NAG Library Routine Document

S30CAF

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

S30CAF computes the price of a binary or digital cash-or-nothing option.

2 Specification

```
SUBROUTINE S30CAF (CALPUT, M, N, X, S, K, T, SIGMA, R, Q, P, LDP, IFAIL)
INTEGER            M, N, LDP, IFAIL
REAL (KIND=nag_wp) X(M), S, K, T(N), SIGMA, R, Q, P(LDP,N)
CHARACTER(1)       CALPUT
```

3 Description

S30CAF computes the price of a binary or digital cash-or-nothing option which pays a fixed amount, K, 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 K, 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

$$P_{\text{call}} = Ke^{-rT}\Phi(d_2)$$

and for a put,

$$P_{\text{put}} = Ke^{-rT}\Phi(-d_2)$$

where Φ is the cumulative Normal distribution function,

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

and

$$d_2 = \frac{\ln(S/X) + \left(r - q - \sigma^2/2\right)T}{\sigma\sqrt{T}}.$$

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 Arguments

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: K – REAL (KIND=nag_wp) *Input*

On entry: the amount, K, to be paid at expiration if the option is in-the-money, i.e., if S $>$ X(i) when CALPUT = 'C', or if S $<$ X(i) when CALPUT = 'P', for $i = 1, 2, \ldots, m$.

Constraint: K ≥ 0.0.

7: 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 $i = 1, 2, \ldots, \mathrm{N}$.

8: 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.

9: 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.

10: 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.

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

On exit: P(i, j) contains P_{ij}, the option price evaluated for the strike price X_i at expiry T_j for $i = 1, 2, \ldots, \mathrm{M}$ and $j = 1, 2, \ldots, \mathrm{N}$.

12: LDP – INTEGER *Input*

On entry: the first dimension of the array P as declared in the (sub)program from which S30CAF is called.

Constraint: LDP $\geq$ M.

13: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this argument you should refer to Section 3.4 in How to Use the NAG Library and its Documentation 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 argument, 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 $= \langle value \rangle$ was an illegal value.

IFAIL $= 2$

On entry, M $= \langle value \rangle$.
Constraint: M ≥ 1.

IFAIL $= 3$

On entry, N $= \langle value \rangle$.
Constraint: N ≥ 1.

IFAIL $= 4$

On entry, X($\langle value \rangle$) $= \langle value \rangle$.
Constraint: X(i) $\geq \langle value \rangle$ and X(i) $\leq \langle value \rangle$.

IFAIL $= 5$

On entry, S $= \langle value \rangle$.
Constraint: S $\geq \langle value \rangle$ and S $\leq \langle value \rangle$.

IFAIL $= 6$

On entry, K $= \langle value \rangle$.
Constraint: K ≥ 0.0.

IFAIL $= 7$

On entry, T($\langle value \rangle$) $= \langle value \rangle$.
Constraint: T(i) $\geq \langle value \rangle$.

IFAIL $= 8$

On entry, SIGMA $= \langle value \rangle$.
Constraint: SIGMA > 0.0.

IFAIL $= 9$

On entry, R $= \langle value \rangle$.
Constraint: R ≥ 0.0.

IFAIL $= 10$

On entry, Q $= \langle value \rangle$.
Constraint: Q ≥ 0.0.

IFAIL $= 12$

On entry, LDP $= \langle value \rangle$ and M $= \langle value \rangle$.
Constraint: LDP $\geq$ M.

IFAIL $= -99$

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

See Section 3.9 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -399$

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

See Section 3.8 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -999$

Dynamic memory allocation failed.

See Section 3.7 in How to Use the NAG Library and its Documentation for further information.

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 Parallelism and Performance

S30CAF is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this routine. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

9 Further Comments

None.

10 Example

This example computes the price of a cash-or-nothing put with a time to expiry of 0.75 years, a stock price of 100 and a strike price of 80. The risk-free interest rate is 6% per year and the volatility is 35% per year. If the option is in-the-money at expiration, i.e., if $S > X$, the payoff is 10.

10.1 Program Text

```
    Program s30cafe

!     S30CAF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: nag_wp, s30caf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: k, 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,*) 'S30CAF Example Program Results'

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

      Read (nin,*) calput
      Read (nin,*) s, k, 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 s30caf(calput,m,n,x,s,k,t,sigma,r,q,p,ldp,ifail)

      Write (nout,*)
      Write (nout,*) 'Binary (Digital): Cash-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) '  Payout     = ', k
      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 s30cafe
```

10.2 Program Data

```
S30CAF Example Program Data
 'P'                       : Call = 'C', Put = 'P'
 100.0 10.0 0.35 0.06 0.0 : S, K, SIGMA, R, Q
 1 1                       : M, N
 80.0                      : X(I), I = 1,2,...M
 0.75                      : T(I), I = 1,2,...N
```

10.3 Program Results

```
 S30CAF Example Program Results

 Binary (Digital): Cash-or-Nothing
 European Put :
  Spot       =  100.0000
  Payout     =   10.0000
  Volatility =    0.3500
  Rate       =    0.0600
  Dividend   =    0.0000

   Strike     Expiry    Option Price
   80.0000    0.7500          2.2155
```