Module 6: Monte Carlo question (Mike Giles)

The aim in this question is to investigate the Longstaff-Schwartz method for obtain a lower bound on the price of an American put option, and the associated upper bound due to Rogers' dual formulation.

As a starting point, you are given two MATLAB codes which you can download from `http://people.maths.ox.ac.uk/gilesm/mc/` and can convert into other languages if you wish:

- `amer_fd.m` – an explicit finite difference approximation to give a reference value
- `amer_mc.m` – a simple Longstaff-Schwartz implementation with some deficiencies:
  - doesn't use a separate set of paths for the final pricing calculation
  - uses all paths in the least-squares regression, not just those which are in-the-money

The model parameters for the put option are $S_0 = 1$, $K = 1$, $T = 1$, $r = 0.05$, $\sigma = 0.2$, and the Monte Carlo computation uses 64 timesteps and 10^5 paths.

Look at the Monte Carlo code and the way in which it relates to the theory given in the lecture. The implementation is actually simpler/cleaner than you might expect, because of the way in which MATLAB defines $A\backslash b$. When there are more equations than unknowns, it actually solves the least squares problem $A^T A\, x = A^T b$ to calculate the output x, so $A\backslash b$ gives the x we need.

Assignment:

1. run the finite difference code, and then uncomment the two lines in the code which will double the number of grid points and quadruple the number of timesteps (the 2:4 ratio is needed for numerical stability) and re-run the code – this gives you an accurate idea of the true value of the American put option.

2. modify the Monte Carlo code to
    - use a separate set of 10^5 independent paths for the final price evaluation;
    - perform 100 independent runs (each with 10^5 paths) of the whole thing, computing the average of the 100 runs and a corresponding confidence interval;
    - repeat with a modification to perform the regression using only paths which are in-the-money.

3. using the Longstaff=Schwartz version which uses all of the paths to determine the regression coefficients (because we need the continuation value for all paths, not just those in the money) write a code to implement Rogers upper bound calculation, using 1000 paths, with 100 sub-paths at each timestep.

You should hand in the Monte Carlo codes you develop and a short writeup presenting your results and discussing whether they are as expected, based on the lectures.