NAG Toolbox

nag_specfun_opt_binary_aon_greeks (s30cd)

1 Purpose

nag_specfun_opt_binary_aon_greeks (s30cd) computes the price of a binary or digital asset-or-nothing option together with its sensitivities (Greeks).

2 Syntax

```
[p, delta, gamma, vega, theta, rho, crho, vanna, charm, speed, colour, zomma,
vomma, ifail] = nag_specfun_opt_binary_aon_greeks(calput, x, s, t, sigma, r, q,
'm', m, 'n', n)
```

```
[p, delta, gamma, vega, theta, rho, crho, vanna, charm, speed, colour, zomma,
vomma, ifail] = s30cd(calput, x, s, t, sigma, r, q, 'm', m, 'n', n)
```

3 Description

nag_specfun_opt_binary_aon_greeks (s30cd) computes the price of a binary or digital asset-or-nothing option, together with the Greeks or sensitivities, which are the partial derivatives of the option price with respect to certain of the other input parameters. This option pays the underlying asset itself, S, at expiration if the option is in-the-money (see Section 2.4 in the S Chapter Introduction). For a strike price, X, underlying asset price, S, and time to expiry, T, the payoff is therefore S, if $S > X$ for a call or $S < X$ for a put. Nothing is paid out when this condition is not met.

The price of a call with volatility, σ, risk-free interest rate, r, and annualised dividend yield, q, is

$$P_{\text{call}} = Se^{-qT}\Phi(d_1)$$

and for a put,

$$P_{\text{put}} = Se^{-qT}\Phi(-d_1)$$

where Φ is the cumulative Normal distribution function,

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

and

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

The option price $P_{ij} = P\left(X = X_i, T = T_j\right)$ 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

Reiner E and Rubinstein M (1991) Unscrambling the binary code *Risk* **4**

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

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

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

3: **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.

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

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

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

5: **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.

6: **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.

7: **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: $\mathbf{n} \geq 1$.

5.3 Output Parameters

1: $\mathbf{p}(\mathit{ldp}, \mathbf{n})$ – REAL (KIND=nag_wp) array

$\mathit{ldp} = \mathbf{m}$.

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

2: $\mathbf{delta}(\mathit{ldp}, \mathbf{n})$ – REAL (KIND=nag_wp) array

$\mathit{ldp} = \mathbf{m}$.

The leading $\mathbf{m} \times \mathbf{n}$ part of the array **delta** contains the sensitivity, $\frac{\partial P}{\partial S}$, of the option price to change in the price of the underlying asset.

3: $\mathbf{gamma}(\mathit{ldp}, \mathbf{n})$ – REAL (KIND=nag_wp) array

$\mathit{ldp} = \mathbf{m}$.

The leading $\mathbf{m} \times \mathbf{n}$ part of the array **gamma** contains the sensitivity, $\frac{\partial^2 P}{\partial S^2}$, of **delta** to change in the price of the underlying asset.

4: $\mathbf{vega}(\mathit{ldp}, \mathbf{n})$ – REAL (KIND=nag_wp) array

$\mathit{ldp} = \mathbf{m}$.

$\mathbf{vega}(i, j)$, contains the first-order Greek measuring the sensitivity of the option price P_{ij} to change in the volatility of the underlying asset, i.e., $\frac{\partial P_{ij}}{\partial \sigma}$, for $i = 1, 2, \ldots, \mathbf{m}$ and $j = 1, 2, \ldots, \mathbf{n}$.

5: $\mathbf{theta}(\mathit{ldp}, \mathbf{n})$ – REAL (KIND=nag_wp) array

$\mathit{ldp} = \mathbf{m}$.

$\mathbf{theta}(i, j)$, contains the first-order Greek measuring the sensitivity of the option price P_{ij} to change in time, i.e., $-\frac{\partial P_{ij}}{\partial T}$, for $i = 1, 2, \ldots, \mathbf{m}$ and $j = 1, 2, \ldots, \mathbf{n}$, where $b = r - q$.

6: $\mathbf{rho}(\mathit{ldp}, \mathbf{n})$ – REAL (KIND=nag_wp) array

$\mathit{ldp} = \mathbf{m}$.

$\mathbf{rho}(i, j)$, contains the first-order Greek measuring the sensitivity of the option price P_{ij} to change in the annual risk-free interest rate, i.e., $-\frac{\partial P_{ij}}{\partial r}$, for $i = 1, 2, \ldots, \mathbf{m}$ and $j = 1, 2, \ldots, \mathbf{n}$.

7: $\mathbf{crho}(\mathit{ldp}, \mathbf{n})$ – REAL (KIND=nag_wp) array

$\mathit{ldp} = \mathbf{m}$.

$\mathbf{crho}(i, j)$, contains the first-order Greek measuring the sensitivity of the option price P_{ij} to change in the annual cost of carry rate, i.e., $-\frac{\partial P_{ij}}{\partial b}$, for $i = 1, 2, \ldots, \mathbf{m}$ and $j = 1, 2, \ldots, \mathbf{n}$, where $b = r - q$.

8: $\mathbf{vanna}(\mathit{ldp}, \mathbf{n})$ – REAL (KIND=nag_wp) array

$\mathit{ldp} = \mathbf{m}$.

$\mathbf{vanna}(i, j)$, contains the second-order Greek measuring the sensitivity of the first-order Greek Δ_{ij} to change in the volatility of the asset price, i.e., $-\frac{\partial \Delta_{ij}}{\partial T} = -\frac{\partial^2 P_{ij}}{\partial S \partial \sigma}$, for $i = 1, 2, \ldots, \mathbf{m}$ and $j = 1, 2, \ldots, \mathbf{n}$.

9: **charm**$(ldp,\mathbf{n})$ – REAL (KIND=nag_wp) array

$ldp = \mathbf{m}$.

charm(i,j), contains the second-order Greek measuring the sensitivity of the first-order Greek Δ_{ij} to change in the time, i.e., $-\frac{\partial \Delta_{ij}}{\partial T} = -\frac{\partial^2 P_{ij}}{\partial S \partial T}$, for $i = 1,2,\ldots,\mathbf{m}$ and $j = 1,2,\ldots,\mathbf{n}$.

10: **speed**$(ldp,\mathbf{n})$ – REAL (KIND=nag_wp) array

$ldp = \mathbf{m}$.

speed(i,j), contains the third-order Greek measuring the sensitivity of the second-order Greek Γ_{ij} to change in the price of the underlying asset, i.e., $-\frac{\partial \Gamma_{ij}}{\partial S} = -\frac{\partial^3 P_{ij}}{\partial S^3}$, for $i = 1,2,\ldots,\mathbf{m}$ and $j = 1,2,\ldots,\mathbf{n}$.

11: **colour**$(ldp,\mathbf{n})$ – REAL (KIND=nag_wp) array

$ldp = \mathbf{m}$.

colour(i,j), contains the third-order Greek measuring the sensitivity of the second-order Greek Γ_{ij} to change in the time, i.e., $-\frac{\partial \Gamma_{ij}}{\partial T} = -\frac{\partial^3 P_{ij}}{\partial S \partial T}$, for $i = 1,2,\ldots,\mathbf{m}$ and $j = 1,2,\ldots,\mathbf{n}$.

12: **zomma**$(ldp,\mathbf{n})$ – REAL (KIND=nag_wp) array

$ldp = \mathbf{m}$.

zomma(i,j), contains the third-order Greek measuring the sensitivity of the second-order Greek Γ_{ij} to change in the volatility of the underlying asset, i.e., $-\frac{\partial \Gamma_{ij}}{\partial \sigma} = -\frac{\partial^3 P_{ij}}{\partial S^2 \partial \sigma}$, for $i = 1,2,\ldots,\mathbf{m}$ and $j = 1,2,\ldots,\mathbf{n}$.

13: **vomma**$(ldp,\mathbf{n})$ – REAL (KIND=nag_wp) array

$ldp = \mathbf{m}$.

vomma(i,j), contains the second-order Greek measuring the sensitivity of the first-order Greek Δ_{ij} to change in the volatility of the underlying asset, i.e., $-\frac{\partial \Delta_{ij}}{\partial \sigma} = -\frac{\partial^2 P_{ij}}{\partial \sigma^2}$, for $i = 1,2,\ldots,\mathbf{m}$ and $j = 1,2,\ldots,\mathbf{n}$.

14: **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 $= 2$

Constraint: $\mathbf{m} \geq 1$.

ifail $= 3$

Constraint: $\mathbf{n} \geq 1$.

ifail $= 4$

Constraint: $\mathbf{x}(i) \geq \langle value \rangle$ and $\mathbf{x}(i) \leq \langle value \rangle$.

ifail $= 5$

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

ifail $= 6$

Constraint: $\mathbf{t}(i) \geq \langle value \rangle$.

ifail $= 7$

Constraint: **sigma** > 0.0.

ifail $= 8$

Constraint: **r** ≥ 0.0.

ifail $= 9$

Constraint: **q** ≥ 0.0.

ifail $= 11$

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

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 an asset-or-nothing put with a time to expiry of 292 days, a stock price of 70 and a strike price of 65. The risk-free interest rate is 5% per year, there is an annual dividend return of 3% and the volatility is 15% per year.

9.1 Program Text

```
     function s30cd_example

fprintf('s30cd example results\n\n');

put   = 'P';
s     = 70;
sigma = 0.15;
r     = 0.05;
```

```
q     = 0.03;
x     = [65.0];
t     = [0.8];

[p, delta, gamma,  vega,  theta,   rho,  crho, ...
    vanna, charm, speed, colour, zomma, vomma, ifail] = ...
    s30cd(...
          put, x, s, t, sigma, r, q);

fprintf('\nBinary (Digital): Asset-or-Nothing\n European Put :\n');
fprintf('  Spot       =   %9.4f\n', s);
fprintf('  Volatility =   %9.4f\n', sigma);
fprintf('  Rate       =   %9.4f\n', r);
fprintf('  Dividend   =   %9.4f\n\n', q);

fprintf(' Time to Expiry : %8.4f\n', t(1));
fprintf('%8s%9s%9s%9s%9s%9s%11s%11s\n','Strike','Price','Delta','Gamma',...
        'Vega','Theta','Rho','CRho');
fprintf('%8.4f%9.4f%9.4f%9.4f%9.4f%9.4f%11.4f%11.4f\n\n', x(1), p(1,1), ...
        delta(1,1), gamma(1,1), vega(1,1), theta(1,1), rho(1,1), crho(1,1));

fprintf('%26s%9s%9s%9s%11s%11s\n','Vanna','Charm','Speed','Colour',...
        'Zomma','Vomma');
fprintf('%17s%9.4f%9.4f%9.4f%9.4f%11.4f%11.4f\n\n', ' ', vanna(1,1), ...
        charm(1,1), speed(1,1), colour(1,1), zomma(1,1), vomma(1,1));
```

9.2 Program Results

```
     s30cd example results

Binary (Digital): Asset-or-Nothing
 European Put :
  Spot       =     70.0000
  Volatility =      0.1500
  Rate       =      0.0500
  Dividend   =      0.0300

 Time to Expiry :   0.8000
  Strike    Price    Delta    Gamma     Vega    Theta        Rho       CRho
 65.0000  15.7211  -1.9852   0.1422  83.6424  -4.2761  -123.7497  -111.1728

                     Vanna    Charm    Speed   Colour      Zomma      Vomma
                    9.3479  -1.1351   0.0118   0.2316    -2.6319  -989.9610
```