0}\beta_{m,i}^{n+1}U_{m+i}^{n+1} - \sum_{i}\beta_{m,i}^{n}U_{m+i}^{n} = 0, \quad \alpha,\beta \geq 0, \alpha \geq \sum_{n,i}\beta > 0.$$

Use this and the discrete maximum principle to prove L^∞-stability when Dirichlet B.C.'s are used.
Note: α and β may depend on U_k^n and U_l^{n+1} for some values of k and l.

c) (Bonus problem) Implement and test the second scheme. Which scheme is faster at a given accuracy?

Hint: You will need to solve a nonlinear system. Use e.g. a Newton method.