# Lyceum

**Aristotle's Lyceum is the institution considered to be the forerunner of the modern university. Opened in 335 BC, the Lyceum was a center of study and research in both science and philosophy.**

# Out Barrier Options and the Finite-difference Method

And now the pricing of our first exotic option, an up-and-out call. Knockout options have the same pay-off as vanilla options unless the underlying asset has reached some prescribed level, the barrier or trigger, prior to expiry. If the barrier is triggered then typically the option expires worthless. Financially and mathematically this means that the option value must be zero when the asset value is the barrier level.

In the code below we have had to include Barrier, the position of the barrier (above the Strike). We've also made the top of the asset-price array coincide with the position of the barrier.

```
Function OutBarrierOptionValue(Asset As Double, Strike As Double, _
     Expiry As Double, Volatility As Double, Divvie As Double, _
     Intrate As Double, Barrier As Double, NoAssetSteps As Integer) '(a)

Dim VOld(0 To 100) As Double
Dim VNew(0 To 100) As Double
Dim S(0 To 100) As Double

dS = Barrier / NoAssetSteps '(b)
NearestGridPt = Int(Asset / dS)
dummy = (Asset - NearestGridPt * dS) / dS
Timestep = dS * dS / Volatility / Volatility / (Barrier * Barrier) '(c)
NoTimesteps = Int(Expiry / Timestep) + 1
Timestep = Expiry / NoTimesteps

For i = 0 To NoAssetSteps
S(i) = i * dS
VOld(i) = Application.Max(S(i) - Strike, 0)
Next i

For j = 1 To NoTimesteps

For i = 1 To NoAssetSteps - 1
Delta = (VOld(i + 1) - VOld(i - 1)) / (2 * dS)
Gamma = (VOld(i + 1) - 2 * VOld(i) + VOld(i - 1)) / (dS * dS)
VNew(i) = VOld(i) + Timestep * (0.5 * Volatility * Volatility * _
     S(i) * S(i) * Gamma + (Intrate – Divvie) * S(i) * Delta - Intrate * VOld(i))
Next i

VNew(0) = 0
VNew(NoAssetSteps) = 0 '(d)

For i = 0 To NoAssetSteps
VOld(i) = VNew(i)
Next i

Next j

OutBarrierOptionValue = (1 - dummy) * VOld(NearestGridPt) + _
     dummy * VOld(NearestGridPt + 1)
End Function
```

**Notes:**

(a) The function call now asks for barrier information

(b) The asset array only needs to go as far as the barrier

(c) The timestep has been chosen to ensure stability of the explicit method

(d) This is the barrier boundary condition

**Exercise:**

Change the above code to value

1. an up-and-out call option if there is a rebate at the time of knockout
2. a down-and-out call option