MATH60082
Examples Sheet 5
Binomial Trees

Stock Tree

5.1 Follow the instructions on the web (click here) using this template code as a start point (click here). Declare all the required parameters and a vector *stock_tree* into which we will place our stock price values. Setup the vector to be a $n+1 \times n+1$ 2D array, where n indicates the number of steps in the tree. Assign the appropriate values to the stock tree using *for* loops. It may look something like:

```
// move through time steps
for(int i=0;i<stock_tree.size();i++)
{
  // move up through j steps (this may need to change)
  for(int j=0;j<stock_tree[i].size();j++)
  {
    stock_tree[i][j] = /* fill this in */
  }
}
```

(i) Print the values out to the screen. Can you print only the nodes that are contained in the tree?

(ii) Try other version of the binomial tree such as the Rendleman and Bartter (1979) tree.

Option Price Tree

5.2 Follow the instructions on the web and declare a vector *value_tree* into which we will place our option price values.

5.3 Using a function to represent the payoff for a particular option, fill in the final values of the tree:

```
// assign payoff at final step
// move up through j steps where n is last step in tree
```

```
for(int j=0;j<value_tree[n].size();j++)
{
  value_tree[n][j] = /* fill this in */
}
```

5.4 Now write *for* loops to move backwards through the vector array caculating the value of the option at each node.

5.5 Print out the tree to screen (with n=3) and compare to the simple example in the lectures to check your code is working.

5.6 If your values don't match - try to work out why!!

5.7 Now print out the value at (0,0), the first step in your tree that represents the option value. Do the results look feasible? Compare them against the exact values from the formula.

Advanced Coding

5.8 Write a code storing just two time-levels, and compare (at every stage if needed) with the previous code.

5.9 Is it possible to store just one time-level? Try to write a code for this.

5.10 Do you notice any difference (time taken for computation) between the codes with different storage requirements?

Problem:

We wish to integrate the function:-

$$f(S;X) = \begin{cases} 0 & \text{if} \quad S < X \\ 1 & \text{if} \quad S \geq X \end{cases}$$

between $S = S_0$ and $S = S_{\max}$. We can numerically approximate the integral using the midpoint rule as

$$\int_{S_0}^{S_{\max}} f(S;X)dS = \sum_{i=0}^{n-1} f(S_0 + (i+0.5)\Delta S;X)\Delta S \tag{1}$$

where $\Delta S = (S_{\max} - S_0)/n$. The initial test parameters are $S_0 = 0.213$, $S_{\max} = 1.376$ and $X = 0.986$.

5.11 Download the example code from the website:
http://www.maths.man.ac.uk/ pjohnson/CompFin/Codes/convegence-example.cpp
Complete the code by adding in extra columns to output the exact value, the error ($\phi - \phi^*$) and the absolute error ($|\phi - \phi^*|$). Analyse your output using a plotting package (such as excel).

5.12 Can you estimate the convergence rate?

5.13 Change the supplied parameters to $S_0 = 0$, $S_{\max} = 2$ and $X = 1$. Why does the graph look different? Why do the errors sometimes appear to be negative?

5.14 How might you ensure that convergence to the solution is smooth? Think about the ways mentioned in the lecture [see Heston and Zhou, 2000, for applications in Binomial trees].

References

S. Heston and G. Zhou. On the Rate of Convergence of Discrete-Time Contingent Claims. *Mathematical Finance*, 10(1):53–75, 2000.