NAG Library Routine Document

S30ABF

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

S30ABF computes the European option price given by the Black–Scholes–Merton formula together with its sensitivities (Greeks).

2 Specification

```
SUBROUTINE S30ABF (CALPUT, M, N, X, S, T, SIGMA, R, Q, P, LDP, DELTA,      &
                   GAMMA, VEGA, THETA, RHO, CRHO, VANNA, CHARM, SPEED,      &
                   COLOUR, ZOMMA, VOMMA, IFAIL)

INTEGER            M, N, LDP, IFAIL
REAL (KIND=nag_wp) X(M), S, T(N), SIGMA, R, Q, P(LDP,N), DELTA(LDP,N),      &
                   GAMMA(LDP,N), VEGA(LDP,N), THETA(LDP,N),                 &
                   RHO(LDP,N), CRHO(LDP,N), VANNA(LDP,N),                   &
                   CHARM(LDP,N), SPEED(LDP,N), COLOUR(LDP,N),               &
                   ZOMMA(LDP,N), VOMMA(LDP,N)
CHARACTER(1)       CALPUT
```

3 Description

S30ABF computes the price of a European call (or put) 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, by the Black–Scholes–Merton formula (see Black and Scholes (1973) and Merton (1973)). The annual volatility, σ, risk-free interest rate, r, and dividend yield, q, must be supplied as input. For a given strike price, X, the price of a European call with underlying price, S, and time to expiry, T, is

$$P_{\mathrm{call}} = Se^{-qT}\Phi(d_1) - Xe^{-rT}\Phi(d_2)$$

and the corresponding European put price is

$$P_{\mathrm{put}} = Xe^{-rT}\Phi(-d_2) - Se^{-qT}\Phi(-d_1)$$

and where Φ denotes the cumulative Normal distribution function,

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

and

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

$$d_2 = d_1 - \sigma\sqrt{T}.$$

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

Black F and Scholes M (1973) The pricing of options and corporate liabilities *Journal of Political Economy* **81** 637–654

Merton R C (1973) Theory of rational option pricing *Bell Journal of Economics and Management Science* **4** 141–183

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: $\mathrm{M} \geq 1$.

3: N – INTEGER *Input*

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

Constraint: $\mathrm{N} \geq 1$.

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

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

Constraint: $\mathrm{X}(i) \geq z$ and $\mathrm{X}(i) \leq 1/z$, where $z = \mathrm{X02AMF}()$, 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: $\mathrm{S} \geq z$ and $\mathrm{S} \leq 1.0/z$, where $z = \mathrm{X02AMF}()$, the safe range parameter.

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

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

Constraint: $\mathrm{T}(i) \geq z$, where $z = \mathrm{X02AMF}()$, 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: $\mathrm{SIGMA} > 0.0$.

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: $\mathrm{R} \geq 0.0$.

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: $\mathrm{Q} \geq 0.0$.

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

On exit: $\mathrm{P}(i,j)$ 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 $j = 1, 2, \ldots, \mathrm{N}$.

11: LDP – INTEGER *Input*

On entry: the first dimension of the arrays P, DELTA, GAMMA, VEGA, THETA, RHO, CRHO, VANNA, CHARM, SPEED, COLOUR, ZOMMA and VOMMA as declared in the (sub)program from which S30ABF is called.

Constraint: LDP $\geq$ M.

12: DELTA(LDP, N) – REAL (KIND=nag_wp) array *Output*

On exit: the leading $\mathrm{M} \times \mathrm{N}$ 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.

13: GAMMA(LDP, N) – REAL (KIND=nag_wp) array *Output*

On exit: the leading $\mathrm{M} \times \mathrm{N}$ 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.

14: VEGA(LDP, N) – REAL (KIND=nag_wp) array *Output*

On exit: $\mathrm{VEGA}(i,j)$, 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 $i = 1, 2, \ldots, \mathrm{M}$ and $j = 1, 2, \ldots, \mathrm{N}$.

15: THETA(LDP, N) – REAL (KIND=nag_wp) array *Output*

On exit: $\mathrm{THETA}(i,j)$, 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 $i = 1, 2, \ldots, \mathrm{M}$ and $j = 1, 2, \ldots, \mathrm{N}$, where $b = r - q$.

16: RHO(LDP, N) – REAL (KIND=nag_wp) array *Output*

On exit: $\mathrm{RHO}(i,j)$, 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 $i = 1, 2, \ldots, \mathrm{M}$ and $j = 1, 2, \ldots, \mathrm{N}$.

17: CRHO(LDP, N) – REAL (KIND=nag_wp) array *Output*

On exit: $\mathrm{CRHO}(i,j)$, 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 $i = 1, 2, \ldots, \mathrm{M}$ and $j = 1, 2, \ldots, \mathrm{N}$, where $b = r - q$.

18: VANNA(LDP, N) – REAL (KIND=nag_wp) array *Output*

On exit: $\mathrm{VANNA}(i,j)$, 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 $i = 1, 2, \ldots, \mathrm{M}$ and $j = 1, 2, \ldots, \mathrm{N}$.

19: CHARM(LDP, N) – REAL (KIND=nag_wp) array *Output*

On exit: $\mathrm{CHARM}(i,j)$, 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 $i = 1, 2, \ldots, \mathrm{M}$ and $j = 1, 2, \ldots, \mathrm{N}$.

20: SPEED(LDP, N) – REAL (KIND=nag_wp) array *Output*

On exit: 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 $i = 1, 2, \ldots, \mathrm{M}$ and $j = 1, 2, \ldots, \mathrm{N}$.

21: COLOUR(LDP, N) – REAL (KIND=nag_wp) array *Output*

On exit: 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 $i = 1, 2, \ldots, \mathrm{M}$ and $j = 1, 2, \ldots, \mathrm{N}$.

22: ZOMMA(LDP, N) – REAL (KIND=nag_wp) array *Output*

On exit: 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 $i = 1, 2, \ldots, \mathrm{M}$ and $j = 1, 2, \ldots, \mathrm{N}$.

23: VOMMA(LDP, N) – REAL (KIND=nag_wp) array *Output*

On exit: 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 $i = 1, 2, \ldots, \mathrm{M}$ and $j = 1, 2, \ldots, \mathrm{N}$.

24: 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, $\mathrm{X}(\langle value\rangle) = \langle value\rangle$.
Constraint: $\mathrm{X}(i) \geq \langle value\rangle$ and $\mathrm{X}(i) \leq \langle value\rangle$.

IFAIL $= 5$

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

IFAIL $= 6$

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

IFAIL $= 7$

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

IFAIL $= 8$

On entry, $\mathrm{R} = \langle value\rangle$.
Constraint: $\mathrm{R} \geq 0.0$.

IFAIL $= 9$

On entry, $\mathrm{Q} = \langle value\rangle$.
Constraint: $\mathrm{Q} \geq 0.0$.

IFAIL $= 11$

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

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

S30ABF 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 European put with a time to expiry of 0.7 years, a stock price of 55 and a strike price of 60. The risk-free interest rate is 10% per year and the volatility is 30% per year.

10.1 Program Text

```
    Program s30abfe

!     S30ABF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: nag_wp, s30abf
!     .. 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  :: charm(:,:), colour(:,:), crho(:,:),  &
                                          delta(:,:), gamma(:,:), p(:,:),      &
                                          rho(:,:), speed(:,:), t(:),          &
                                          theta(:,:), vanna(:,:), vega(:,:),   &
                                          vomma(:,:), x(:), zomma(:,:)
!     .. Executable Statements ..
      Write (nout,*) 'S30ABF 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 (charm(ldp,n),colour(ldp,n),crho(ldp,n),delta(ldp,n),           &
        gamma(ldp,n),p(ldp,n),rho(ldp,n),speed(ldp,n),t(n),theta(ldp,n),       &
        vanna(ldp,n),vega(ldp,n),vomma(ldp,n),x(m),zomma(ldp,n))

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

      ifail = 0
      Call s30abf(calput,m,n,x,s,t,sigma,r,q,p,ldp,delta,gamma,vega,theta,rho, &
        crho,vanna,charm,speed,colour,zomma,vomma,ifail)

      Write (nout,*)

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

```
      Write (nout,99997) '  Spot       = ', s
      Write (nout,99997) '  Volatility = ', sigma
      Write (nout,99997) '  Rate       = ', r
      Write (nout,99997) '  Dividend   = ', q

      Write (nout,*)

      Do j = 1, n
        Write (nout,*)
        Write (nout,99999) t(j)
        Write (nout,*) '  Strike    Price    Delta    Gamma     Vega    ' //   &
          'Theta      Rho     CRho'

        Do i = 1, m
          Write (nout,99998) x(i), p(i,j), delta(i,j), gamma(i,j), vega(i,j),  &
            theta(i,j), rho(i,j), crho(i,j)
        End Do

        Write (nout,*) '  Strike    Price    Vanna    Charm    Speed    ' //    &
          'Colour    Zomma    Vomma'

        Do i = 1, m
          Write (nout,99998) x(i), p(i,j), vanna(i,j), charm(i,j), speed(i,j), &
            colour(i,j), zomma(i,j), vomma(i,j)
        End Do

      End Do

99999 Format (1X,'Time to Expiry : ',1X,F8.4)
99998 Format (1X,8(F8.4,1X))
99997 Format (A,1X,F8.4)
    End Program s30abfe
```

10.2 Program Data

```
S30ABF Example Program Data
 'P'                    : Call = 'C', Put = 'P'
 55.0 0.3 0.1 0.0       : S, SIGMA, R, Q
 1 1                    : M, N
60.0                    : X(I), I = 1,2,...M
0.7                     : T(I), I = 1,2,...N
```

10.3 Program Results

```
 S30ABF Example Program Results

 European Put :
  Spot       =   55.0000
  Volatility =    0.3000
  Rate       =    0.1000
  Dividend   =    0.0000


 Time to Expiry :    0.7000
   Strike    Price    Delta    Gamma     Vega    Theta      Rho     CRho
  60.0000   6.0245  -0.4770   0.0289  18.3273  -0.7014 -22.5811 -18.3639
   Strike    Price    Vanna    Charm    Speed   Colour    Zomma    Vomma
  60.0000   6.0245   0.2566  -0.2137  -0.0006   0.0215  -0.0972  -0.6816
```