1^{st} Assignment - Lab Report

Numerical Methods for Finance

Fabio Caironi, Murat Aydin

February 2021

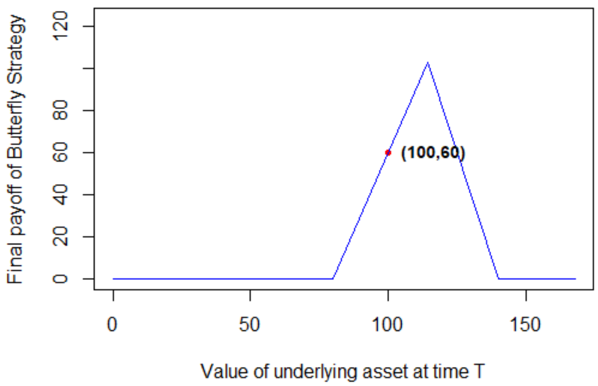
1 Exercise 1

The final payoff of this portfolio is a linear combination of the payoffs of the single options, therefore:

$$\begin{aligned}\Pi_T(S_T) &= j\max\{S_T - K_1, 0\} - (j+h)\max\{S_T - \overline{K}, 0\} + h\max\{S_T - K_2, 0\} \\ &= \begin{cases} 0 & \text{if } S_T \leq \min\{K_1, K_2\} \\ j(S_T - K_1) & \text{if } \min\{K_1, K_2\} < S_T \leq \overline{K} \\ h(K_2 - S_T) & \text{if } \overline{K} < S_T < \max\{K_1, K_2\} \\ 0 & \text{if } S_T \geq \max\{K_1, K_2\} \end{cases}\end{aligned}$$

The above definition of Π_T always holds true since the convex combination $\overline{K}$ satisfies $\min\{K_1, K_2\} \leq \overline{K} \leq \max\{K_1, K_2\}$.

To give a representation of the payoff function, calling `butterfly(j=3, h=4, S=100, K1=80, K2=140)` will yield the following payoff chart:



2 Exercise 2

Given S_0, K, t_0, T, u, d and r (discrete compound regime), the requested outputs are:

$$P_{t_0} = \frac{1}{(1+r)^{T-t_0}} \left[q_u \cdot \max\{K - S_0 u, 0\} + q_d \cdot \max\{K - S_0 d, 0\} \right]$$

$$q_u = \frac{(1+r)^{T-t_0} - d}{u - d} \qquad , \qquad q_d = 1 - q_u$$

$$y^* = \frac{1}{S_0} \left(\frac{\max\{K - S_0 u, 0\} + \max\{K - S_0 d, 0\}}{u - d} \right)$$

Exploiting the put-call parity formula, the price of a Call at time t_0 is:

$$C_{t_0} = S_0 + P_{t_0} - \frac{K}{(1+r)^{T-t_0}}$$

Finally, Merton constraints for C_{t_0} are satisfied if and only if:

$$\max\left\{ S_0 - \frac{K}{(1+r)^{T-t_0}}, 0 \right\} \leq C_{t_0} \leq S_0$$

3 Exercise 3

The function `multi_binom` takes advantage of a matrix representation of the three trajectory trees (Value of Underlying Asset, Price, Payoff), where the matrix is upper triangular, its rows are possible scenarios and its columns are time points. In particular, an element of position `(i,j)` in any of the trajectory matrices is associated with a scenario of `i-1` down-movements and `i-j` up-movements.

The algorithm is composed of:

- Construction of the underlying asset value tree, whose element in a scenario at time t_m with h up movements is $S_{t_m}(h) = S_0 u^h d^{m-h}$.

- Finding of the pricing tree through backward induction up to time t_0.

  Initialization: $P_T(h) = P_T^A(h) = \max\{K - S_{t_N}(h), 0\}$

  Iteration:

  - European Put: $P_{t_m}(h) = \frac{1}{(1+r)^{\Delta t}} \cdot \Big[P_{t_{m+1}}(h+1) \cdot q_u + P_{t_{m+1}}(h) \cdot q_d\Big]$
  - American Put: $P_{t_m}^A(h) = \max\Big\{P_{t_m}(h), \max\{K - S_{t_m}(h), 0\}\Big\}$

- Finding of the payoff tree, which is:

  - European Put: $\Pi_{t_m}(h) = \begin{cases} 0 & \text{if } t_m < T \\ \max\{K - S_{t_N}(h), 0\} & \text{if } t_m = T \end{cases}$
  - American Put: $\Pi_{t_m}^A(h) = \max\{K \cdot D(t,T) - S_{t_m}(h), 0\}$

As theory suggests, the final payoff for an American Put Option is never lower than that of the European relative ad this is true in particular for an early exercise decision. Our function called on the same parameters except `European=TRUE` / `European=FALSE` brings evidence of the latter result.

4 Exercise 4

We exploit the pricing formula for an option in a Binomial Biperiodal Model. First, we have to compute A_{uu} (Payoff with 2 ups), $A_{ud} = A_{du}$ (Payoff with 1 up and 1 down) and A_{dd} (Payoff with 2 downs).

$$\begin{aligned}
A_{uu} &= \max\left\{S_{uu} - \frac{1}{3}(S_0 + S_u + S_{uu}), 0\right\} = \frac{1}{3}S_0 \cdot \max\{(2u+1)(u-1), 0\} \\
A_{ud} &= \max\left\{S_{ud} - \frac{1}{3}(S_0 + S_u + S_{ud}), 0\right\} = \frac{1}{3}S_0 \cdot \max\{(2ud - u - 1), 0\} \\
A_{du} &= \max\left\{S_{ud} - \frac{1}{3}(S_0 + S_d + S_{ud}), 0\right\} = \frac{1}{3}S_0 \cdot \max\{(2ud - d - 1), 0\} \\
A_{dd} &= \max\left\{S_{dd} - \frac{1}{3}(S_0 + S_d + S_{dd}), 0\right\} = \frac{1}{3}S_0 \cdot \max\{(2d+1)(d-1), 0\}
\end{aligned}$$

Finally, the price at time 0 is the discounted expectation of A_T via $Q = (q_u, q_d)$:

$$A_0 = \frac{1}{(1+r)^{T-t_0}} \left[q_u^2 A_{uu} + q_u q_d A_{ud} + q_u q_d A_{du} + q_d^2 A_{dd} \right]$$

The call `price_asian_binomial(S0=1,u=2,d=0.5,r=0,t0=0,maturity=1)` returns $0.\overline{2}$.