

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,
1              GAMMA, VEGA, THETA, RHO, CRHO, VANNA, CHARM, SPEED,
2              COLOUR, ZOMMA, VOMMA, IFAIL)

INTEGER          M, N, LDP, IFAIL
double precision X(M), S, T(N), SIGMA, R, Q, P(LDP,N), DELTA(LDP,N),
1              GAMMA(LDP,N), VEGA(LDP,N), THETA(LDP,N), RHO(LDP,N),
2              CRHO(LDP,N), VANNA(LDP,N), CHARM(LDP,N), SPEED(LDP,N),
3              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

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

and the corresponding European put price is

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

and where Φ denotes the cumulative Normal distribution function,

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

and

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

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

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 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: $M \geq 1$.

- 3: N – INTEGER *Input*
On entry: the number of times to expiry to be used.
Constraint: $N \geq 1$.

- 4: X(M) – **double precision** array *Input*
On entry: X(*i*) must contain X_i , the *i*th strike price, for $i = 1, 2, \dots, M$.
Constraint: $X(i) \geq z$ and $X(i) \leq 1/z$, where $z = X02AMF()$, the safe range parameter, for $i = 1, 2, \dots, M$.

- 5: S – **double precision** *Input*
On entry: S, the price of the underlying asset.
Constraint: $S \geq z$ and $S \leq 1/z$, where $z = X02AMF()$, the safe range parameter.

- 6: T(N) – **double precision** array *Input*
On entry: T(*i*) must contain T_i , the *i*th time, in years, to expiry, for $i = 1, 2, \dots, N$.
Constraint: $T(i) \geq z$, where $z = X02AMF()$, the safe range parameter, for $i = 1, 2, \dots, N$.

- 7: SIGMA – **double precision** *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 – **double precision** *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: $R \geq 0.0$.

- 9: Q – **double precision** *Input*
On entry: q , the annual continuous yield rate. Note that a rate of 8% should be entered as 0.08.
Constraint: $Q \geq 0.0$.

- 10: P(LDP,N) – **double precision** array *Output*
On exit: the $m \times n$ array P contains the computed option prices.

- 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) – **double precision** array *Output*
On exit: 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.
- 13: GAMMA(LDP,N) – **double precision** array *Output*
On exit: 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.
- 14: VEGA(LDP,N) – **double precision** array *Output*
On exit: 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.
- 15: THETA(LDP,N) – **double precision** array *Output*
On exit: 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.
- 16: RHO(LDP,N) – **double precision** array *Output*
On exit: 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.
- 17: CRHO(LDP,N) – **double precision** array *Output*
On exit: 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 $b = r - q$.
- 18: VANNA(LDP,N) – **double precision** array *Output*
On exit: 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.
- 19: CHARM(LDP,N) – **double precision** array *Output*
On exit: 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.
- 20: SPEED(LDP,N) – **double precision** array *Output*
On exit: 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.
- 21: COLOUR(LDP,N) – **double precision** array *Output*
On exit: 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.

- 22: ZOMMA(LDP,N) – *double precision* array *Output*
On exit: 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.
- 23: VOMMA(LDP,N) – *double precision* array *Output*
On exit: the $m \times n$ array VOMMA contains the sensitivity, $\frac{\partial^2 P}{\partial \sigma}$, of VEGA to change in the volatility of the underlying asset.
- 24: 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.
On exit: IFAIL = 0 unless the routine detects an error (see Section 6).
 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.**

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 $\neq$ '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, 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 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.

9.1 Program Text

```
*      S30ABF Example Program Text
*      Mark 22 Release. NAG Copyright 2007.
*      .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          LDP, MMAX, NMAX
      PARAMETER        (LDP=50,MMAX=50,NMAX=50)
*      .. Local Scalars ..
      DOUBLE PRECISION Q, R, S, SIGMA
      INTEGER          I, IFAIL, J, M, N
      CHARACTER        PUT
*      .. Local Arrays ..
      DOUBLE PRECISION CHARM(LDP,NMAX), COLOUR(LDP,NMAX),
+                     CRHO(LDP,NMAX), DELTA(LDP,NMAX), GAMMA(LDP,NMAX),
+                     P(LDP,NMAX), RHO(LDP,NMAX), SPEED(LDP,NMAX),
+                     T(NMAX), THETA(LDP,NMAX), VANNA(LDP,NMAX),
+                     VEGA(LDP,NMAX), VOMMA(LDP,NMAX), X(MMAX),
+                     ZOMMA(LDP,NMAX)
*      .. External Subroutines ..
      EXTERNAL         S30ABF
*      .. Executable Statements ..
      WRITE (NOUT,*) 'S30ABF Example Program Results'
      WRITE (NOUT,*)
*      Skip heading in data file.
      READ (NIN,*)
*      Read problem parameters.
      READ (NIN,*) PUT
      READ (NIN,*) S, SIGMA, R, Q
      READ (NIN,*) M, N
*
      IF (M.LE.MMAX .AND. N.LE.NMAX) THEN
*      Read array of strike/exercise prices, X
      READ (NIN,*) (X(I),I=1,M)
*      Read array of times to expiry
      READ (NIN,*) (T(I),I=1,N)
*
      IFAIL = 1
*
      CALL S30ABF(PUT,M,N,X,S,T,SIGMA,R,Q,P,LDP,DELTA,GAMMA,VEGA,
+              THETA,RHO,CRHO,VANNA,CHARM,SPEED,COLOUR,ZOMMA,
+              VOMMA,IFAIL)
*
      IF (IFAIL.EQ.0) THEN
        IF (PUT.EQ.'C') THEN
```

```

        WRITE (NOUT,*) 'European Call : '
    ELSE IF (PUT.EQ.'P') THEN
        WRITE (NOUT,*) 'European Put : '
    END IF
    WRITE (NOUT, '(A,1X,F8.4)') ' Spot      = ', S
    WRITE (NOUT, '(A,1X,F8.4)') ' Volatility = ', SIGMA
    WRITE (NOUT, '(A,1X,F8.4)') ' Rate      = ', R
    WRITE (NOUT, '(A,1X,F8.4)') ' Dividend  = ', Q
*
    WRITE (NOUT,*)
    DO 60 J = 1, N
        WRITE (NOUT,*)
        WRITE (NOUT,99999) T(J)
        WRITE (NOUT,*)
+ ' Strike Price Delta Gamma Vega Theta Rho'
+ // ' CRho'
        DO 20 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)
20        CONTINUE
*
        WRITE (NOUT,*)
+ ' Strike Price Vanna Charm Speed Colour Zomma'
+ // ' Vomma'
        DO 40 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)
40        CONTINUE
*
60        CONTINUE
    ELSE
        WRITE (NOUT,*)
        WRITE (NOUT,99997) IFAIL
    END IF
*
    END IF
*
99999 FORMAT (1X,'Time to Expiry : ',1X,F8.4)
99998 FORMAT (1X,8(F8.4,1X))
99997 FORMAT (1X,' ** S30ABF returned with IFAIL = ',I5)
END

```

9.2 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

```

9.3 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