

# NAG Library Routine Document

## S30SBF

**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

S30SBF computes the Asian geometric continuous average-rate option price together with its sensitivities (Greeks).

### 2 Specification

```

SUBROUTINE S30SBF (CALPUT, M, N, X, S, T, SIGMA, R, B, 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, B, 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

S30SBF computes the price of an Asian geometric continuous average-rate 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. The annual volatility,  $\sigma$ , risk-free rate,  $r$ , and cost of carry,  $b$ , are constants (see Kemna and Vorst (1990)). For a given strike price,  $X$ , the price of a call option with underlying price,  $S$ , and time to expiry,  $T$ , is

$$P_{\text{call}} = Se^{(\bar{b}-r)T}\Phi(\bar{d}_1) - Xe^{-rT}\Phi(\bar{d}_2),$$

and the corresponding put option price is

$$P_{\text{put}} = Xe^{-rT}\Phi(-\bar{d}_2) - Se^{(\bar{b}-r)T}\Phi(-\bar{d}_1),$$

where

$$\bar{d}_1 = \frac{\ln(S/X) + (\bar{b} + \bar{\sigma}^2/2)T}{\bar{\sigma}\sqrt{T}}$$

and

$$\bar{d}_2 = \bar{d}_1 - \bar{\sigma}\sqrt{T},$$

with

$$\bar{\sigma} = \frac{\sigma}{\sqrt{3}}, \quad \bar{b} = \frac{1}{2}\left(b - \frac{\sigma^2}{6}\right).$$

$\Phi$  is the cumulative Normal distribution function,

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

The option price  $P_{ij} = P(X = X_i, T = T_j)$  is computed for each strike price in a set  $X_i$ ,  $i = 1, 2, \dots, m$ , and for each expiry time in a set  $T_j$ ,  $j = 1, 2, \dots, n$ .

## 4 References

Kemna A and Vorst A (1990) A pricing method for options based on average asset values *Journal of Banking and Finance* **14** 113–129

## 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:*  $M \geq 1$ .
- 3: N – INTEGER *Input*  
*On entry:* the number of times to expiry to be used.  
*Constraint:*  $N \geq 1$ .
- 4: X(M) – REAL (KIND=nag\_wp) 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 – 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 *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 – REAL (KIND=nag\_wp) *Input*  
*On entry:*  $\sigma$ , 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:*  $R \geq 0.0$ .
- 9: B – REAL (KIND=nag\_wp) *Input*  
*On entry:*  $b$ , the annual cost of carry rate. Note that a rate of 8% should be entered as 0.08.

- 10: 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  $i = 1, 2, \dots, M$  and  $j = 1, 2, \dots, N$ .
- 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 S30SBF is called.  
*Constraint:* LDP  $\geq$  M.
- 12: DELTA(LDP,N) – REAL (KIND=nag\_wp) array Output  
*On exit:* the leading  $M \times N$  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  $M \times N$  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:* 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  $i = 1, 2, \dots, M$  and  $j = 1, 2, \dots, N$ .
- 15: THETA(LDP,N) – REAL (KIND=nag\_wp) array Output  
*On exit:* 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  $i = 1, 2, \dots, M$  and  $j = 1, 2, \dots, N$ , where  $b = r - q$ .
- 16: RHO(LDP,N) – REAL (KIND=nag\_wp) array Output  
*On exit:* 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  $i = 1, 2, \dots, M$  and  $j = 1, 2, \dots, N$ .
- 17: CRHO(LDP,N) – REAL (KIND=nag\_wp) array Output  
*On exit:* DELTA( $i, j$ ), contains the first-order Greek measuring the sensitivity of the option price  $P_{ij}$  to change in the price of the underlying asset, i.e.,  $-\frac{\partial P_{ij}}{\partial S}$ , for  $i = 1, 2, \dots, M$  and  $j = 1, 2, \dots, N$ .
- 18: VANNA(LDP,N) – REAL (KIND=nag\_wp) array Output  
*On exit:* VANNA( $i, j$ ), contains the second-order Greek measuring the sensitivity of the first-order Greek  $\Delta_{ij}$  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  $i = 1, 2, \dots, M$  and  $j = 1, 2, \dots, N$ .
- 19: CHARM(LDP,N) – REAL (KIND=nag\_wp) array Output  
*On exit:* CHARM( $i, j$ ), contains the second-order Greek measuring the sensitivity of the first-order Greek  $\Delta_{ij}$  to change in the time, i.e.,  $-\frac{\partial \Delta_{ij}}{\partial T} = -\frac{\partial^2 P_{ij}}{\partial S \partial T}$ , for  $i = 1, 2, \dots, M$  and  $j = 1, 2, \dots, N$ .

- 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  $\Gamma_{ij}$  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  $i = 1, 2, \dots, M$  and  $j = 1, 2, \dots, N$ .
- 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  $\Gamma_{ij}$  to change in the time, i.e.,  $-\frac{\partial \Gamma_{ij}}{\partial T} = -\frac{\partial^3 P_{ij}}{\partial S \partial T}$ , for  $i = 1, 2, \dots, M$  and  $j = 1, 2, \dots, N$ .
- 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  $\Gamma_{ij}$  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  $i = 1, 2, \dots, M$  and  $j = 1, 2, \dots, N$ .
- 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  $\Delta_{ij}$  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  $i = 1, 2, \dots, M$  and  $j = 1, 2, \dots, N$ .
- 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:  $M \geq 1$ .

IFAIL = 3

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

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

IFAIL = 7

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

IFAIL = 8

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

IFAIL = 11

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,  $\Phi$ . 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 S15ABF and S15ADF). An accuracy close to *machine precision* can generally be expected.

## 8 Parallelism and Performance

S30SBF 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 an Asian geometric continuous average-rate call with a time to expiry of 3 months, a stock price of 80 and a strike price of 97. The risk-free interest rate is 5% per year, the cost of carry is 8% and the volatility is 20% per year.

### 10.1 Program Text

```

Program s30sbfe

!      S30SBF Example Program Text

!      Mark 26 Release. NAG Copyright 2016.

!      .. Use Statements ..
Use nag_library, Only: nag_wp, s30sbf
!      .. Implicit None Statement ..
Implicit None
!      .. Parameters ..
Integer, Parameter          :: nin = 5, nout = 6
!      .. Local Scalars ..
Real (Kind=nag_wp)         :: b, 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,*) 'S30SBF Example Program Results'

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

Read (nin,*) calput
Read (nin,*) s, sigma, r, b
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 s30sbf(calput,m,n,x,s,t,sigma,r,b,p,ldp,delta,gamma,vega,theta,rho, &
            crho,vanna,charm,speed,colour,zomma,vomma,ifail)

Write (nout,*)
Write (nout,*) 'Asian Option: Geometric Continuous Average-Rate'

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

Write (nout,99997) ' Spot          = ', s
Write (nout,99997) ' Volatility    = ', sigma

```

```

Write (nout,99997) ' Rate          = ', r
Write (nout,99997) ' Cost of carry = ', b

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 s30sbfe

```

## 10.2 Program Data

```

S30SBF Example Program Data
'C'          : Call = 'C', Put = 'P'
80.0 0.2 0.05 0.08 : S, SIGMA, R, B
1 1          : M, N
97.0         : X(I), I = 1,2,...M
0.25        : T(I), I = 1,2,...N

```

## 10.3 Program Results

S30SBF Example Program Results

Asian Option: Geometric Continuous Average-Rate

Asian Call :

```

Spot          = 80.0000
Volatility    = 0.2000
Rate          = 0.0500
Cost of carry = 0.0800

```

```

Time to Expiry : 0.2500
Strike   Price   Delta   Gamma   Vega   Theta   Rho   CRho
97.0000  0.0010  0.0008  0.0006  0.0638 -0.0281  0.0079  0.0081
Strike   Price   Vanna   Charm   Speed   Colour   Zomma   Vomma
97.0000  0.0010  0.0443 -0.0196  0.0004 -0.0122  0.0272  3.1893

```

---