### Problem Set 3

Total 15 points

Due Friday, 15 February 2018 before 11:59pm

*Instructions: This is a GROUP assignment.*

I think it makes sense to submit an excel spreadsheet if you work with excel/VBA or a word document with embedded figures if you use Matlab/Python/R etc.

*(a) Only submit one assignment per group but make it very clear who the group members are.*

*(b) The file name should be: PSn.FAMILYNAME1Initial1\_FAMILYNAME2Initial2….xls,*

*for example my solution to Problem set 1 would have the file name PS1.WIDDICKSM.xls.*

**1.** [3 points] Consider a European **put** option where S0 = 100, K = 105, r = 0.1,  = 0,  = 0.30 and T = 0.2.

1. Calculate the value of the European put options using the Black-Scholes formula.

We will now compare the performance of the following binomial lattice methods.

* Cox, Ross and Rubinstein, 1979 (*u = e√t*, *d = 1/u*)
* Rendleman and Bartter, 1979 (*u = exp((r--0.52)t+√t)*, *d = exp((r-0.52)t-√t)*)
* Leisen and Reimer, 1996 (see Lecture 4 notes)

**For each model perform the following:**

1. Calculate the value of the European put option for time steps ranging from N = 50 to N = 1000 (Do 51 to 999 for LR and only do odd numbers of steps)
2. Plot a graph of N (x-axis) against error (y-axis) when compared to the Black-Scholes price.
3. Explain the graph you obtain in c.
4. Attempt to improve the efficiency of the method by extrapolating with M and N = 2M (or M+1 and N = 2M-1 for LR) steps for a selection of M values (e.g. I will choose M = 25, 50, 100, 150,…500 for CRR/RB and M+1 = 51, 101, 151, ..501 (N = 999) for LR).

Does this give you more accurate option values than simply using a lattice with N timesteps? Why?

*Note: For the LR method focus only on the odd numbers of steps, it is exceptionally accurate with error of O(1/n2), so this will affect your extrapolation scheme – you will have to use 1/N^2 as the first error term and recalculate your extrapolation formula.*

**2.** [5 points] Consider an American put option where S0 = 100, K = 105, r = 0.1,  = 0,  = 0.3 and T = 0.2.

1. Calculate the value of the American put option to within $0.0000001 (using any binomial model you like), to do this you should use more and more timesteps (this should be lots, e.g 10000) until you are happy that you have converged to the correct value.

We will now compare the performance of the following binomial lattice methods.

* Cox, Ross and Rubinstein, 1979
* **Broadie and Detemple, 1996 (NOT Rendleman and Bartter, 1979 - see Lecture 4 notes)**
* Leisen and Reimer, 1996 (see Lecture 4 notes)

**For each model perform the following,**

1. Calculate the value of the American put option for time steps ranging from N = 50 to N = 1000 (again only do odd numbers of steps for LR). Plot a graph of N (x-axis) against error (y-axis) when compared to your answer from a. Explain the graph you obtain.
2. For the ***CRR model***, with N = 100, plot out the position of the early exercise boundary on a graph with time on the x-axis and the boundary level on the y-axis. (The exercise boundary is the value of S, Sf, at which you exercise at Sf and below and hold above Sf). Do you think that the boundary you obtain looks correct compared to the true (non-binomial) boundary? Do you think this will have any effect upon the accuracy of your option value?
3. Attempt to improve the efficiency of the method by extrapolating with M and   
   N = 2M (or M+1 and N = 2M-1 for LR) for the same M/N values as for the European option. Does this give you more accurate option values than simply using a lattice with N timesteps? Why?

*Note: For the LR method you may want to try the O(1/n) extrapolation as well as the O(1/n^2) to see which one gives better results – of course only use the odd number of steps.*

**3.** [4 points] Consider a down-and-call option where **S0 = 100, K = 100, B = 95, r = 0.1,  = 0,  = 0.3 and T = 0.2**. The down-and-out call is a barrier option where you receive max(ST-K,0) at expiry but if at any time before expiry St < B the option expires worthless.

The analytic formula is given by:



where d1 and d2 are as in the Black-Scholes formulae and



We will now analyze the performance of the Cox, Ross and Rubinstein, 1979 model:

1. Calculate the value of the Down-and-out call option for time steps ranging from N = 50 to N = 1000 and plot a graph of N (x-axis) against error (y-axis) when compared to the analytic price.
2. Can you explain the error profile that you observe? It may be useful to try to plot the position of the nodes relative to the *barrier* (similar to the Lambda graph for K in Lecture 4) to understand what is happening in this question.
3. Can you think of anyway of improving the convergence to the correct value? [I’m looking for ideas here rather than calculations but these may be useful for future valuations]

**4.** [3 points] Consider a down-and-call option where S0 = 100, K = 100, B =95, r = 0.1, d = 0,  = 0.3 and **T = 0.2** **only now the barrier is only applied on 4 dates, at t = 0.04, 0.08, 0.12 and 0.16**. This is a discretely monitored barrier option and does not have an analytic formula but following the methods in Kou (2008), the accurate value is 5.6711051343.

We will now analyze the performance of the Cox, Ross and Rubinstein, 1979 model:

* 1. Calculate the value of the Down-and-out call option for time steps ranging from N = 50 to N = 1000 **in steps of 10** (i.e 50, 60, 70, …) to ensure that the barrier times are always matched by a tree step and all the barrier steps are even numbers. Plot a graph of N (x-axis) against error (y-axis) when compared to the analytic price and explain the graph you obtain.
  2. Plot the position of the nodes relative to the barrier (as in the Lambda graph in Lecture 5). When does it appear that the tree gives the most accurate option values?