Math G4077 – Homework Assignment 7* – Fall 2012 © Paul Feehan

Practice or *advanced* problems are optional and are not graded: *practice problems* are intended as drill problems and aides to exam preparation while *advanced problems* are intended for students with additional mathematics background. Please consult the *Homework Submission Requirements* before commencing work on this assignment.

1. Reading Assignments and Sample Code

The primary reading assignments for this homework set are

- Y. Achdou and O. Pironneau, *Computational methods for option pricing*, chapter 6, American options.
- S. Crepey, sections 9–14, Finite difference and finite element methods for PIDEs.
- D. Duffy, *Finite difference methods in financial engineering.*
- L. Lapidus and G. F. Pinder, *Numerical solution of partial differential equations in science and engineering.*
- R. Seydel, *Computational finance.*
- G. Smith, *Numerical solution of partial differential equations.*
- D. Tavella, *Quantitative methods in derivatives pricing: an introduction to computational finance.*
- J. Thomas, *Numerical partial differential equations: finite difference methods.*
- P. Wilmott, *Quantitative Finance.*

2. Programming and Written Assignments

There are **no** required programming assignments this week due to final projects.

3. Practice Programming and Written Assignments

Problem 3.1. Write a C++ program to price American-style vanilla and single barrier put options with the Crank-Nicolson finite difference scheme using the following data.

- Spot process parameters:
  - Asset price process, $S(t)$ is geometric Brownian motion;
  - Volatility, $\sigma = 0.3$;
  - Initial asset price, $S(0) = 100$;
  - Dividend yield, $q = 0.02$;
- Discount curve parameters:
  - Constant risk-free interest rate, $r = 0.05$;
- Payoff parameters:
  - Put strike, $K = 90$;
  - Down-and-out barrier, $L = 80$;
  - Up-and-out barrier, $U = 120$;
  - Maturity, $T = 1$ year;
- Numerical method parameters:
  - $N_t = 252$/year time steps, $N_S = 50$ space intervals, minimum spot size $S_{\min} = 0.0$, and maximum spot size $S_{\max} = 200$ (without barrier).

*Last update: November 29, 2012 14:27

(a) Modify your Crank-Nicolson finite difference class `CrankNicolsonFD1` (header and source files `CrankNicolsonFD1.h` and `CrankNicolsonFD1.cpp`) from the previous assignment to `CrankNicolsonSORFD` (header and source files `CrankNicolsonSORFD.h` and `CrankNicolsonSORFD.cpp`), which now allows for early exercise by comparison with a payoff function at each time step.

(b) Add the SOR method to solve the matrix equations when the options are now allowed to be American or European-style, using projected SOR (PSOR) with American-style. Allow the user to select LU decomposition with either exercise style so the LU and SOR methods can be compared with both exercise styles.

(c) Write a *separate* main program, `CrankNicolsonSORFDMain.cpp`, which calls the above methods and produces the following outputs:

```
Closed-form European-style vanilla call price =
Crank-Nicolson PSOR American-style vanilla call price =
Crank-Nicolson PSOR American-style vanilla put price =
Crank-Nicolson LUD American-style vanilla put price =
Closed-form European-style DO barrier call price =
Crank-Nicolson PSOR American-style DO barrier call price =
Crank-Nicolson PSOR American-style DO barrier put price =
Crank-Nicolson LUD American-style DO barrier put price =
Closed-form European-style UO barrier call price =
Crank-Nicolson PSOR American-style UO barrier call price =
Crank-Nicolson PSOR American-style UO barrier put price =
Crank-Nicolson LUD American-style UO barrier put price =
```

(d) **Program notes:** The program data should be included in the main program file, but not the class definition. Your program code archive file (`*.zip` or `*.tar.gz` **only**) must be complete and self-contained: include **all** required files.

(e) **Benchmarking.** Benchmark your results using the Excel-VBA spreadsheets of Haug, Back, or Rouah-Vainberg, or the MATLAB functions of Brandimarte or Mathworks' toolboxes.

(f) **Report.** Write a report (LaTeXpreferred, though Word is acceptable) which includes a short explanation of the algorithms and their implementation and an analysis of your results and answers to the following questions.
- How do your results vary with use of LUD or PSOR?
- Can you choose $S_{\min} = L$ or $S_{\max} = U$?
- Are the stability and convergence criteria satisfied for each of the four suggested values of θ? Try a combination of step sizes to test robustness of the stability and convergence criteria.
- Compare the terminal condition and the upper or lower barrier boundary conditions: are the terminal and boundary conditions continuous where they meet at the corners of the rectangle $[0, T] \times (0, U]$ or $[0, T] \times [L, \infty)$? When there is discontinuity or continuity, which are the best FD schemes?
- Carefully explain why you need to replace LU decomposition with the SOR method to solve the matrix equation when pricing American-style options.
- Compare your numerical results with those the VBA code of Haug, Back, indicating the method used (binomial or trinomial tree, FD, or closed-form approximation),

or the MATLAB code of Brandimarte. You may choose $q = 0$ in your code when conducting benchmarking comparisons.