NAG Library Routine Document

S30CCF

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of ***bold italicised*** terms and other implementation-dependent details.

1 Purpose

S30CCF computes the price of a binary or digital asset-or-nothing option.

2 Specification

```
SUBROUTINE S30CCF (CALPUT, M, N, X, S, T, SIGMA, R, Q, P, LDP, IFAIL)

INTEGER             M, N, LDP, IFAIL
REAL (KIND=nag_wp) X(M), S, T(N), SIGMA, R, Q, P(LDP,N)
CHARACTER(1)        CALPUT
```

3 Description

S30CCF computes the price of a binary or digital asset-or-nothing option which 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

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

4 References

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

5 Parameters

1: CALPUT – CHARACTER(1) *Input*

On entry: 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: M – INTEGER *Input*

On entry: the number of strike prices to be used.

Constraint: M ≥ 1.

3: N – INTEGER *Input*

On entry: the number of times to expiry to be used.

Constraint: N ≥ 1.

4: X(M) – REAL (KIND=nag_wp) array *Input*

On entry: X(i) must contain X_i, the ith strike price, for $i = 1, 2, \ldots, \mathrm{M}$.

Constraint: X(i) $\geq z$ and X(i) $\leq 1/z$, where $z =$ X02AMF(), the safe range parameter, for $i = 1, 2, \ldots, \mathrm{M}$.

5: S – REAL (KIND=nag_wp) *Input*

On entry: S, the price of the underlying asset.

Constraint: S $\geq z$ and S $\leq 1.0/z$, where $z =$ X02AMF(), the safe range parameter.

6: T(N) – REAL (KIND=nag_wp) array *Input*

On entry: T(i) must contain T_i, the ith time, in years, to expiry, for $i = 1, 2, \ldots, \mathrm{N}$.

Constraint: T(i) $\geq z$, where $z =$ X02AMF(), the safe range parameter, for $i = 1, 2, \ldots, \mathrm{N}$.

7: SIGMA – REAL (KIND=nag_wp) *Input*

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

Constraint: SIGMA > 0.0.

8: R – REAL (KIND=nag_wp) *Input*

On entry: 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.

9: Q – REAL (KIND=nag_wp) *Input*

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

Constraint: Q ≥ 0.0.

10: P(LDP,N) – REAL (KIND=nag_wp) array *Output*

On exit: the leading M $\times$ N part of the array P contains the computed option prices.

11: LDP – INTEGER *Input*

On entry: the first dimension of the array P as declared in the (sub)program from which S30CCF is called.

Constraint: LDP $\geq$ M.

12: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this parameter you should refer to Section 3.3 in the Essential Introduction for details.

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then

the value 1 is recommended. Otherwise, if you are not familiar with this parameter, the recommended value is 0. **When the value −1 or 1 is used it is essential to test the value of IFAIL on exit.**

On exit: IFAIL = 0 unless the routine detects an error or a warning has been flagged (see Section 6).

6 Error Indicators and Warnings

If on entry IFAIL = 0 or −1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

IFAIL = 1

On entry, CALPUT ≠ 'C' or 'P'.

IFAIL = 2

On entry, M ≤ 0.

IFAIL = 3

On entry, N ≤ 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 > $1.0/z$, 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 ≤ 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, 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 S15ABF and S15ADF). 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 0.5 years, a stock price of 70 and a strike price of 65. The risk-free interest rate is 7% per year, there is an annual dividend return of 5% and the volatility is 27% per year.

9.1 Program Text

```
    Program s30ccfe

!     S30CCF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: nag_wp, s30ccf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: q, r, s, sigma
      Integer                          :: i, ifail, j, ldp, m, n
      Character (1)                    :: calput
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: p(:,:), t(:), x(:)
!     .. Executable Statements ..
      Write (nout,*) 'S30CCF Example Program Results'

!     Skip heading in data file
      Read (nin,*)

      Read (nin,*) calput
      Read (nin,*) s, sigma, r, q
      Read (nin,*) m, n

      ldp = m
      Allocate (p(ldp,n),t(n),x(m))

      Read (nin,*)(x(i),i=1,m)
      Read (nin,*)(t(i),i=1,n)

      ifail = 0
      Call s30ccf(calput,m,n,x,s,t,sigma,r,q,p,ldp,ifail)

      Write (nout,*)
      Write (nout,*) 'Binary (Digital): Asset-or-Nothing'

      Select Case (calput)
      Case ('C','c')
        Write (nout,*) 'European Call :'
      Case ('P','p')
        Write (nout,*) 'European Put :'
      End Select

      Write (nout,99998) '  Spot       = ', s
      Write (nout,99998) '  Volatility = ', sigma
      Write (nout,99998) '  Rate       = ', r
      Write (nout,99998) '  Dividend   = ', q

      Write (nout,*)
      Write (nout,*) '   Strike    Expiry    Option Price'

      Do i = 1, m

        Do j = 1, n
          Write (nout,99999) x(i), t(j), p(i,j)
        End Do
```

```
      End Do

99999 Format (1X,2(F9.4,1X),6X,F9.4)
99998 Format (A,1X,F8.4)
    End Program s30ccfe
```

9.2 Program Data

```
S30CCF Example Program Data
 'P'                 : Call = 'C', Put = 'P'
 70.0 0.27 0.07 0.05 : S, SIGMA, R, Q
 1 1                 : M, N
 65.0                : X(I), I = 1,2,...M
 0.5                 : T(I), I = 1,2,...N
```

9.3 Program Results

```
 S30CCF Example Program Results

 Binary (Digital): Asset-or-Nothing
 European Put :
  Spot       =   70.0000
  Volatility =    0.2700
  Rate       =    0.0700
  Dividend   =    0.0500

    Strike    Expiry    Option Price
   65.0000    0.5000         20.2069
```