

FE545 Homework Assignment #3

Due Date: April 6th (Saturday).

Problem 1: The trinomial option pricing model is an option pricing model incorporating three possible values that an underlying asset can have in one time period. The three possible values the underlying asset can have in a time period may be greater than, the same as, or less than the current value. The trinomial model uses an iterative procedure, allowing for the specification of nodes, or points in time, during the time span between the valuation date and the option's expiration date.

The trinomial option pricing model, proposed by Phelim Boyle in 1986, is considered to be more accurate than the binomial model, and will compute the same results, but in fewer steps. However, the trinomial model has not gained as much popularity as the other models.

The trinomial option pricing model differs from the binomial option pricing model in one key aspect by incorporating another possible value in one time period. Under the binomial option pricing model, it is assumed that the value of the underlying asset will either be greater than or less than, its current value.

The trinomial model, on the other hand, incorporates a third possible value, which incorporates a zero change in value over a time period. This assumption makes the trinomial model more

relevant to real life situations, as it is possible that the value of an underlying asset may not change over a time period, such as a month or a year.

Under the trinomial method, the underlying stock price is modeled as a recombining tree, where, at each node the price has three possible paths: an up, down and stable or middle path. We will use a trinomial tree model defined by

$$S(t + \Delta t) = \begin{cases} S(t)u & \text{with probability } p_u \\ S(t) & \text{with probability } 1 - p_u - p_d \\ S(t)d & \text{with probability } p_d \end{cases} \quad (1)$$

$$\begin{aligned} \mathbb{E}[S(t_{i+1})|S(t_i)] &= e^{r\Delta t}S(t_i) \\ \text{Var}[S(t_{i+1})|S(t_i)] &= \Delta t S(t_i)^2 \sigma^2 + \mathcal{O}(\Delta t) \end{aligned} \quad (2)$$

Condition 2 is a standard equilibrium or no arbitrage condition: it states that the average return from the asset should be equal to the risk free-return. It can be re-written in the following explicit form:

$$1 - p_u - p_d + p_u u + p_d d = e^{r\Delta t}$$

Conditions (2, 4) impose two constraints on 4 parameters of the tree. An extra constraint comes from the requirement that the size of the upward jump is the reciprocal of the size of the

downward jump,

$$ud = 1 \quad (3)$$

While this condition is not always used for a trinomial tree construction, it greatly simplifies the complexity of the numerical scheme: it leads to a recombining tree, which has the number of nodes growing polynomially with the number of levels rather than exponentially.

Given the knowledge of jump sizes u , d and the transition probabilities p_u , p_d it is now possible to find the value of the underlying asset, S , for any sequence of price movements. A directed graph with nodes labeled by asset prices and edges connecting nodes separated by one time step and a single price jump is the desired trinomial pricing tree, see Fig. 1.

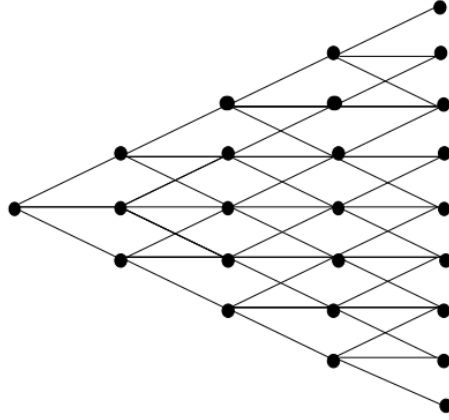


Figure 1: Trinomial Tree

Let us define the number of up, down and middle jumps as N_u , N_d , N_m , respectively, and so the value of the underlying share price at node j for time i is given by

$$S_{i,j} = u^{N_u} d^{N_d} S(t_0), \text{ where } N_u + N_d + N_m = N \quad (4)$$

We have imposed three constraints (1, 2 and 3) on four parameters u , d , p_u and p_m . As a result there exists a family of trinomial tree models. In the project we consider the following popular representative of the family: its jump sizes are

$$\begin{aligned} u &= e^{\sigma\sqrt{2\Delta t}} \\ d &= e^{-\sigma\sqrt{2\Delta t}} = \frac{1}{u} \\ m &= 1 \end{aligned} \quad (5)$$

and the corresponding probabilities are:

$$\begin{aligned} p_u &= \left(\frac{e^{(r-q)\Delta t/2} - e^{-\sigma\sqrt{\Delta t/2}}}{e^{\sigma\sqrt{\Delta t/2}} - e^{-\sigma\sqrt{\Delta t/2}}} \right)^2 \\ p_d &= \left(\frac{e^{\sigma\sqrt{\Delta t/2}} - e^{(r-q)\Delta t/2}}{e^{\sigma\sqrt{\Delta t/2}} - e^{-\sigma\sqrt{\Delta t/2}}} \right)^2 \\ p_m &= 1 - p_u - p_d \end{aligned} \quad (6)$$

where Δt is the length of time per step in the tree and is simply time to maturity divided by the number of time steps; r is the

risk-free interest rate over this maturity; σ , is the corresponding volatility of the underlying; q is its corresponding dividend yield.

Note that for p_u , p_d , and p_m to be in the interval $(0, 1)$ the following condition on Δt has to be satisfied $\Delta t < 2\frac{\sigma^2}{(r-q)^2}$. Once the tree of prices has been calculated, the option price is found at each node largely as for the binomial model, by working backwards from the final nodes to the present node (t_0). The difference being that the option value at each non-final node is determined based on the three - as opposed to two - later nodes and their corresponding probabilities.

European call (put) option is a contract which gives its owner the right to buy (sell) an agreed asset (underlying asset) at the agreed price K (strike price) at the specified time T (maturity time). The methodology when pricing options using a trinomial tree is exactly the same as when using a binomial tree. Once the share price tree is built, and the option payoffs at maturity time T are calculated:

$$C(S_T, T) = \max(S_T - K, 0) \text{ (European call option)}, \quad (7)$$

$$C(S_T, T) = \max(K - S_T, 0) \text{ (European put option)}. \quad (8)$$

After that it remains to apply the following backward induction algorithm, where n represents the time position and j the space position:

$$C_{n,j} = e^{-r\Delta t} [p_u C_{n+1,j+1} + p_m C_{n+1,j} + p_d C_{n+1,j-1}] \quad (9)$$

Note that the American call (put) option always must be worth at least $C(S_t)$ since it can be exercised at any time prior to the expiry date. This makes it more interesting and complex to evaluate than a European option which can only be exercised at the expiry date. Since an American option can be exercised at any time prior to the expiry date, choosing the optimal time to exercise is a crucial problem.

$$C(S_t, t) = \max(S_t - K, 0) \text{ (American call option)}, \quad (10)$$

$$C(S_t, t) = \max(K - S_t, 0) \text{ (American put option)}. \quad (11)$$

The backward induction algorithm can be derived from the risk-neutrality principle and is the same for put and call options. When applied in the context of a trinomial tree (using the exact same methodology as the binomial tree), we can calculate the option value at interior nodes of the tree by considering it as a weighting of the option value at the future nodes, discounted by one time step.

Thus we can calculate the option price at time n , C_n , as the option price of an up move $p_u C_{n+1}$ plus the option price of the middle move by $p_m C_{n+1}$ plus the option price of a down move

by $p_d C_{n+1}$, discounted by one time step, $e^{-r\Delta t}$. So at any node on the tree our backward induction formula (9), is applied to give us the option prices at any node in the tree.

The name of the algorithm should now be clear since we only need to value the option at maturity, i.e the leaf nodes, and then work our way backwards through the tree calculating option values at all the nodes until we reach the root S_0, C_0 .

Overall, your option tree pricer should be able to price both the European options and American options include the following files:

- Arrays.h
- Arrays.cpp
- BinomialTree.h
- BinomialTree.cpp
- TrinomialTree.h
- TrinomialTree.cpp
- Normals.h
- Normals.cpp
- Parameters.h
- Parameters.cpp

- PayOffBridge.h
- PayOffBridge.cpp
- PayOff3.h
- PayOff3.cpp
- TreeEuropean.h
- TreeEuropean.cpp
- TreeAmerican.h
- TreeAmerican.cpp
- TreeProducts.h
- TreeProducts.cpp
- TreeMain1.cpp

Please use the following parameter and generate European and American (Put and Call) option prices:

$$\begin{aligned}
 T &= 1 \\
 S(T) &= 50 \\
 \sigma &= 0.30 \\
 r &= 0.05 \\
 N &= 100
 \end{aligned}
 \tag{12}$$

Please use the set of classes provided for binomial tree European and American option pricing program, and develop a trinomial European and American option pricing program. Please note that the trinomial tree method is based on Boyle (1986), and it is different from the binomial tree method we introduced in the class. But the software design should be similar. Price the European and American (put and call) options using the set of parameters given below. Produce tabulated output of your answers.

Homework Honor Policy: You are allowed to discuss the problems between yourselves, but once you begin writing up your solution, you must do so independently, and cannot show one another any parts of your written solutions.