NAG Library Routine Document

G01TFF

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

G01TFF returns a number of deviates associated with given probabilities of the gamma distribution.

2 Specification

```
SUBROUTINE G01TFF (LTAIL, TAIL, LP, P, LA, A, LB, B, TOL, G, IVALID,       &
                   IFAIL)
INTEGER             LTAIL, LP, LA, LB, IVALID(*), IFAIL
REAL (KIND=nag_wp) P(LP), A(LA), B(LB), TOL, G(*)
CHARACTER(1)        TAIL(LTAIL)
```

3 Description

The deviate, g_{p_i}, associated with the lower tail probability, p_i, of the gamma distribution with shape parameter α_i and scale parameter β_i, is defined as the solution to

$$P\left(G_i \leq g_{p_i} : \alpha_i, \beta_i\right) = p_i = \frac{1}{{\beta_i}^{\alpha_i}\Gamma(\alpha_i)}\int_0^{g_{p_i}} {e_i}^{-G_i/\beta_i} {G_i}^{\alpha_i - 1}\, dG_i, \quad 0 \leq g_{p_i} < \infty;\ \alpha_i, \beta_i > 0.$$

The method used is described by Best and Roberts (1975) making use of the relationship between the gamma distribution and the χ^2-distribution.

Let $y_i = 2\dfrac{g_{p_i}}{\beta_i}$. The required y_i is found from the Taylor series expansion

$$y_i = y_0 + \sum_r \frac{C_r(y_0)}{r!}\left(\frac{E_i}{\phi(y_0)}\right)^r,$$

where y_0 is a starting approximation

$$C_1(u_i) = 1,$$

$$C_{r+1}(u_i) = \left(r\Psi + \frac{d}{du_i}\right)C_r(u_i),$$

$$\Psi_i = \tfrac{1}{2} - \frac{\alpha_i - 1}{u_i},$$

$$E_i = p_i - \int_0^{y_0} \phi_i(u_i)\, du_i,$$

$$\phi_i(u_i) = \frac{1}{2^{\alpha_i}\Gamma(\alpha_i)} {e_i}^{-u_i/2} {u_i}^{\alpha_i - 1}.$$

For most values of p_i and α_i the starting value

$$y_{01} = 2\alpha_i\left(z_i\sqrt{\frac{1}{9\alpha_i}} + 1 - \frac{1}{9\alpha_i}\right)^3$$

is used, where z_i is the deviate associated with a lower tail probability of p_i for the standard Normal distribution.

For p_i close to zero,

$$y_{02} = (p_i \alpha_i 2^{\alpha_i} \Gamma(\alpha_i))^{1/\alpha_i}$$

is used.

For large p_i values, when $y_{01} > 4.4\alpha_i + 6.0$,

$$y_{03} = -2\left[\ln(1-p_i) - (\alpha_i - 1)\ln\left(\tfrac{1}{2}y_{01}\right) + \ln(\Gamma(\alpha_i))\right]$$

is found to be a better starting value than y_{01}.

For small α_i ($\alpha_i \leq 0.16$), p_i is expressed in terms of an approximation to the exponential integral and y_{04} is found by Newton–Raphson iterations.

Seven terms of the Taylor series are used to refine the starting approximation, repeating the process if necessary until the required accuracy is obtained.

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

Best D J and Roberts D E (1975) Algorithm AS 91. The percentage points of the χ^2 distribution *Appl. Statist.* **24** 385–388

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 which tail the supplied probabilities represent. For $j = ((i-1) \bmod \text{LTAIL}) + 1$, for $i = 1, 2, \ldots, \max(\text{LTAIL}, \text{LP}, \text{LA}, \text{LB})$:

TAIL(j) = 'L'
The lower tail probability, i.e., $p_i = P\left(G_i \leq g_{p_i} : \alpha_i, \beta_i\right)$.

TAIL(j) = 'U'
The upper tail probability, i.e., $p_i = P\left(G_i \geq g_{p_i} : \alpha_i, \beta_i\right)$.

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

3: LP – INTEGER *Input*

On entry: the length of the array P.

Constraint: LP > 0.

4: P(LP) – REAL (KIND=nag_wp) array *Input*

On entry: p_i, the probability of the required gamma distribution as defined by TAIL with $p_i = \text{P}(j)$, $j = ((i-1) \bmod \text{LP}) + 1$.

Constraints:

if TAIL(k) = 'L', $0.0 \leq \text{P}(j) < 1.0$;
otherwise $0.0 < \text{P}(j) \leq 1.0$.

Where $k = (i-1) \bmod \text{LTAIL} + 1$ and $j = (i-1) \bmod \text{LP} + 1$.

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: α_i, the first parameter of the required gamma distribution with $\alpha_i = \mathrm{A}(j)$, $j = ((i-1) \bmod \mathrm{LA}) + 1$.

Constraint: $0.0 < \mathrm{A}(j) \leq 10^6$, for $j = 1, 2, \ldots, \mathrm{LA}$.

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: β_i, the second parameter of the required gamma distribution with $\beta_i = \mathrm{B}(j)$, $j = ((i-1) \bmod \mathrm{LB}) + 1$.

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

9: TOL – REAL (KIND=nag_wp) *Input*

On entry: the relative accuracy required by you in the results. If G01TFF is entered with TOL greater than or equal to 1.0 or less than $10 \times$ ***machine precision*** (see X02AJF), then the value of $10 \times$ ***machine precision*** is used instead.

10: G($*$) – REAL (KIND=nag_wp) array *Output*

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

On exit: g_{p_i}, the deviates for the gamma distribution.

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

Note: the dimension of the array IVALID must be at least max(LTAIL, LP, 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 g_{p_i}.

IVALID(i) $= 2$

On entry, invalid value for p_i.

IVALID(i) $= 3$

On entry, $\alpha_i \leq 0.0$,
or $\alpha_i > 10^6$,
or $\beta_i \leq 0.0$.

IVALID(i) $= 4$
p_i is too close to 0.0 or 1.0 to enable the result to be calculated.

IVALID(i) $= 5$
The solution has failed to converge. The result may be a reasonable approximation.

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, 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: G01TFF 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 TAIL, P, A, or B 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: LP > 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

In most cases the relative accuracy of the results should be as specified by TOL. However, for very small values of α_i or very small values of p_i there may be some loss of accuracy.

8 Parallelism and Performance

G01TFF is not threaded in any implementation.

9 Further Comments

None.

10 Example

This example reads lower tail probabilities for several gamma distributions, and calculates and prints the corresponding deviates until the end of data is reached.

10.1 Program Text

```
    Program g01tffe
!     G01TFF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: g01tff, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: tol
      Integer                          :: i, ifail, la, lb, lout, lp, ltail
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: a(:), b(:), g(:), p(:)
      Integer, Allocatable             :: ivalid(:)
      Character (1), Allocatable       :: tail(:)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: max, mod, repeat
!     .. Executable Statements ..
      Write (nout,*) 'G01TFF Example Program Results'
      Write (nout,*)

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

!     Read in the tolerance
      Read (nin,*) tol

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

      Read (nin,*) lp
      Allocate (p(lp))
      Read (nin,*) p(1:lp)

      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,lp,la,lb)
      Allocate (g(lout),ivalid(lout))

!     Calculate deviates (inverse CDF)
      ifail = -1
      Call g01tff(ltail,tail,lp,p,la,a,lb,b,tol,g,ivalid,ifail)

      If (ifail==0 .Or. ifail==1) Then
!       Display titles
        Write (nout,*) '   TAIL    P         A         B         G     IVALID'
        Write (nout,*) repeat('-',55)

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

99999 Format (5X,A,4X,F6.3,2(4X,F6.2),3X,F7.3,4X,I3)
    End Program g01tffe
```

10.2 Program Data

```
G01TFF Example Program Data
0.0                     :: TOL
1                       :: LTAIL
'L'                     :: TAIL
3                       :: LP
0.01 0.428 0.869        :: P
3                       :: LA
1.0 7.500 45.0          :: A
3                       :: LB
20.0 0.1 10.0           :: B
```

10.3 Program Results

```
 G01TFF Example Program Results

    TAIL    P         A         B         G     IVALID
 -------------------------------------------------------
     L     0.010      1.00     20.00     0.201      0
     L     0.428      7.50      0.10     0.670      0
     L     0.869     45.00     10.00   525.839      0
```