Course work Part A for IF2102 Financial Modeling with Visual Basic Year 2012-13 Term 2

Each question below is worth 5% weight, so total of 30% module weight

For each of the following questions please write separate subroutine named after the problems e.g. sub problemone() sub problemtwo() etc. Within each subroutine you can call any functions you have written that may be named differently.

1. Create a one period binomial tree of Stock Prices. Allow the user to inputs explained in class viz. current stock price S, future stock prices Su and Sd, probability of up move p, future value R of one unit of currency at time zero. Take these inputs from the spreadsheet and output a tree of prices as well as the discounted expected future stock price which for now we denote by X.

2. Then rewriting the formula to solve for p which makes X=S, write a separate VBA function that computes the desired probability. We will denote this special value of probability by q, the risk neutral probability. Name this function ComputeQ()

3. Now replace inputs Su and Sd by inputs u and d, the factors which determine the stock price appreciation and depreciation (rather than actual prices themselves).Drop the input p as it is no longer relevant - we determine q and use that instead. Write a function that computes the discounted expectation of future stock prices.Discounting is using R and expectation is using q. Future asset prices are passed as parameters. Name this function ComputePrice() Using this function, price a European call option on the above stock with a user determined strike price K.

4. Extend the above subroutine to incorporate two periods instead of one, introducing more data variables, and functions written for the 1 period tree. Do not use arrays or for loops yet - we do that later. You need to make a recombining two period Binomial trees one for stock and the other for option prices. Use constant q,u,d,R i.e. they do not change with time or position in the tree. As explained in class, compute European call option prices first at the end of 1 period, then use these newly computed prices to back out the option price at time 0.

5. Extend the solution to question 4 above to write a new subroutine for three periods using static arrays and for loops. Navigate along the tree to move backwards in time computing option prices at each stage until you reach time zero and hence solve for the current option price. Show both stock and option price trees on the spreadsheet. Then extend the solution to question 5 above to write a subroutine for n periods, where n is determined by the user and the arrays are dynamic (use Redim command).

6. For a given maturity say 256 days, use smaller and smaller time steps (larger and larger n) and show that the option price converges. For this part the user does not input n. You generate a sequence of n values . For this question do let the user decide u, d and R per day.

For each question above 40% marks allotted to that question will be assigned to clarity of design and code. This includes choice of variable names, flow of control, comments etc. Your code should work but it should also be very easily readable. There is no explicit credit for faster execution time; you are encouraged to write a good program that may run slower but is almost self-explanatory rather than a cryptic piece of code that is highly efficient but hard to read.