TD8

December 2022

1 American option in the binomial model

In addition to pricing European options, we want to include the ability to price American options in the binomial model.

The holder of an American option has the right to exercise it at any time up to and including the expiry date N. If the option is exercised at time step n and node i of the binomial tree, then the holder will receive payoff $h\left(S\left(n,i\right)\right)$.

The price $H\left(n,i\right)$ of an American option at any time step n and node i in the binomial tree can be computed by the following procedure, which proceeds by backward induction on n:

- At the expiry date N: $H\left(N,i\right)$ has the same value as for the option's European counterpart. Financial interpretation: if not exercised before the expiry, there is no advantage holding an American option over holding a European option.
- If $H\left(n+1,i\right)$ is already known at each node $i \in \{0,\cdots,n+1\}$ for some $n < N$, then

  $$H\left(n,i\right) = \max\left\{\underbrace{\frac{qH\left(n+1,i+1\right)+\left(1-q\right)H\left(n+1,i\right)}{1+R}}_{\text{continuation value}}, \underbrace{h\left(S\left(n,i\right)\right)}_{\text{intrinsic value}}\right\}$$

  for each $i \in \{0,\cdots,n\}$.
  Financial interpretation: the option holder chooses the maximum between the continuation value (expected gain if they do not exercise, under the risk-neutral measure) and the intrinsic value (the value of the option if exercised immediately).

In particular, $H\left(0,0\right)$ at the root node of the tree is the price of the American option at time 0.

We would like to compute and store the price of an American option for each time step n and node i in the binomial tree. In addition, we want to compute the early exercise policy, which should be of Boolean type and tells if the American option should be exercised or not for each state of the tree. The early exercise policy should also be stored using the class BinLattice.

Programming: extend your Binomial tree class to price American options as well.

2 Black-Scholes as limit of the binomial tree

The binomial model can be used to approximate the Black-Scholes model if N is large.

One of the scheme is to divide the time interval $[0, T]$ into N steps of length $h = \dfrac{T}{N}$, and set the parameters of the binomial model to be:

$$\begin{aligned} U &= e^{\left(r+\frac{\sigma^2}{2}\right)h+\sigma\sqrt{h}} - 1, \\ D &= e^{\left(r+\frac{\sigma^2}{2}\right)h-\sigma\sqrt{h}} - 1, \\ R &= e^{rh} - 1, \end{aligned}$$

where σ is the volatility and r is the continuously compounded interest rate in the Black-Scholes model.

Implement a method to initialize a Binomial tree as a Black-Scholes approximation (using the Black-Scholes parameters). Compare option prices with the Monte Carlo method and the closed form method for European options.

3 Programming

1. Augment the *Option* class with *bool isAmericanOption()* which returns false in its non-overriden version
2. Derive *Option* into *AmericanOption*, and override *isAmericanOption()* properly
3. Derive *AmericanOption* into *AmericanCallOption* and *AmericanPutOption*, write proper constructors and override their respective *payoff()* methods
4. Modify *CRRPricer* in order for it to price properly American options; the exercise condition for American options is stored in a *BinaryTree<bool>*, accessible through a getter method *bool getExercise(int, int)*.
   The exercise condition is true when the intrinsic value is larger or equal to the continuous value, it is computed during the CRR procedure.
5. Overload the CRRPricer with *CRRPricer(Option* option, int depth, double asset_price, double r, double volatility)*, which initializes U, D and R as described in Section 2.