FE 621: HW2

Due date: March 4th at 11:59 pm

- Each problem is worth 40 points but 100 points is considered a full score for this assignment.
- Please make sure that your solutions are presented in an organized and readable manner.

Problem 1

(a) Write code that takes S_0, K, σ, r, T and N as inputs and computes the prices of options using an additive N-step binomial tree. Your code should be able to handle European and American call and put options. For the additive tree, use the parametrization described in class and also given in Eq. (2.19) in Clewlow/Strickland.

(b) Analyze the convergence of the binomial tree prices to the Black-Scholes options prices as the number of steps gets larger. Specifically, for the four types of options, plot the difference between the Black-Scholes price P^{BS} and the N-step binomial tree price P_N^{Tree} for different values of N. Comment on your findings.

To compute P^{BS} you can use the Black-Scholes formula for European options and for American options you can use an online option calculator of your choice. As reference parameters you can use the parameters in Figure 2.8 of Clewlow/Strickland, but feel free to experiment with different parameter values.

(c) *Extra credit:* Visualize the exercise barrier implied by the binomial tree for an American put option.

Remark: Figures 2.10 and 2.12 in Clewlow/Strickland contain pseudocode for pricing options using an additive binomial tree. To save storage space this code uses a single one-dimensional array to store the asset prices.

This trick is also used in Listing 3.2 in `https://epubs.siam.org/doi/pdf/10.1137/S0036144501393266` which shows a Matlab implementation of a multiplicative binomial tree (Listings 3.3-3.5 show implementations that replace loops by array operations in order to speed up the code - loops are generally slow in Matlab/Python/R).

While a one-dimensional array uses less memory, it may be useful to use a two-dimensional array to store the asset prices when implementing a binomial tree for the first time.

Problem 2

(a) Write code that takes S_0, K, H, σ, r, T and N as inputs and uses an N-step (multiplicative) binomial tree to price a European Up-and-Out call option with barrier H.

(b) A closed-form formula exists for the price of the option in part (a). See for example Equation (5.2) in `https://www.math.kth.se/matstat/seminarier/reports/K-exjobb09/090601a.pdf`. Analyze the convergence of the tree-price to the exact price in the same way as in Problem 1(b). As reference parameters you can use $S_0 = 10$, $K = 10$, $T = 0.3$, $\sigma = 0.2$, $r = 0.01$ and $H = 11$, but feel free to experiment with different parameter values.

(c) Provide a numerical recipe for using a binomial tree to price an American Up-and-In call option. Clearly explain the key steps in your backward dynamic programming procedure. In this problem you do not need to write any code - just provide a description of your computational procedure.

Remark: Computing the price of an American Up-and-Out call is easier, but the In-Out parity does not hold for American options. That is, a portfolio of Up-and-In and Up-and-Out call options is not equivalent to a standard American call option. For example, the holder of such a portfolio may exercise the Out-option when the stock price approaches the barrier, and exercise the In-option after the price crosses the barrier. This would result in a payoff different from the payoff of the standard American call option.

Problem 3

Installment options are such that the option premium is paid in installments, and at each installment date, the option holder has the right to terminate the contract. This way they only face the risk of losing the installments already paid rather than the entire option premium.

Consider an installment call option with strike price K, maturity T, and installment dates 0, T/n, $2T/n$, $\ldots$, $(n-1)T/n$, for some n. At each of these dates, the holder can choose to pay an installment p or to terminate the contract. At time T, the holder receives $(S_T - K)^+$ if all installments have been paid.

(a) Write a function that takes p as input and uses a (multiplicative) binomial tree to compute the value $V_0(p)$ of an installment call option. Use parameters $S_0 = 100$, $\sigma = 0.20$, $r = 0.04$, $K = 90$, $T = 1$, and $n = 4$.

Note: The value of the option at a given time is the discounted value of all future cash flows, i.e., after the current installment has been paid. In particular, $V_0(p)$ *is the value of being long the option at time 0, i.e., after paying the first installment.*

(b) The arbitrage-free premium of an installment option is the value of p such that $V_0(p) = p$. Plot $V_0(p)$ as a function of p and find the arbitrage-free value of p.

Extra credit: Provide justification for why the equation $V_0(p) = p$ has a uniquely determined solution.

(c) Price a standard European call option using the binomial tree and the same parameters as in part (a). How does its price compare to the sum of the discounted installment payments $p\sum_{i=0}^{n-1} e^{-ri\frac{T}{n}}$? Report both values and explain your findings.