EN.550.443 Financial Computing in C++

Huachen Qin

1. **Problem 1**

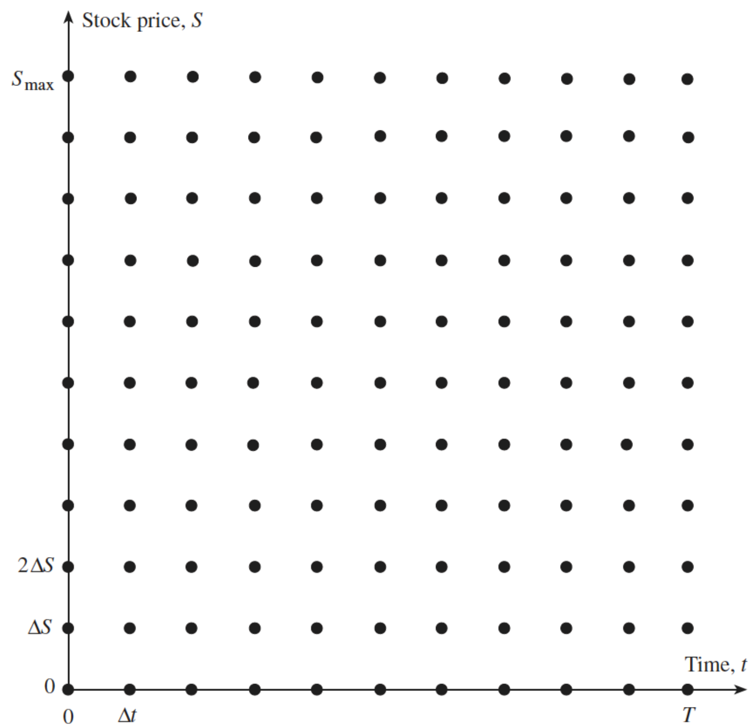
a) $v(T_n-, S, k) = 0, when\ k = 0;$
   $v(T_n-, S, k) = (S - K)^+, when\ k \geq 1.$

b) $v(T_j-, S, k) = maximum\{v(T_j, S, k), (S - K)^+ + v(T_j, S, k - 1)\}.$

c) **Comment**:

For this question, we solving the problem by Finite Difference Method and Dynamic Programming Method. Swing option is very similar to American option in that it has limited early exercises opportunities, in order to calculate the prices of swing options, we use the following method:

Consider m possible exercises date $\{t_1, t_2, \dots, t_m\}$ out of total excises date $\{T_1, T_2, \dots, T_n\}$ provided that $m \leq n$, then we use Finite Difference Method to create the following grid:



We divide stock price S by M steps and time T by N steps, for boundary conditions, we set: $v(0, m) = 0, v(N, m) = S.$

Then apply Crank Nicolson Method (central difference method) to calculate $v(n, m), v(n, m + 1), v(n, m - 1)$ based on $v(n + 1, m), v(n + 1, m + 1), v(n + 1, m - 1)$, if we arrive at the possible exercise date , there are two things that we should consider: whether or not to exercise, and the number of exercise rights k.

Then we need to consider:

$$v(T_j-, S, k) = maximum\{v(T_j, S, k), (S - K)^+ + v(T_j-, S, k - 1)\}$$

According to this method, we calculate $v(0{,}50{,}2)$ with 2 possible exercise dates from $\left\{T_1 = \frac{1}{3}, T_2 = \frac{2}{3}, T_3 = 1\right\}$, we use Backward Dynamic Programming Method. The chart is as follows:

Chart 1: Possible Exercise Scenarios

	Number of Possible Exercises	Possible Exercise Date
At Time T3	k = 2	T3
	k=1	(T1,T3) or (T2,T3)
	k=0	(T1,T2)
At Time T2	k=2	(T2,T3)
	k=1	(T1,T2) or (T2,T3)
At Time T1	k = 2	(T1,T3), (T1, T3) or (T2,T3)

Consider possible combinations of exercises, there are total 3 combinations: $\{T_1, T_2\}, \{T_1, T_3\}, \{T_2, T_3\}$, use Crank-Nicolson method described above and compare the payoff with early exercises payoff to calculate the swing option prices corresponding to 3 combinations, and define the maximum of these prices to be the swing option price.

d) No. There cannot be a closed form solution to this problem. Because it is never optimal to exercise an American option on a non dividend paying stock before maturity, so the optimal exercise date should always include the maturity time. Hence there are $\binom{n-1}{m-1}$ possible early exercise dates combinations to consider, which are finite. Also the number of possible exercises $k \geq 1$ always holds at the maturity time $T = T_n$, swing option is very similar to American option, which do not have closed form solution. Hence at each of these finite possible exercise dates, we need to compare the value to decide whether or not to exercise. Hence there should not be closed form solution to early exercise.

2. Problem 2

a) Methodology:

In this question, we need to calculate price of caplet based on Hull White Model. In order to do that, we implement our programming according to the following algorithm:

I. Set up all the parameters, $T = 5, \delta = 0.25, then\ Steps = \frac{T}{\delta} = 20$. In that case, the time interval of the tree matches the term structure of caplet.

II. Calculate zero coupon bond price $B(t, t+\delta), t = T_0, T_1, \dots, T_{steps}$. To calculate that, we first run the Hull White Tree as in Homework 11, we have the instantaneous rate tree from Hull White Tree(in the c++ program we use *TheTree[i][j]*), then we can define $B(t, t+\delta) = \exp\left(-TheTree[i][j] * \Delta t\right)$.

III. Set up LIBOR tree, and its structure should match the Hull White Tree structure. The formula for LIBOR Tree is:

$$L(t, t+\delta) = \frac{1 - B(t, t+\delta)}{\delta B(t, t+\delta)}$$

IV. Then we can calculate the instantaneous payoff from each of the nodes on the LIBOR Tree. The formula for instantaneous payoff is:

$$payoff = N\delta(L(t, t+\delta) - K)^+$$

V. Use backwards method to calculate cumulative payoff and discount it back. The formula for cumulative payoff is:

$$cumulative_{payoff}[i][j] = payoff[i][j] + \exp(-TheTree[i][j] * \Delta t)\big(pu * cumulative_{payoff}[i+1][j+2] + pm * cumulative_{payoff}[i+1][j+2] + pd * cumulative\_payoff[i+1][j]\big)$$

Also consider the fact that pruning from the Hull White Tree, so $j_{min} \leq j \leq j_{max}$.

VI. Finally we have $cumulative_{payoff}[0][0]$ as the expected payoff from caplet at time 0, hence that should be the no-arbitrage price of the caplet.

3. Acknowledgement

a) For problem 1, I reuse the Matrix class from Homework 9 to solve the matrix-form linear equations.

b) For problem 2, I reuse the Hull White class on blackboard to do further calculations.