

## NAG Toolbox for Matlab

### s30cd

#### 1 Purpose

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] = s30cd(calput, x, s, t, sigma, r, q, 'm', m, 'n',
n)
```

#### 3 Description

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,  $\sigma$ , 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  $\Phi$  is the cumulative Normal distribution function,

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

and

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

#### 4 References

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

#### 5 Parameters

##### 5.1 Compulsory Input Parameters

1: **calput** – string

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) – double array**  
**m**, the dimension of the array, must satisfy the constraint  $\mathbf{m} \geq 1$ .  
**x(i)** must contain  $X_i$ , the  $i$ th strike price, for  $i = 1, 2, \dots, \mathbf{m}$ .  
*Constraint:*  $\mathbf{x}(i) \geq z$  and  $\mathbf{x}(i) \leq 1/z$ , where  $z = \text{X02AMF}()$ , the safe range parameter, for  $i = 1, 2, \dots, \mathbf{m}$ .
- 3: **s – double scalar**  
**S**, the price of the underlying asset.  
*Constraint:*  $\mathbf{s} \geq z$  and  $\mathbf{s} \leq 1/z$ , where  $z = \text{X02AMF}()$ , the safe range parameter.
- 4: **t(n) – double array**  
**n**, the dimension of the array, must satisfy the constraint  $\mathbf{n} \geq 1$ .  
**t(i)** must contain  $T_i$ , the  $i$ th time, in years, to expiry, for  $i = 1, 2, \dots, \mathbf{n}$ .  
*Constraint:*  $\mathbf{t}(i) \geq z$ , where  $z = \text{X02AMF}()$ , the safe range parameter, for  $i = 1, 2, \dots, \mathbf{n}$ .
- 5: **sigma – double scalar**  
 $\sigma$ , the volatility of the underlying asset. Note that a rate of 15% should be entered as 0.15.  
*Constraint:* **sigma** > 0.0.
- 6: **r – double scalar**  
 $r$ , the annual risk-free interest rate, continuously compounded. Note that a rate of 5% should be entered as 0.05.  
*Constraint:* **r**  $\geq$  0.0.
- 7: **q – double scalar**  
 $q$ , the annual continuous yield rate. Note that a rate of 8% should be entered as 0.08.  
*Constraint:* **q**  $\geq$  0.0.

## 5.2 Optional Input Parameters

- 1: **m – int32 scalar**  
*Default:* The dimension of the array **x**.  
the number of strike prices to be used.  
*Constraint:* **m**  $\geq$  1.
- 2: **n – int32 scalar**  
*Default:* The dimension of the array **t**.  
the number of times to expiry to be used.  
*Constraint:* **n**  $\geq$  1.

## 5.3 Input Parameters Omitted from the MATLAB Interface

ldp

## 5.4 Output Parameters

- 1: **p(ldp,n) – double array**  
The  $m \times n$  array **p** contains the computed option prices.

2: **delta**(*ldp,n*) – double array

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

3: **gamma**(*ldp,n*) – double array

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

4: **vega**(*ldp,n*) – double array

The  $m \times n$  array **vega** contains the sensitivity,  $\frac{\partial P}{\partial \sigma}$ , of the option price to change in the volatility of the underlying asset.

5: **theta**(*ldp,n*) – double array

The  $m \times n$  array **theta** contains the sensitivity,  $-\frac{\partial P}{\partial T}$ , of the option price to change in the time to expiry of the option.

6: **rho**(*ldp,n*) – double array

The  $m \times n$  array **rho** contains the sensitivity,  $\frac{\partial P}{\partial r}$ , of the option price to change in the annual risk-free interest rate.

7: **crho**(*ldp,n*) – double array

The  $m \times n$  array **crho** containing the sensitivity,  $\frac{\partial P}{\partial b}$ , of the option price to change in the annual cost of carry rate,  $b$ , where  $b = r - q$ .

8: **vanna**(*ldp,n*) – double array

The  $m \times n$  array **vanna** contains the sensitivity,  $\frac{\partial^2 P}{\partial S \partial \sigma}$ , of **vega** to change in the price of the underlying asset or, equivalently, the sensitivity of **delta** to change in the volatility of the asset price.

9: **charm**(*ldp,n*) – double array

The  $m \times n$  array **charm** contains the sensitivity,  $-\frac{\partial^2 P}{\partial S \partial T}$ , of **delta** to change in the time to expiry of the option.

10: **speed**(*ldp,n*) – double array

The  $m \times n$  array **speed** contains the sensitivity,  $\frac{\partial^3 P}{\partial S^3}$ , of **gamma** to change in the price of the underlying asset.

11: **colour**(*ldp,n*) – double array

The  $m \times n$  array **colour** contains the sensitivity,  $-\frac{\partial^3 P}{\partial S^2 \partial T}$ , of **gamma** to change in the time to expiry of the option.

12: **zomma**(*ldp,n*) – double array

The  $m \times n$  array **zomma** contains the sensitivity,  $\frac{\partial^3 P}{\partial S^2 \partial \sigma}$ , of **gamma** to change in the volatility of the underlying asset.

13: **vomma**(*ldp,n*) – double array

The  $m \times n$  array **vomma** contains the sensitivity,  $\frac{\partial^2 P}{\partial \sigma^2}$ , of **vega** to change in the volatility of the underlying asset.

14: **ifail** – **int32 scalar**

**ifail** = 0 unless the function detects an error (see Section 6).

## 6 Error Indicators and Warnings

Errors or warnings detected by the function:

**ifail** = 1

On entry, **calput**  $\neq$  'C' or 'P'.

**ifail** = 2

On entry, **m**  $\leq$  0.

**ifail** = 3

On entry, **n**  $\leq$  0.

**ifail** = 4

On entry, **x**(*i*)  $< z$  or **x**(*i*)  $> 1/z$ , where  $z = X02AMF()$ , the safe range parameter.

**ifail** = 5

On entry, **s**  $< z$  or **s**  $> 1/z$ , where  $z = X02AMF()$ , the safe range parameter.

**ifail** = 6

On entry, **t**(*i*)  $< z$ , where  $z = X02AMF()$ , the safe range parameter.

**ifail** = 7

On entry, **sigma**  $\leq$  0.0.

**ifail** = 8

On entry, **r**  $<$  0.0.

**ifail** = 9

On entry, **q**  $<$  0.0.

**ifail** = 11

On entry, **ldp**  $<$  **m**.

## 7 Accuracy

The accuracy of the output is dependent on the accuracy of the cumulative Normal distribution function,  $\Phi$ . 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 s15ab and s15ad). An accuracy close to *machine precision* can generally be expected.

## 8 Further Comments

None.

## 9 Example

```
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(' Strike Price Delta Gamma Vega Theta Rho
CRho\n');
fprintf('%8.4f %8.4f %8.4f %8.4f %8.4f %8.4f %8.4f %8.4f\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(' Strike Price Vanna Charm Speed Colour Zomma
Vomma\n');
fprintf('%8.4f %8.4f %8.4f %8.4f %8.4f %8.4f %8.4f %8.4f\n', x(1), ...
p(1,1), vanna(1,1), charm(1,1), speed(1,1), colour(1,1), ...
zomma(1,1), vomma(1,1));

```

```

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

Strike Price Vanna Charm Speed Colour Zomma Vomma
65.0000 15.7211 9.3479 -1.1351 0.0118 0.2316 -2.6319 -989.9610

```