NAG Toolbox

nag_specfun_opt_barrier_std_price (s30fa)

1 Purpose

nag_specfun_opt_barrier_std_price (s30fa) computes the price of a standard barrier option.

2 Syntax

```
[p, ifail] = nag_specfun_opt_barrier_std_price(calput, type, x, s, h, k, t,
sigma, r, q, 'm', m, 'n', n)
```

```
[p, ifail] = s30fa(calput, type, x, s, h, k, t, sigma, r, q, 'm', m, 'n', n)
```

3 Description

nag_specfun_opt_barrier_std_price (s30fa) computes the price of a standard barrier option, where the exercise, for a given strike price, X, depends on the underlying asset price, S, reaching or crossing a specified barrier level, H. Barrier options of type **In** only become active (are knocked in) if the underlying asset price attains the pre-determined barrier level during the lifetime of the contract. Those of type **Out** start active and are knocked out if the underlying asset price attains the barrier level during the lifetime of the contract. A cash rebate, K, may be paid if the option is inactive at expiration. The option may also be described as **Up** (the underlying price starts below the barrier level) or **Down** (the underlying price starts above the barrier level). This gives the following options which can be specified as put or call contracts.

Down-and-In: the option starts inactive with the underlying asset price above the barrier level. It is knocked in if the underlying price moves down to hit the barrier level before expiration.

Down-and-Out: the option starts active with the underlying asset price above the barrier level. It is knocked out if the underlying price moves down to hit the barrier level before expiration.

Up-and-In: the option starts inactive with the underlying asset price below the barrier level. It is knocked in if the underlying price moves up to hit the barrier level before expiration.

Up-and-Out: the option starts active with the underlying asset price below the barrier level. It is knocked out if the underlying price moves up to hit the barrier level before expiration.

The payoff is $\max(S - X, 0)$ for a call or $\max(X - S, 0)$ for a put, if the option is active at expiration, otherwise it may pay a pre-specified cash rebate, K. Following Haug (2007), the prices of the various standard barrier options can be written as shown below. The volatility, σ, risk-free interest rate, r, and annualised dividend yield, q, are constants. The integer parameters, j and k, take the values ± 1, depending on the type of barrier.

$$
\begin{aligned}
A &= jSe^{-qT}\Phi(jx_1) - jXe^{-rT}\Phi\left(j\left[x_1 - \sigma\sqrt{T}\right]\right) \\
B &= jSe^{-qT}\Phi(jx_2) - jXe^{-rT}\Phi\left(j\left[x_2 - \sigma\sqrt{T}\right]\right) \\
C &= jSe^{-qT}\left(\tfrac{H}{S}\right)^{2(\mu+1)}\Phi(ky_1) - jXe^{-rT}\left(\tfrac{H}{S}\right)^{2\mu}\Phi\left(k\left[y_1 - \sigma\sqrt{T}\right]\right) \\
D &= jSe^{-qT}\left(\tfrac{H}{S}\right)^{2(\mu+1)}\Phi(ky_2) - jXe^{-rT}\left(\tfrac{H}{S}\right)^{2\mu}\Phi\left(k\left[y_2 - \sigma\sqrt{T}\right]\right) \\
E &= Ke^{-rT}\left\{\Phi\left(k\left[x_2 - \sigma\sqrt{T}\right]\right) - \left(\tfrac{H}{S}\right)^{2\mu}\Phi\left(k\left[y_2 - \sigma\sqrt{T}\right]\right)\right\} \\
F &= K\left\{\left(\tfrac{H}{S}\right)^{\mu+\lambda}\Phi(kz) + \left(\tfrac{H}{S}\right)^{\mu-\lambda}\Phi\left(k\left[z - \sigma\sqrt{T}\right]\right)\right\}
\end{aligned}
$$

with

$$
\begin{aligned}
x_1 &= \frac{\ln(S/X)}{\sigma\sqrt{T}} + (1+\mu)\sigma\sqrt{T} \\
x_2 &= \frac{\ln(S/H)}{\sigma\sqrt{T}} + (1+\mu)\sigma\sqrt{T} \\
y_1 &= \frac{\ln\left(H^2/(SX)\right)}{\sigma\sqrt{T}} + (1+\mu)\sigma\sqrt{T} \\
y_2 &= \frac{\ln(H/S)}{\sigma\sqrt{T}} + (1+\mu)\sigma\sqrt{T} \\
z &= \frac{\ln(H/S)}{\sigma\sqrt{T}} + \lambda\sigma\sqrt{T} \\
\mu &= \frac{r-q-\sigma^2/2}{\sigma^2} \\
\lambda &= \sqrt{\mu^2 + \frac{2r}{\sigma^2}}
\end{aligned}
$$

and where Φ denotes the cumulative Normal distribution function,

$$\Phi(x) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x} \exp\left(-y^2/2\right)dy.$$

Down-and-In ($S > H$):

When $X \geq H$, with $j = k = 1$,

$$P_{\text{call}} = C + E$$

and with $j = -1,\ k = 1$

$$P_{\text{put}} = B - C + D + E$$

When $X < H$, with $j = k = 1$

$$P_{\text{call}} = A - B + D + E$$

and with $j = -1,\ k = 1$

$$P_{\text{put}} = A + E.$$

Down-and-Out ($S > H$):

When $X \geq H$, with $j = k = 1$,

$$P_{\text{call}} = A - C + F$$

and with $j = -1,\ k = 1$

$$P_{\text{put}} = A - B + C - D + F$$

When $X < H$, with $j = k = 1$,

$$P_{\text{call}} = B - D + F$$

and with $j = -1,\ k = 1$

$$P_{\text{put}} = F.$$

Up-and-In ($S < H$):

When $X \geq H$, with $j = 1,\ k = -1$,

$$P_{\text{call}} = A + E$$

and with $j = k = -1$,

$$P_{\text{put}} = A - B + D + E$$

When $X < H$, with $j = 1$, $k = -1$,

$$P_{\text{call}} = B - C + D + E$$

and with $j = k = -1$,

$$P_{\text{put}} = C + E.$$

Up-and-Out ($S < H$):

When $X \geq H$, with $j = 1$, $k = -1$,

$$P_{\text{call}} = F$$

and with $j = k = -1$,

$$P_{\text{put}} = B - D + F$$

When $X < H$, with $j = 1$, $k = -1$,

$$P_{\text{call}} = A - B + C - D + F$$

and with $j = k = -1$,

$$P_{\text{put}} = A - C + F.$$

The option price $P_{ij} = P(X = X_i, T = T_j)$ is computed for each strike price in a set X_i, $i = 1, 2, \ldots, m$, and for each expiry time in a set T_j, $j = 1, 2, \ldots, n$.

4 References

Haug E G (2007) *The Complete Guide to Option Pricing Formulas* (2nd Edition) McGraw-Hill

5 Parameters

5.1 Compulsory Input Parameters

1: **calput** – CHARACTER(1)

Determines whether the option is a call or a put.

calput = 'C'
A call; the holder has a right to buy.

calput = 'P'
A put; the holder has a right to sell.

Constraint: **calput** = 'C' or 'P'.

2: **type** – CHARACTER(2)

Indicates the barrier type as **In** or **Out** and its relation to the price of the underlying asset as **Up** or **Down**.

type = 'DI'
Down-and-In.

type = 'DO'
Down-and-Out.

type = 'UI'
Up-and-In.

type = 'UO'
Up-and-Out.

Constraint: **type** = 'DI', 'DO', 'UI' or 'UO'.

3: **x**(**m**) – REAL (KIND=nag_wp) array

x(i) must contain X_i, the ith strike price, for $i = 1, 2, \ldots, \mathbf{m}$.

Constraint: **x**(i) $\geq z$ and **x**(i) $\leq 1/z$, where $z = $ x02am(), the safe range parameter, for $i = 1, 2, \ldots, \mathbf{m}$.

4: **s** – REAL (KIND=nag_wp)

S, the price of the underlying asset.

Constraint: **s** $\geq z$ and **s** $\leq 1.0/z$, where $z = $ x02am(), the safe range parameter.

5: **h** – REAL (KIND=nag_wp)

The barrier price.

Constraint: **h** $\geq z$ and **h** $\leq 1/z$, where $z = $ x02am(), the safe range parameter.

6: **k** – REAL (KIND=nag_wp)

The value of a possible cash rebate to be paid if the option has not been knocked in (or out) before expiration.

Constraint: **k** ≥ 0.0.

7: **t**(**n**) – REAL (KIND=nag_wp) array

t(i) must contain T_i, the ith time, in years, to expiry, for $i = 1, 2, \ldots, \mathbf{n}$.

Constraint: **t**(i) $\geq z$, where $z = $ x02am(), the safe range parameter, for $i = 1, 2, \ldots, \mathbf{n}$.

8: **sigma** – REAL (KIND=nag_wp)

σ, the volatility of the underlying asset. Note that a rate of 15% should be entered as 0.15.

Constraint: **sigma** > 0.0.

9: **r** – REAL (KIND=nag_wp)

r, the annual risk-free interest rate, continuously compounded. Note that a rate of 5% should be entered as 0.05.

Constraint: **r** ≥ 0.0.

10: **q** – REAL (KIND=nag_wp)

q, the annual continuous yield rate. Note that a rate of 8% should be entered as 0.08.

Constraint: **q** ≥ 0.0.

5.2 Optional Input Parameters

1: **m** – INTEGER

Default: the dimension of the array **x**.

The number of strike prices to be used.

Constraint: **m** ≥ 1.

2: **n** – INTEGER

Default: the dimension of the array **t**.

The number of times to expiry to be used.

Constraint: **n** ≥ 1.

5.3 Output Parameters

1: **p**(ldp, **n**) – REAL (KIND=nag_wp) array

$ldp =$ **m**.

p(i, j) contains P_{ij}, the option price evaluated for the strike price $\mathbf{x}_i$ at expiry $\mathbf{t}_j$ for $i = 1, 2, \ldots,$ **m** and $j = 1, 2, \ldots,$ **n**.

2: **ifail** – INTEGER

ifail $= 0$ unless the function detects an error (see Section 5).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail $= 1$

On entry, **calput** $= \langle$*value*$\rangle$ was an illegal value.

ifail $= 2$

On entry, **type** $= \langle$*value*$\rangle$ was an illegal value.

ifail $= 3$

Constraint: **m** ≥ 1.

ifail $= 4$

Constraint: **n** ≥ 1.

ifail $= 5$

Constraint: **x**$(i) \geq \langle$*value*$\rangle$ and **x**$(i) \leq \langle$*value*$\rangle$.

ifail $= 6$

Constraint: **s** $\geq \langle$*value*$\rangle$ and **s** $\leq \langle$*value*$\rangle$.

ifail $= 7$

Constraint: **h** $\geq \langle$*value*$\rangle$ and **h** $\leq \langle$*value*$\rangle$.

ifail $= 8$

Constraint: **k** ≥ 0.0.

ifail $= 9$

Constraint: **t**$(i) \geq \langle$*value*$\rangle$.

ifail $= 10$

Constraint: **sigma** > 0.0.

ifail $= 11$

Constraint: **r** ≥ 0.0.

ifail $= 12$

Constraint: **q** ≥ 0.0.

ifail $= 14$

Constraint: $ldp \geq$ **m**.

ifail $= 15$

On entry, **s** and **h** are inconsistent with **type**.

ifail $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

ifail $= -399$

Your licence key may have expired or may not have been installed correctly.

ifail $= -999$

Dynamic memory allocation failed.

7 Accuracy

The accuracy of the output is dependent on the accuracy of the cumulative Normal distribution function, Φ. This is evaluated using a rational Chebyshev expansion, chosen so that the maximum relative error in the expansion is of the order of the ***machine precision*** (see nag_specfun_cdf_normal (s15ab) and nag_specfun_erfc_real (s15ad)). An accuracy close to ***machine precision*** can generally be expected.

8 Further Comments

None.

9 Example

This example computes the price of a Down-and-In put with a time to expiry of 6 months, a stock price of 100 and a strike price of 100. The barrier value is 95 and there is a cash rebate of 3, payable on expiry if the option has not been knocked in. The risk-free interest rate is 8% per year, there is an annual dividend return of 4% and the volatility is 30% per year.

9.1 Program Text

```
     function s30fa_example

fprintf('s30fa example results\n\n');

put = 'P';
type = 'DI';
s = 100.0;
h = 95.0;
k = 3.0;
sigma = 0.3;
r = 0.08;
q = 0.04;
x = [100.0];
t = [0.5];

[p, ifail] = s30fa( ...
                    put, type, x, s, h, k, t, sigma, r, q);

fprintf('\nStandard Barrier Option\n Put :\n');
fprintf('  Spot       =   %9.4f\n', s);
fprintf('  Barrier    =   %9.4f\n', h);
fprintf('  Rebate     =   %9.4f\n', k);
fprintf('  Volatility =   %9.4f\n', sigma);
fprintf('  Rate       =   %9.4f\n', r);
```

```
fprintf('  Dividend   =   %9.4f\n\n', q);

fprintf('   Strike    Expiry   Option Price\n');

for i=1:1
  for j=1:1
    fprintf('%9.4f %9.4f %9.4f\n', x(i), t(j), p(i,j));
  end
end
```

9.2 Program Results

```
     s30fa example results

Standard Barrier Option
 Put :
  Spot       =    100.0000
  Barrier    =     95.0000
  Rebate     =      3.0000
  Volatility =      0.3000
  Rate       =      0.0800
  Dividend   =      0.0400

   Strike    Expiry   Option Price
 100.0000    0.5000    7.7988
```