Analytical models for American options.
Build on application in Excel/VBA to compare Barone-Adesi-Whaly, Bjerksund-Stensland.
Also use CRR with Black-Scholes smoothing and Richardson extrapolation for any number of steps to compare with analytical models.

Barone-Adesi-Whaley model

1 .Call option value

```
Public Function BAW_call(S As Double, X As Double, T As Double, r As Double, D As Double, sigma As Double) As Double
   Dim K As Double, a2 As Double
     Dim n As Double, q1 As Double, q2 As Double
     Dim d1 As Double, Sx As Double
```

If $S_t < S^*$

$$C_t = S_t + A_2\left(\frac{S_t}{S^*}\right)^{q_2}$$

And else $C_t = S_t - X$

```
If S < Sx Then
          BAW_call = Call(S, X, T, r, D, sigma) + a2 * (S / Sx) ^ q2
Else
          BAW_call = S - X
End If
```

Defining the variables as :

$$A_2 = \frac{S^*\left\{1 - e^{\sigma(T-t)}N(d_1)\right\}}{q_2}$$

```
a2 = (1 - Exp(-D * T) * Application.NormSDist(d1)) * (Sx / q2)
```

Where

```
d1 = (Log(Sx / X) + (r - D + 0.5 * sigma ^ 2) * T) / (sigma * Sqr(T))
```

$$q_2 = \frac{1 - n + \sqrt{(n-1)^2 - 4k}}{2}$$

```
q2 = (1 - n + Sqr((n - 1) ^ 2 + 4 * K)) / 2
```

$$n = \frac{2(r-\delta)}{\sigma^2}$$

```
n = 2 * (r - D) / sigma ^ 2
```

$$k = \frac{2r}{\sigma^2(1 - e^{-r(T-t)})}$$

```
K = (2 * r / sigma ^ 2) / (1 - Exp(-r * T))
```

The critical value of s* is defined as:

$$S^* - K = c_t(S^*, K, T-t) + \{1 - e^{-r(T-t)} N(d_1)\}\frac{S^*}{q_2}$$

```
Sx = Sx_call(S, X, T, r, D, sigma)
```

2. Put option values, we have a set of formulas to determine the value of an American put

```
Public Function BAW_Put(S As Double, X As Double, T As Double, r As Double,
D As Double, sigma As Double) As Double
Dim A1 As Double, a2 As Double, n As Double, K As Double
Dim q1 As Double, q2 As Double
Dim d1 As Double, Sxx As Double
```

If $S_t > S^{**}$

$$P_t = S_t + A_1\left(\frac{S_t}{S^{**}}\right)^{q_1}$$

And else $C_t = X - S_t$

```
If S > Sxx Then
        BAW_Put = Put(S, X, T, r, D, sigma) + A1 * (S / Sxx) ^ q1
Else
        BAW_Put = X - S
```

$$A_1 = \frac{S^{**}\left[1 - e^{\sigma(T-t)} N(d_1)\right]}{q_1}$$

```
A1 = -(1 - Exp(-D * T) * Application.NormSDist(-d1)) * (Sxx / q1)
```

$$q_1 = \frac{1 - n - \sqrt{(n-1)^2 + 4k}}{2}$$

```
q1 = (1 - n - Sqr((n - 1) ^ 2 + 4 * K)) / 2
```

n and k is the same as call option

```
Sxx = Sxx_put(S, X, T, r, D, sigma)
```

Input : S, T, t ,r , σ , δ , X

Output : C_t , P_t

Bjerksund-Stensland

```
Public Function BjS_call(S As Double, X As Double, T As Double, r As
Double, D As Double, sigma As Double, gamma As Double) As Double
Dim C As Double
```

```
Dim alpha As Double
Dim beta As Double
Dim I As Double
Dim B0, B00, f As Double
```

$$C = \alpha S - \alpha\phi(S,T,\beta,I,I) + \phi(S,T,1,I,I) - \phi(S,T,1,K,I)$$
$$- K\phi(S,T,0,I,I) + K\phi(S,T,0,K,I)$$

```
C = alpha * S - alpha * phi(S, T, beta, I, I) + phi(S, T, 1, I, I) - phi(S,
T, 1, X, I) - X * phi(S, T, 0, I, I) + X * phi(S, T, 0, X, I)
BjS_call = C
```

Where $\alpha = (I - K)I^{\beta}$

```
alpha = (I - X) / I ^ beta
```

$$\beta = \left(\frac{1}{2} - \frac{r-D}{\sigma^2}\right) + \sqrt{\left(\frac{r-D}{\sigma^2} - \frac{1}{2}\right)^2 + 2\left(\frac{r}{\sigma^2}\right)}$$

```
beta = (0.5 - (r - D) / sigma ^ 2) + Sqr(((r - D) / sigma ^ 2 - 1 / 2)
^ 2 + 2 * r / sigma ^ 2)
```

The function ϕ is given as

$$\phi(S,T,\gamma,H,I) = e^{\lambda}S^{\gamma}\left[N(d) - \left(\frac{I}{S}\right)^{k} N\left(d - \frac{2\ln(I/S)}{\sigma\sqrt{T}}\right)\right]$$

```
phi = Exp(lamda) * S ^ gamma * (d - (I / S) ^ k * (d - 2 * Log(I / S) /
sigma/ Sqr(T)))
```

Where

$$\lambda = \left[\gamma(r-D) - r + \frac{1}{2}\gamma(\gamma-1)\sigma^2\right]T$$

```
lamda = (gamma * (r - D) - r + 0.5 * gamma * (gamma - 1) * sigma^ 2) *
T
```

$$d = \frac{\ln(S/H) + (r - D + (\gamma - 0.5)\sigma^2)T}{\sigma\sqrt{T}}$$

```
d = -(Log(S / H) + (r - D + (gamma - 0.5) * sigma ^ 2) * T) / sigma / Sqr(T)
```

$$k = \frac{2(r-D)}{\sigma^2} + (2\gamma - 1)$$

```
k = 2 * (r - D) / sigma^ 2 + (2 * gamma - 1)
```

The trigger price I is given as the following:

$$I = B_0 + (B_\infty - B_0)(1 - e^{f})$$

```
I = B0 + (B00 - B0) * (1 - Exp(f))
```

$$f = -\left(T(r-D) + 2\sigma\sqrt{T}\right)\cdot\left(\frac{B_0}{B_\infty - B_0}\right)$$

```
f = -(T * (r - D) + 2 * sigma * Sqr(T)) * (B0 / (B00 - B0))
```

$$B_\infty = \frac{\beta}{\beta - 1}\cdot K$$

```
B00 = beta / (beta - 1) * X
```

$$B_0 = X\cdot\max\left(1, \frac{r}{D}\right)$$

```
B0 = X * max(1, r / D)

Public Function Bjs_put(S As Double, X As Double, T As Double, r As Double,
D As Double, sigma As Double, gamma As Double) As Double
```

Price of American put option is approximation for the call option and apply put-call parity in the form of

$$P(S,K,T,r,r-D,\sigma) = C(S,K,T,D,D-r,\sigma)$$

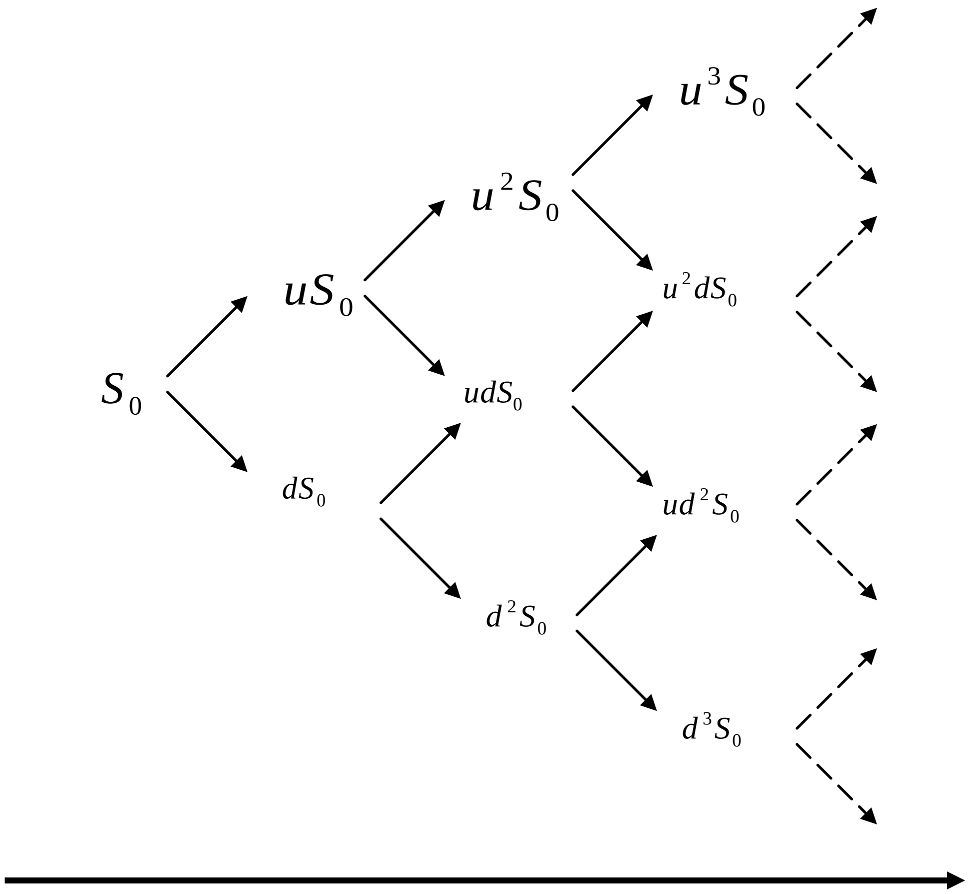
```
Bjs_put = BjS_call(S, X, T, r - D, D, sigma, gamma)
```

Input：S, K, T, t , D, σ , γ

Output：C_t P_t

CRR with Black-Scholes Smoothing

The Multi Period Binomial Model



CRR formular：

$$S_i = \begin{cases} uS_{i-1} & prob \quad p_i \\ dS_{i-1} & \quad 1-p_i \end{cases}$$

$$u = e^{\sigma\sqrt{\frac{T}{n}}}, \quad d = e^{-\sigma\sqrt{\frac{T}{n}}}, \qquad p = \frac{e^{\frac{rT}{n}} - d}{u - d} \quad where \quad \frac{\mathrm{rt}}{\mathrm{n}}$$

Input: s, σ ,r , k , T, n,   q,

call option price：

If s> k ,then

At n step all of points value：C=max(s—k , 0)

The n—1 step all of points value：

$$C = Se^{-qT}N(d_1) - Ke^{-rT}N(d_2)$$

$$d_1 = \frac{\ln\left(\frac{S}{K}\right) + \left(r - q + \frac{\sigma^2}{2}\right)T}{\sigma \cdot \sqrt{T}}$$

$$d_2 = d_1 - \sigma \cdot \sqrt{T}$$

From n-1 to 0 value：

$$C_{i,j} = e^{-1}\left[C_{i+1,j+1} \times p + C_{i+1,j} \times (1-p)\right]$$

Input：s, σ ,r , k , T, n, q

Output： $C_{i,j}$

If s<k then

the n step all of points value：P=max(k—s , 0)

the n-1 step all of points value:

$$P = Ke^{-rT}N(-d_2) - Se^{-qT}N(-d_1)$$

$$d_1 = \frac{\ln\left(\frac{S}{K}\right) + \left(r - q + \frac{\sigma^2}{2}\right)T}{\sigma \cdot \sqrt{T}}$$

$$d_2 = d_1 - \sigma \cdot \sqrt{T}$$

From n-1 to 0 step：

$$P_{i,j} = e^{-1}\left[P_{i+1,j+1} \times p + P_{i+1,j} \times (1-p)\right]$$

Input：s, σ ,r , k , T, n, q

Output： $P_{i,j}$

Richardson extrapolation

Input：m

if s> k ,then

$$F_c = \frac{2^m C(n/2) - C(n)}{2^m - 1}$$

If s < k ,then

$$F_p = \frac{2^m P(n/2) - P(n)}{2^m - 1}$$

Output： F_c F_p

```
Public Function RE_call(S As Double, X As Double, T As Double, r As Double,
D As Double, sigma As Double, n As Double, m As Double) As Double

 If S > X Then
    RE_call = 2 ^ m / (2 ^ m - 1) * (BS_Call(S, X, T, r, D, sigma, n /
2) - BS_Call(S, X, T, r, D, sigma, n))
 End If
End Function

Public Function RE_put(S As Double, X As Double, T As Double, r As Double,
D As Double, sigma As Double, n As Double, m As Double) As Double
  If S < X Then
    RE_put = 2 ^ m / (2 ^ m - 1) * (BS_Put(S, X, T, r, D, sigma, n / 2)
- BS_Put(S, X, T, r, D, sigma, n))
  End If
End Function
```