NAG Library Routine Document

G01SEF

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

G01SEF computes a number of lower or upper tail probabilities for the beta distribution.

2 Specification

```
SUBROUTINE G01SEF (LTAIL, TAIL, LBETA, BETA, LA, A, LB, B, P, IVALID,      &
                   IFAIL)
INTEGER            LTAIL, LBETA, LA, LB, IVALID(*), IFAIL
REAL (KIND=nag_wp) BETA(LBETA), A(LA), B(LB), P(*)
CHARACTER(1)       TAIL(LTAIL)
```

3 Description

The lower tail probability, $P\left(B_i \leq \beta_i : a_i, b_i\right)$ is defined by

$$P\left(B_i \leq \beta_i : a_i, b_i\right) = \frac{\Gamma(a_i + b_i)}{\Gamma(a_i)\Gamma(b_i)} \int_0^{\beta_i} {B_i}^{a_i - 1}(1 - B_i)^{b_i - 1}\, dB_i = I_{\beta_i}(a_i, b_i), \quad 0 \leq \beta_i \leq 1; \quad a_i, b_i > 0.$$

The function $I_{\beta_i}(a_i, b_i)$, also known as the incomplete beta function is calculated using S14CCF.

The input arrays to this routine are designed to allow maximum flexibility in the supply of vector arguments by re-using elements of any arrays that are shorter than the total number of evaluations required. See Section 2.6 in the G01 Chapter Introduction for further information.

4 References

Abramowitz M and Stegun I A (1972) *Handbook of Mathematical Functions* (3rd Edition) Dover Publications

Hastings N A J and Peacock J B (1975) *Statistical Distributions* Butterworth

Majumder K L and Bhattacharjee G P (1973) Algorithm AS 63. The incomplete beta integral *Appl. Statist.* **22** 409–411

5 Arguments

1: LTAIL – INTEGER *Input*

On entry: the length of the array TAIL.

Constraint: LTAIL > 0.

2: TAIL(LTAIL) – CHARACTER(1) array *Input*

On entry: indicates whether a lower or upper tail probabilities are required. For $j = ((i - 1) \bmod \text{LTAIL}) + 1$, for $i = 1, 2, \ldots, \max(\text{LTAIL}, \text{LBETA}, \text{LA}, \text{LB})$:

TAIL(j) = 'L'
The lower tail probability is returned, i.e., $p_i = P\left(B_i \leq \beta_i : a_i, b_i\right)$.

TAIL(j) = 'U'
The upper tail probability is returned, i.e., $p_i = P(B_i \geq \beta_i : a_i, b_i)$.

Constraint: TAIL(j) = 'L' or 'U', for $j = 1, 2, \ldots,$ LTAIL.

3: LBETA – INTEGER *Input*

On entry: the length of the array BETA.

Constraint: LBETA > 0.

4: BETA(LBETA) – REAL (KIND=nag_wp) array *Input*

On entry: β_i, the value of the beta variate with $\beta_i =$ BETA(j), $j = ((i-1) \bmod \text{LBETA}) + 1$.

Constraint: $0.0 \leq$ BETA$(j) \leq 1.0$, for $j = 1, 2, \ldots,$ LBETA.

5: LA – INTEGER *Input*

On entry: the length of the array A.

Constraint: LA > 0.

6: A(LA) – REAL (KIND=nag_wp) array *Input*

On entry: a_i, the first parameter of the required beta distribution with $a_i =$ A(j), $j = ((i-1) \bmod \text{LA}) + 1$.

Constraint: A$(j) > 0.0$, for $j = 1, 2, \ldots,$ LA.

7: LB – INTEGER *Input*

On entry: the length of the array B.

Constraint: LB > 0.

8: B(LB) – REAL (KIND=nag_wp) array *Input*

On entry: b_i, the second parameter of the required beta distribution with $b_i =$ B(j), $j = ((i-1) \bmod \text{LB}) + 1$.

Constraint: B$(j) > 0.0$, for $j = 1, 2, \ldots,$ LB.

9: P$(*)$ – REAL (KIND=nag_wp) array *Output*

Note: the dimension of the array P must be at least max(LTAIL, LBETA, LA, LB).

On exit: p_i, the probabilities for the beta distribution.

10: IVALID$(*)$ – INTEGER array *Output*

Note: the dimension of the array IVALID must be at least max(LTAIL, LBETA, LA, LB).

On exit: IVALID(i) indicates any errors with the input arguments, with

IVALID$(i) = 0$
No error.

IVALID$(i) = 1$

On entry, invalid value supplied in TAIL when calculating p_i.

IVALID$(i) = 2$

On entry, $\beta_i < 0.0$,
or $\beta_i > 1.0$.

IVALID$(i) = 3$

On entry, $a_i \leq 0.0$,
or $b_i \leq 0.0$,

11: 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, because for this routine the values of the output arguments may be useful even if IFAIL $\neq 0$ on exit, the recommended value is -1. **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).

Note: G01SEF may return useful information for one or more of the following detected errors or warnings.

Errors or warnings detected by the routine:

IFAIL $= 1$

On entry, at least one value of BETA, A, B or TAIL was invalid.
Check IVALID for more information.

IFAIL $= 2$

On entry, array size $= \langle value \rangle$.
Constraint: LTAIL > 0.

IFAIL $= 3$

On entry, array size $= \langle value \rangle$.
Constraint: LBETA > 0.

IFAIL $= 4$

On entry, array size $= \langle value \rangle$.
Constraint: LA > 0.

IFAIL $= 5$

On entry, array size $= \langle value \rangle$.
Constraint: LB > 0.

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 is limited by the error in the incomplete beta function. See Section 7 in S14CCF for further details.

8 Parallelism and Performance

G01SEF is not threaded in any implementation.

9 Further Comments

None.

10 Example

This example reads values from a number of beta distributions and computes the associated lower tail probabilities.

10.1 Program Text

```
    Program g01sefe
!     G01SEF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: g01sef, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Integer                          :: i, ifail, la, lb, lbeta, lout, ltail
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: a(:), b(:), beta(:), p(:)
      Integer, Allocatable             :: ivalid(:)
      Character (1), Allocatable       :: tail(:)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: max, mod, repeat
!     .. Executable Statements ..
      Write (nout,*) 'G01SEF Example Program Results'
      Write (nout,*)

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

!     Read in the input vectors
      Read (nin,*) ltail
      Allocate (tail(ltail))
      Read (nin,*) tail(1:ltail)

      Read (nin,*) lbeta
      Allocate (beta(lbeta))
      Read (nin,*) beta(1:lbeta)

      Read (nin,*) la
      Allocate (a(la))
      Read (nin,*) a(1:la)
```

```
      Read (nin,*) lb
      Allocate (b(lb))
      Read (nin,*) b(1:lb)

!     Allocate memory for output
      lout = max(ltail,lbeta,la,lb)
      Allocate (p(lout),ivalid(lout))

!     Calculate probability
      ifail = -1
      Call g01sef(ltail,tail,lbeta,beta,la,a,lb,b,p,ivalid,ifail)

      If (ifail==0 .Or. ifail==1) Then
!       Display titles
        Write (nout,*)                                                    &
          '  TAIL    BETA        A         B        P      IVALID'
        Write (nout,*) repeat('-',56)

!       Display results
        Do i = 1, lout
          Write (nout,99999) tail(mod(i-1,ltail)+1), beta(mod(i-1,lbeta)+1),   &
            a(mod(i-1,la)+1), b(mod(i-1,lb)+1), p(i), ivalid(i)
        End Do
      End If

99999 Format (5X,A1,3(4X,F6.2),4X,F6.3,4X,I3)
    End Program g01sefe
```

10.2 Program Data

```
G01SEF Example Program Data
1                       :: LTAIL
'L'                     :: TAIL
3                       :: LBETA
0.26 0.75 0.5           :: BETA
3                       :: LA
1.0 1.5 2.0             :: A
3                       :: LB
2.0 1.5 1.0             :: B
```

10.3 Program Results

```
 G01SEF Example Program Results

   TAIL    BETA        A         B        P      IVALID
 ------------------------------------------------------
     L      0.26      1.00      2.00     0.452      0
     L      0.75      1.50      1.50     0.804      0
     L      0.50      2.00      1.00     0.250      0
```