

# NAG Library Routine Document

## S30BAF

**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

S30BAF computes the price of a floating-strike lookback option.

### 2 Specification

```
SUBROUTINE S30BAF (CALPUT, M, N, SM, S, T, SIGMA, R, Q, P, LDP, IFAIL)
INTEGER                M, N, LDP, IFAIL
REAL (KIND=nag_wp)    SM(M), S, T(N), SIGMA, R, Q, P(LDP,N)
CHARACTER(1)          CALPUT
```

### 3 Description

S30BAF computes the price of a floating-strike lookback call or put option. A call option of this type confers the right to buy the underlying asset at the lowest price,  $S_{\min}$ , observed during the lifetime of the contract. A put option gives the holder the right to sell the underlying asset at the maximum price,  $S_{\max}$ , observed during the lifetime of the contract. Thus, at expiry, the payoff for a call option is  $S - S_{\min}$ , and for a put,  $S_{\max} - S$ .

For a given minimum value the price of a floating-strike lookback call with underlying asset price,  $S$ , and time to expiry,  $T$ , is

$$P_{\text{call}} = Se^{-qT}\Phi(a_1) - S_{\min}e^{-rT}\Phi(a_2) + Se^{-rT} \frac{\sigma^2}{2b} \left[ \left( \frac{S}{S_{\min}} \right)^{-2b/\sigma^2} \Phi\left(-a_1 + \frac{2b}{\sigma}\sqrt{T}\right) - e^{bT}\Phi(-a_1) \right],$$

where  $b = r - q \neq 0$ . The volatility,  $\sigma$ , risk-free interest rate,  $r$ , and annualised dividend yield,  $q$ , are constants. When  $r = q$ , the option price is given by

$$P_{\text{call}} = Se^{-qT}\Phi(a_1) - S_{\min}e^{-rT}\Phi(a_2) + Se^{-rT}\sigma\sqrt{T}[\phi(a_1) + a_1(\Phi(a_1) - 1)].$$

The corresponding put price is (for  $b \neq 0$ ),

$$P_{\text{put}} = S_{\max}e^{-rT}\Phi(-a_2) - Se^{-qT}\Phi(-a_1) + Se^{-rT} \frac{\sigma^2}{2b} \left[ -\left( \frac{S}{S_{\max}} \right)^{-2b/\sigma^2} \Phi\left(a_1 - \frac{2b}{\sigma}\sqrt{T}\right) + e^{bT}\Phi(a_1) \right].$$

When  $r = q$ ,

$$P_{\text{put}} = S_{\max}e^{-rT}\Phi(-a_2) - Se^{-qT}\Phi(-a_1) + Se^{-rT}\sigma\sqrt{T}[\phi(a_1) + a_1\Phi(a_1)].$$

In the above,  $\Phi$  denotes the cumulative Normal distribution function,

$$\Phi(x) = \int_{-\infty}^x \phi(y)dy$$

where  $\phi$  denotes the standard Normal probability density function

$$\phi(y) = \frac{1}{\sqrt{2\pi}} \exp(-y^2/2)$$

and

$$a_1 = \frac{\ln(S/S_m) + (b + \sigma^2/2)T}{\sigma\sqrt{T}}$$

$$a_2 = a_1 - \sigma\sqrt{T}$$

where  $S_m$  is taken to be the minimum price attained by the underlying asset,  $S_{\min}$ , for a call and the maximum price,  $S_{\max}$ , for a put.

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

## 4 References

Goldman B M, Sosin H B and Gatto M A (1979) Path dependent options: buy at the low, sell at the high *Journal of Finance* **34** 1111–1127

## 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 minimum or maximum 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: SM(M) – REAL (KIND=nag\_wp) array *Input*  
*On entry:* SM( $i$ ) must contain  $S_{\min}(i)$ , the  $i$ th minimum observed price of the underlying asset when CALPUT = 'C', or  $S_{\max}(i)$ , the maximum observed price when CALPUT = 'P', for  $i = 1, 2, \dots, M$ .  
*Constraints:*  
 $SM(i) \geq z$  and  $SM(i) \leq 1/z$ , where  $z = X02AMF()$ , the safe range parameter, for  $i = 1, 2, \dots, M$ ;  
 if CALPUT = 'C',  $SM(i) \leq S$ , for  $i = 1, 2, \dots, M$ ;  
 if CALPUT = 'P',  $SM(i) \geq S$ , 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: 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  $\geq$  0.0.
- 10: P(LDP, N) – REAL (KIND=nag\_wp) array Output  
*On exit:* P(*i*, *j*) contains  $P_{ij}$ , the option price evaluated for the minimum or maximum observed price  $S_{\min}(i)$  or  $S_{\max}(i)$  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 array P as declared in the (sub)program from which S30BAF 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 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,  $SM(\langle value \rangle) = \langle value \rangle$ .  
Constraint:  $\langle value \rangle \leq SM(i) \leq \langle value \rangle$  for all  $i$ .  
On entry with a call option,  $SM(\langle value \rangle) = \langle value \rangle$ .  
Constraint: for call options,  $SM(i) \leq \langle value \rangle$  for all  $i$ .  
On entry with a put option,  $SM(\langle value \rangle) = \langle value \rangle$ .  
Constraint: for put options,  $SM(i) \geq \langle value \rangle$  for all  $i$ .

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:  $T(i) \geq \langle value \rangle$  for all  $i$ .

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 = 9

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

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

S30BAF 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 floating-strike lookback call with a time to expiry of 6 months and a stock price of 120. The minimum price observed so far is 100. The risk-free interest rate is 10% per year and the volatility is 30% per year with an annual dividend return of 6%.

### 10.1 Program Text

```

Program s30baf

!      S30BAF Example Program Text
!
!      Mark 26 Release. NAG Copyright 2016.
!
!      .. Use Statements ..
Use nag_library, Only: nag_wp, s30baf
!      .. 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(:,,:), sm(:,), t(:)
!      .. Executable Statements ..
Write (nout,*) 'S30BAF 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),sm(m),t(n))

```

```

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

ifail = 0
Call s30baf(calput,m,n,sm,s,t,sigma,r,q,p,ldp,ifail)

Write (nout,*)
Write (nout,*) 'Floating-strike Lookback'

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) sm(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 s30baf

```

## 10.2 Program Data

```

S30BAF Example Program Data
'C'      : Call = 'C', Put = 'P'
120.0 0.3 0.1 0.06 : S, SIGMA, R, Q
1 1      : M, N
100.0    : SM(I), I = 1,2,...M
0.5      : T(I), I = 1,2,...N

```

## 10.3 Program Results

S30BAF Example Program Results

```

Floating-strike Lookback
European Call :
Spot      = 120.0000
Volatility =  0.3000
Rate      =  0.1000
Dividend  =  0.0600

Strike    Expiry    Option Price
100.0000  0.5000        25.3534

```

---