NAG Library Routine Document

S30JAF

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

S30JAF computes the European option price using the Merton jump-diffusion model.

2 Specification

```
SUBROUTINE S30JAF (CALPUT, M, N, X, S, T, SIGMA, R, LAMBDA, JVOL, P,        &
                   LDP, IFAIL)
INTEGER              M, N, LDP, IFAIL
REAL (KIND=nag_wp) X(M), S, T(N), SIGMA, R, LAMBDA, JVOL, P(LDP,N)
CHARACTER(1)         CALPUT
```

3 Description

S30JAF uses Merton's jump-diffusion model (Merton (1976)) to compute the price of a European option. This assumes that the asset price is described by a Brownian motion with drift, as in the Black–Scholes–Merton case, together with a compound Poisson process to model the jumps. The corresponding stochastic differential equation is,

$$\frac{dS}{S} = (\alpha - \lambda k)dt + \hat{\sigma} dW_t + dq_t.$$

Here α is the instantaneous expected return on the asset price, S; $\hat{\sigma}^2$ is the instantaneous variance of the return when the Poisson event does not occur; dW_t is a standard Brownian motion; q_t is the independent Poisson process and $k = E[Y - 1]$ where $Y - 1$ is the random variable change in the stock price if the Poisson event occurs and E is the expectation operator over the random variable Y.

This leads to the following price for a European option (see Haug (2007))

$$P_{\text{call}} = \sum_{j=0}^{\infty} \frac{e^{-\lambda T}(\lambda T)^j}{j!} C_j\left(S, X, T, r, \sigma'_j\right),$$

where T is the time to expiry; X is the strike price; r is the annual risk-free interest rate; $C_j\left(S, X, T, r, \sigma'_j\right)$ is the Black–Scholes–Merton option pricing formula for a European call (see S30AAF).

$$\sigma'_j = \sqrt{z^2 + \delta^2\left(\tfrac{j}{T}\right)},$$
$$z^2 = \sigma^2 - \lambda\delta^2,$$
$$\delta^2 = \tfrac{\gamma\sigma^2}{\lambda},$$

where σ is the total volatility including jumps; λ is the expected number of jumps given as an average per year; γ is the proportion of the total volatility due to jumps.

The value of a put is obtained by substituting the Black–Scholes–Merton put price for $C_j\left(S, X, T, r, \sigma'_j\right)$.

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

Haug E G (2007) *The Complete Guide to Option Pricing Formulas* (2nd Edition) McGraw-Hill

Merton R C (1976) Option pricing when underlying stock returns are discontinuous *Journal of Financial Economics* **3** 125–144

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 ≥ 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 $i = 1, 2, \ldots,$ M.

Constraint: X(i) $\geq z$ and X(i) $\leq 1/z$, where $z =$ X02AMF(), the safe range parameter, for $i = 1, 2, \ldots,$ 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 ith time, in years, to expiry, for $i = 1, 2, \ldots,$ N.

Constraint: T(i) $\geq z$, where $z =$ X02AMF(), the safe range parameter, for $i = 1, 2, \ldots,$ N.

7: SIGMA – REAL (KIND=nag_wp) *Input*

On entry: σ, the annual total volatility, including jumps.

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 ≥ 0.0.

9: LAMBDA – REAL (KIND=nag_wp) *Input*

On entry: λ, the number of expected jumps per year.

Constraint: LAMBDA > 0.0.

10: JVOL – REAL (KIND=nag_wp) *Input*

On entry: the proportion of the total volatility associated with jumps.

Constraint: $0.0 \leq$ JVOL < 1.0.

11: 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 $j = 1, 2, \ldots, \mathrm{N}$.

12: LDP – INTEGER *Input*

On entry: the first dimension of the array P as declared in the (sub)program from which S30JAF 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 parameter you should refer to Section 3.3 in the Essential Introduction 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 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.**

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, $\mathrm{S} = \langle value \rangle$.
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, $\mathrm{LAMBDA} = \langle value \rangle$.
Constraint: $\mathrm{LAMBDA} > 0.0$.

IFAIL $= 10$

On entry, $\mathrm{JVOL} = \langle value \rangle$.
Constraint: $\mathrm{JVOL} \geq 0.0$ and $\mathrm{JVOL} < 1.0$.

IFAIL $= 12$

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 3.8 in the Essential Introduction for further information.

IFAIL $= -399$

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

See Section 3.7 in the Essential Introduction for further information.

IFAIL $= -999$

Dynamic memory allocation failed.

See Section 3.6 in the Essential Introduction for further information.

7 Accuracy

The accuracy of the output is dependent on the accuracy of the cumulative Normal distribution function, Φ, occurring in C_j. 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

S30JAF is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

S30JAF makes calls to BLAS and/or LAPACK routines, which may be threaded within the vendor library used by this implementation. Consult the documentation for the vendor library for further information.

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 call with jumps. The time to expiry is 3 months, the stock price is 45 and the strike price is 55. The number of jumps per year is 3 and the percentage of the total volatility due to jumps is 40%. The risk-free interest rate is 10% per year and the total volatility is 25% per year.

10.1 Program Text

```
    Program s30jafe

!     S30JAF Example Program Text

!     Mark 25 Release. NAG Copyright 2014.

!     .. Use Statements ..
      Use nag_library, Only: nag_wp, s30jaf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: jvol, lambda, 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,*) 'S30JAF Example Program Results'

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

      Read (nin,*) calput
      Read (nin,*) lambda
      Read (nin,*) s, sigma, r, jvol
      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 s30jaf(calput,m,n,x,s,t,sigma,r,lambda,jvol,p,ldp,ifail)

      Write (nout,*)
      Write (nout,*) 'Merton Jump-Diffusion Model'

      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) '  Jumps      = ', lambda
      Write (nout,99998) '  Jump vol   = ', jvol

      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 s30jafe
```

10.2 Program Data

```
S30JAF Example Program Data
 'C'                        : Call = 'C', Put = 'P'
 3.0                        : LAMBDA (jumps)
 45.0 0.25 0.1 0.4          : S, SIGMA, R, JVOL
 1 1                        : M, N
 55.0                       : X(I), I = 1,2,...M
 0.25                       : T(I), I = 1,2,...N
```

10.3 Program Results

```
 S30JAF Example Program Results

 Merton Jump-Diffusion Model
 European Call :
  Spot       =   45.0000
  Volatility =    0.2500
  Rate       =    0.1000
  Jumps      =    3.0000
  Jump vol   =    0.4000

    Strike    Expiry    Option Price
   55.0000    0.2500          0.2417
```