THE COMPLETE GUIDE TO Option Pricing Formulas

SECOND EDITION

ESPEN GAARDER HAUG

McGraw-Hill
New York Chicago San Francisco
Lisbon London Madrid
Mexico City Milan Seoul New Delhi San Juan
Singapore Sydney Toronto

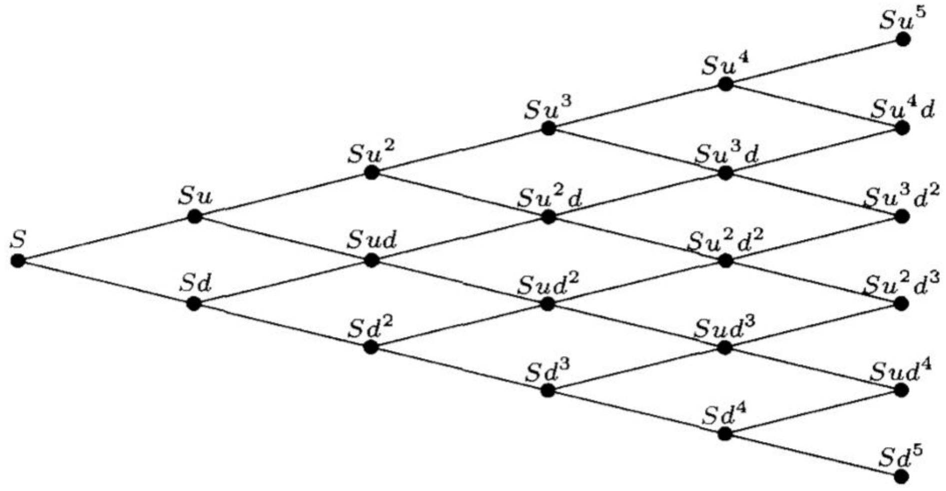
```
If TypeFlag = 1 Then 'Plain Vanilla
    BinomialPayoff = Max(z * (S - X), 0)
  ElseIf TypeFlag = 2 Then ' Power contract
    BinomialPayoff = S^pow
 ElseIf TypeFlag = 3 Then ' Capped Power contract
    BinomialPayoff = Min(S^pow, cap)
 ElseIf TypeFlag = 4 Then ' Power contract
    BinomialPayoff = (S / X)^pow
ElseIf TypeFlag = 5 Then ' Power contract
    BinomialPayoff = z * (S - X)^pow
ElseIf TypeFlag = 6 Then 'Standard power option
    BinomialPayoff = Max(z * (S^pow - X), 0)
 ElseIf TypeFlag = 7 Then 'Capped power option
    BinomialPayoff = Min(Max(z * (S^pow - X), 0), cap)
ElseIf TypeFlag = 8 Then ' Powered option
    BinomialPayoff = Max((z * (S - X)), 0)^pow
ElseIf TypeFlag = 9 Then ' Capped powered option
    BinomialPayoff = Min(Max((z * (S - X)), 0)^pow, cap)
 ElseIf TypeFlag = 10 Then ' Sinus option
    BinomialPayoff = Max(z * (Sin(S) - X), 0)
 ElseIf TypeFlag = 11 Then ' Cosinus option
    BinomialPayoff = Max(z * (Cos(S) - X), 0)
  ElseIf TypeFlag = 12 Then ' Tangens option
    BinomialPayoff = Max(z * (Tan(S) - X), 0)
 ElseIf TypeFlag = 13 Then ' Log contract
    BinomialPayoff = Log(S)
 ElseIf TypeFlag = 14 Then ' Log contract
    BinomialPayoff = Log(S / X)
 ElseIf TypeFlag = 15 Then ' Log option
    BinomialPayoff = Max(Log(S / X), 0)
 ElseIf TypeFlag = 16 Then 'Square root contract
    BinomialPayoff = Sqr(S)
ElseIf TypeFlag = 17 Then 'Square root contract
    BinomialPayoff = Sqr(S / X)
ElseIf TypeFlag = 18 Then 'Square root option
    BinomialPayoff = Sqr(Max(z * (S - X), 0))
End If

End Function
```

7.1.1 Cox-Ross-Rubinstein American Binomial Tree

Here we will look at how to use the Cox-Ross-Rubinstein binomial tree to value American-style options. The asset price at each node is set equal to

$$Su^i d^{j-i}, \qquad i = 0, 1, \ldots, j,$$

where the up and down jump factors for a time interval $\Delta t = T/n$ is given by (7.5), where n is the number of time steps, as before. The probability of the stock price increasing by the factor u is now given by equation (7.6). Since probabilities must sum to unity, the probability of the stock price decreasing by the factor d must be $1 - p$. Again, the up and down factors and probabilities are chosen to match the first two moments of the stock price distribution. This ensures that the probability distribution implied by the binomial tree converges to geometric Brownian motion when Δt goes to zero.



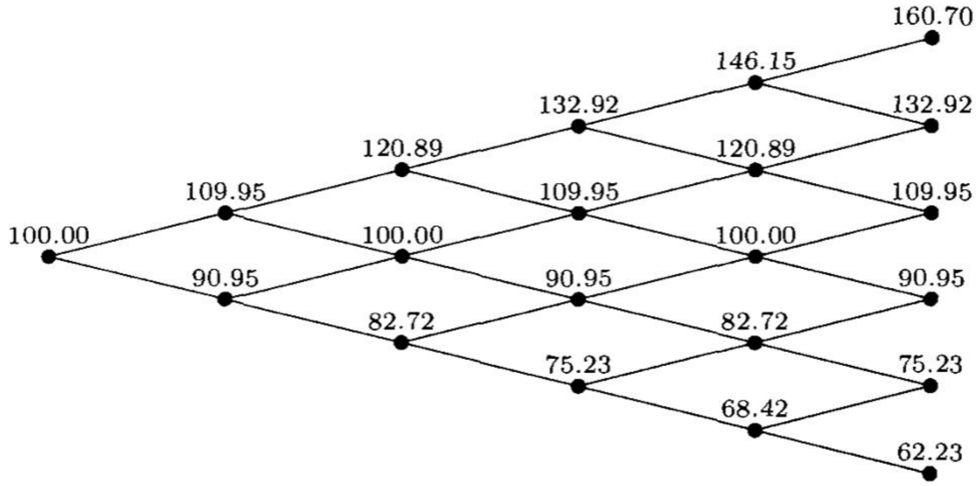
Example

Consider an American stock put option with six months to expiration. The stock price is 100, the strike price is 95, the risk-free interest rate is 8%, and the volatility is 30%. The option is priced in a binomial tree with five time steps. $S = 100$, $X = 95$, $T = 0.5$, $r = b = 0.08$, $\sigma = 0.3$, and $n = 5$.

$$\Delta t = \frac{0.5}{5} = 0.1$$

$$u = e^{0.3\sqrt{0.1}} = 1.0995 \qquad d = e^{-0.3\sqrt{0.1}} = 0.9095$$

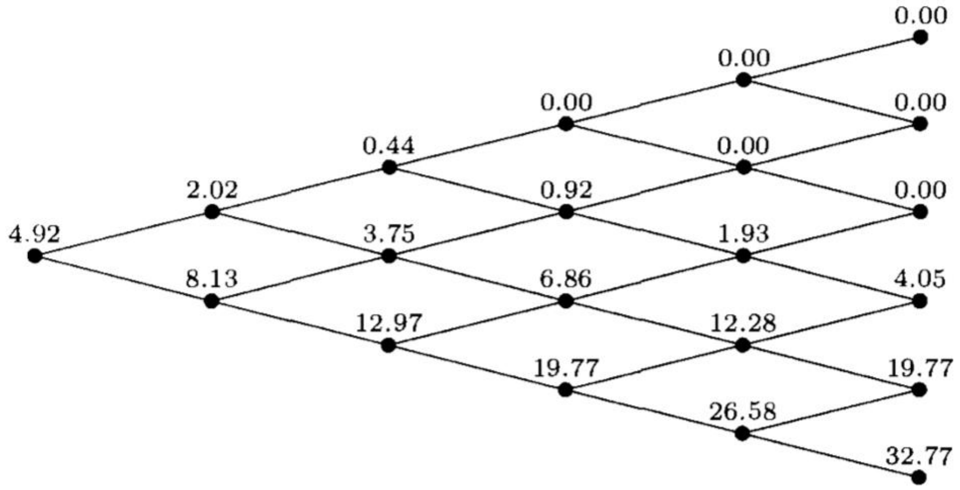
$$p = \frac{e^{0.08\times 0.1} - 0.9095}{1.0995 - 0.9095} = 0.5186$$



First, we start at the end of the tree to see if it is optimal to exercise the option $\max[X - S, 0]$. For example, in the end node with asset price 62.23, it is naturally optimal to exercise the put option: $\max[95 - 62.23, 0] = 32.77$, while at, for example, the end node with asset price

100.95, it is not optimal to exercise the option $\max[95 - 100.95, 0] = 0$. After checking for optimal exercise at each end node, we can now easily find the value of the American put option by standard backward induction (rolling back through the tree), where we check at each node if it is optimal with early exercise:

$$P_{j,i} = \max\{X - Su^i d^{j-i}, e^{-r\Delta t}[pP_{j+1,i+1} + (1-p)P_{j+1,i}]\}$$



The value of the American put option is therefore approximately 4.92.

Number of Nodes in a Binomial Tree
The number of nodes in the binomial tree is $\frac{n(n+1)}{2}$ when we count the number of time steps from 1. When we count the number of time steps from 0, as we have done in this chapter, the number of nodes in the binomial tree is $\frac{(n+1)(n+2)}{2}$.

Local Volatility
The local volatility in a standard CRR binomial tree is naturally constant for each time step and is given by

$$\sigma_{j,i} = \frac{1}{\sqrt{\Delta t}}\sqrt{p(1-p)}\ln(u^2)$$

The local volatility is generally different from the input volatility. It converges to the global input volatility as the number of time steps becomes large.

Negative Probabilities in the CRR Tree
A low volatility and relatively high cost-of-carry can induce negative risk-neutral probabilities in the CRR tree; see Chriss (1997). More