Implied Binomial Tree

Chen Jiahao A0207304U
Dai Peijie A0207134R

Derivation Σ

When k is even, we have known the explicit expression of iteration upward formula and its summation part Σ. Using the same derivation process, we can get the Σ for downward formula:

$$\Sigma = \sum_{j=i+1}^{k-1} \lambda_{k-1,j}\left(S_{k-1,i} - F_{k-1,j}\right)$$

When k is odd, the last one constraint changes to $S0_0^2 = S_{k,\left[\frac{k}{2}\right]} \times S_{k,\left[\frac{k}{2}\right]+1}$. Adding this constraint to formula (13), we get

$$\sum_{j=0}^{i-1} \lambda_{k-1,j}\left(F_{k-1,j} - S_{k-1,i}\right) \quad i = floor(\frac{k}{2})$$

Algorithm

After we have all explicit expressions of Σ, we can construct the implied binomial tree. We write a function called GetImpVol to get implied volatility. After given strike and maturity, this function firstly searches the interval including maturity and do linear interpolation to maturity, and then use cubic spline to interpolate strike. Finally, we can get the implied volatility to calculate the call and put option price and then construct the tree.

The Algorithm is showed below:
Set basic parameters S_0, risk-free rate r, strike k, dividend q, maturity T, number of interval N
#Forward Induction Part to Generate Implied Tree
Set probability matrix p, price λ, stock price S and forward price F
#Set the middle node when k is even and the first two nodes when k is odd
for (int k = 1; k <= N; k++)
    if k is even
        set $S_{k,k/2} = S_0$
        set index_1 = k/2 -1
        set index_2 = k/2
    else
        set $S_{k,\lfloor k/2 \rfloor} = \frac{S_0\left[e^{r\Delta t}C(S_0,k\Delta t) + \lambda_{k-1,\lfloor k/2 \rfloor}S_0 - \Sigma\right]}{\lambda_{k-1,\lfloor k/2 \rfloor}F_{k-1,\lfloor k/2 \rfloor} - e^{r\Delta t}C(S_0,k\Delta t) + \Sigma}$

set $S_{k,[k/2]+1} = S_0^2/S_{k,[k/2]}$

set index_1 = floor(k/2) - 1

set index_2 = floor(k/2) + 1

#Upward Iteration and Downward Iteration

for (int i = index_1; i >= 0; i--)

get impvol = GetImpVol(impvol, Maturity, stock_, strike) #GetImpVol is a self-defined

$$S_{k,i} = \frac{S_{k,i+1}\left(e^{r\Delta t}C(S_{k-1,i},k\Delta t)-\Sigma\right)-\lambda_{k-1,i}S_{k-1,i}(F_{k-1,i}-S_{k,i+1})}{e^{r\Delta t}C(S_{k-1,i},k\Delta t)-\Sigma-\lambda_{k-1,i}(F_{k-1,i}-S_{k,i+1})}$$

for (int i = index_2; i < k; i++)

get impvol = GetImpVol(impvol, Maturity, stock_, strike)#GetImpVol is a self-defined

$$S_{k,i+1} = \frac{S_{k,i}\left(e^{r\Delta t}C(S_{k-1,i},k\Delta t)-\Sigma\right)-\lambda_{k-1,i}S_{k-1,i}(F_{k-1,i}-S_{k,i+1})}{e^{r\Delta t}C(S_{k-1,i},k\Delta t)-\Sigma-\lambda_{k-1,i}(F_{k-1,i}-S_{k,i+1})}$$

#Calculate forward, probability and lambda

$$F_{k-1,i} = S_{k-1,i}e^{(r-q)\Delta t}$$

$$P_{k,i} = \frac{F_{k-1.i}-S_{k,i+1}}{S_{k,i}-S_{k,i+1}}$$

$$\text{lambda} = \begin{cases} \lambda_{k,0} = e^{-r\Delta t}P_{k,0}\lambda_{k-1,0} \\ \lambda_{k,i} = e^{-r\Delta t}((1-P_{k,i-1})\lambda_{k-1,i-1} + P_{k,i}\lambda_{k-1,i}\,, \quad 1 \le i \le k-1 \\ \lambda_{k,k} = e^{-r\Delta t}\lambda_{k-1,k-1}(1-p_{k,k-1}) \end{cases}$$

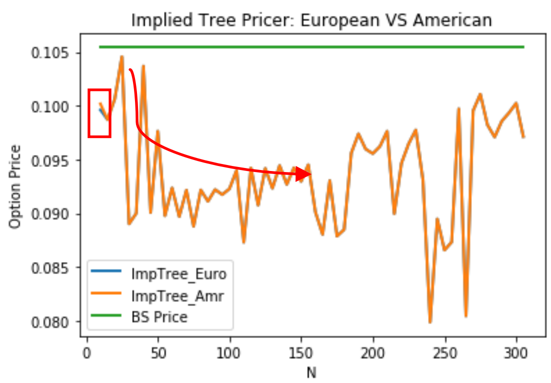
#Pricing Algorithm is the same as previous tree. So, we don't talk about it

Result

Question 1, Test Cases for Different Tradable:

We use implied tree to price European call option with s0 = 1.25, r = 0.01, strike = 1.2, q = 0, American call option with s0 = 1.25, r = 0.01, strike = 1.2, q = 0,, and down-and-out Barrier option with s0 = 1.25, r = 0.01, strike = 1.2, q = 0 and barrier H = 1.2.

Since we set q = 0, the vanilla European option price will be identical to the American option price. The two results are compared in the following graph:

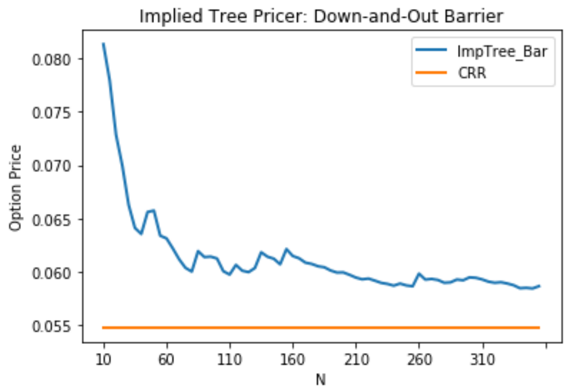


The implied tree price is compared with BS formula price with implied volatility 0.144395 which is interpolated by linear and cubic spline interpolation. From the graph above, we can find that the European option and American option price have totally overlapped, showing that our tree is reasonable.

However, the price is not as good as other pricing model like CRR or Binomial Tree. We consider that the deviation from BS formula results from following reasons: The BS formula assumes the price volatility is constant. However, in implied tree model, the stock price at each node is calculated by the iteration in which implied volatility is used to solve the equation. Each node and each time will have different implied volatility, making different risk-neutral probability. Apart from the volatility difference, at each node, there might be some stock price generating problems. When invalid price happens, we should use the approximation to fix it otherwise there are arbitrage opportunities. The last reason is implied tree model includes more market information, especially the market expectation. So, it is reasonable this pricer will have different price calculated by BS formula.

The implied tree model also has fluctuating feature. we can see that as N goes larger, the amplitude of the fluctuation will be higher. There is a convergence trend from small N to about N = 170 shown by the red arrow. As discretization steps go up, the tree can simulate more situation. It is obvious the tree should converge. However, when N is larger than 170, everything changed. We think this divergence problem is caused by the collective effect of interpolation error and approximation error. To be specific, when N is large, dt = T/N will be very small and far away from the first maturity T = 0.2 in which situation our interpolation might have negative implied volatility and this cannot happen in reality. Meanwhile, if N is large, the nodes number will increase too, leading to the probability of generating wrong stock price increase.

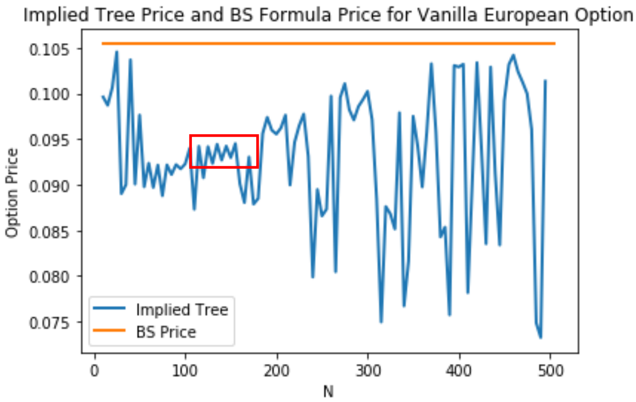
The result of down-and-out barrier option is:



The implied tree price is compared with CRR price with discretization steps = 1000 which converges to the true price. From the graph shown above, we can find the implied tree price declines since barrier option loses some chances to exercise. Meanwhile, the implied tree price is higher the CRR price.

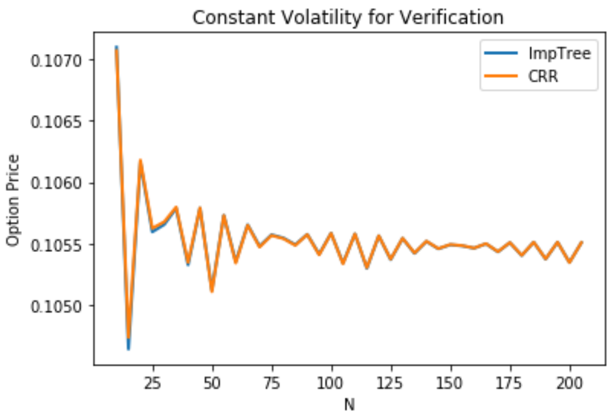
Actually, the smooth of the Barrier price surprises us. We think that the barrier will remove a lot of difference between traditional binomial tree and implied tree because all nodes under barrier will have zero value. Although barrier removes the difference, there are still some fluctuations.

Question 2, Appropriate Model Setting- Discretization Steps:
Our result is based on parameter value: s0 = 1.25, r = 0.01, strike = 1.2, q = 0. We use implied tree pricer and BS formula to price the European vanilla option from N = 10 to N = 500 and each step is 5. So, we got 100 prices pairs. The result is shown below:
From graph below we can find that the BS formula price is 0.1054. However, the implied tree price is extremely weird since it fluctuates all the time.



The implied tree price at the beginning has large fluctuating amplitude and from around N = 80 to N = 170, the price tends to be relatively flat. After N goes beyond 170, the price totally loses itself and goes crazy. So, we think the implied tree price needs N take values from around 120 to 160.

Question 3, Under Constant Volatility Situation, CRR vs Implied Tree Model:
In order to verify our algorithm, we use two models with constant volatility surface to price vanilla European option. From the graph we can clearly see that the implied tree pricer degenerate to CRR pricer, proving our algorithm is correctly implemented.
The parameter we use is the same as above.



Summary:

We conclude that the implied tree model has:

Advantages: Including more market information

Disadvantage: Unstable discretization steps, Approximation used