NAG Library Routine Document

G01TDF

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

G01TDF returns a number of deviates associated with given probabilities of the F or variance-ratio distribution with real degrees of freedom.

2 Specification

```
SUBROUTINE G01TDF (LTAIL, TAIL, LP, P, LDF1, DF1, LDF2, DF2, F, IVALID,        &
                   IFAIL)
INTEGER            LTAIL, LP, LDF1, LDF2, IVALID(*), IFAIL
REAL (KIND=nag_wp) P(LP), DF1(LDF1), DF2(LDF2), F(*)
CHARACTER(1)       TAIL(LTAIL)
```

3 Description

The deviate, f_{p_i}, associated with the lower tail probability, p_i, of the F-distribution with degrees of freedom u_i and v_i is defined as the solution to

$$P\left(F_i \leq f_{p_i} : u_i, v_i\right) = p_i = \frac{u_i^{\frac{1}{2}u_i} v_i^{\frac{1}{2}v_i} \Gamma\left(\frac{u_i+v_i}{2}\right)}{\Gamma\left(\frac{u_i}{2}\right)\Gamma\left(\frac{v_i}{2}\right)} \int_0^{f_{p_i}} F_i^{\frac{1}{2}(u_i-2)} \left(v_i + u_i F_i\right)^{-\frac{1}{2}(u_i+v_i)} \, dF_i,$$

where $u_i, v_i > 0$; $0 \leq f_{p_i} < \infty$.

The value of f_{p_i} is computed by means of a transformation to a beta distribution, $P_{i\beta_i}\left(B_i \leq \beta_i : a_i, b_i\right)$:

$$P\left(F_i \leq f_{p_i} : u_i, v_i\right) = P_{i\beta_i}\left(B_i \leq \frac{u_i f_{p_i}}{u_i f_{p_i} + v_i} : u_i/2, v_i/2\right)$$

and using a call to G01TEF.

For very large values of both u_i and v_i, greater than 10^5, a Normal approximation is used. If only one of u_i or v_i is greater than 10^5 then a χ^2 approximation is used; see Abramowitz and Stegun (1972).

The input arrays to this routine are designed to allow maximum flexibility in the supply of vector parameters 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

5 Parameters

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 $i = 1, 2, \ldots, \max(\text{LTAIL}, \text{LP}, \text{LDF1}, \text{LDF2})$:

TAIL(j) = 'L'
The lower tail probability, i.e., $p_i = P\left(F_i \leq f_{p_i} : u_i, v_i\right)$.

TAIL(j) = 'U'
The upper tail probability, i.e., $p_i = P\left(F_i \geq f_{p_i} : u_i, v_i\right)$.

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 F-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: LDF1 – INTEGER *Input*

On entry: the length of the array DF1.

Constraint: LDF1 > 0.

6: DF1(LDF1) – REAL (KIND=nag_wp) array *Input*

On entry: u_i, the degrees of freedom of the numerator variance with $u_i = \text{DF1}(j)$, $j = ((i-1) \bmod \text{LDF1}) + 1$.

Constraint: DF1$(j) > 0.0$, for $j = 1, 2, \ldots, \text{LDF1}$.

7: LDF2 – INTEGER *Input*

On entry: the length of the array DF2.

Constraint: LDF2 > 0.

8: DF2(LDF2) – REAL (KIND=nag_wp) array *Input*

On entry: v_i, the degrees of freedom of the denominator variance with $v_i = \text{DF2}(j)$, $j = ((i-1) \bmod \text{LDF2}) + 1$.

Constraint: DF2$(j) > 0.0$, for $j = 1, 2, \ldots, \text{LDF2}$.

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

Note: the dimension of the array F must be at least $\max(\text{LTAIL}, \text{LP}, \text{LDF1}, \text{LDF2})$.

On exit: f_{p_i}, the deviates for the F-distribution.

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

Note: the dimension of the array IVALID must be at least max(LTAIL, LP, LDF1, LDF2).

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 f_{p_i}.

IVALID(i) $= 2$

On entry, invalid value for p_i.

IVALID(i) $= 3$

On entry, $u_i \leq 0.0$,
or $v_i \leq 0.0$.

IVALID(i) $= 4$
The solution has not converged. The result should still be a reasonable approximation to the solution.

IVALID(i) $= 5$
The value of p_i is too close to 0.0 or 1.0 for the result to be computed. This will only occur when the large sample approximations are used.

11: 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, because for this routine the values of the output parameters 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: G01TDF 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, DF1, DF2 was invalid, or the solution failed to converge. 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: LDF1 > 0.

IFAIL $= 5$

On entry, array size $= \langle value \rangle$.
Constraint: LDF2 > 0.

IFAIL $= -999$

Dynamic memory allocation failed.

7 Accuracy

The result should be accurate to five significant digits.

8 Further Comments

For higher accuracy G01TEF can be used along with the transformations given in Section 3.

9 Example

This example reads the lower tail probabilities for several F-distributions, and calculates and prints the corresponding deviates.

9.1 Program Text

```
    Program g01tdfe
!     G01TDF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: g01tdf, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Integer                          :: i, ifail, ldf1, ldf2, lout, lp, ltail
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: df1(:), df2(:), f(:), p(:)
      Integer, Allocatable             :: ivalid(:)
      Character (1), Allocatable       :: tail(:)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: max, mod, repeat
!     .. Executable Statements ..
      Write (nout,*) 'G01TDF 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,*) lp
      Allocate (p(lp))
      Read (nin,*) p(1:lp)

      Read (nin,*) ldf1
      Allocate (df1(ldf1))
```

```
      Read (nin,*) df1(1:ldf1)

      Read (nin,*) ldf2
      Allocate (df2(ldf2))
      Read (nin,*) df2(1:ldf2)

!     Allocate memory for output
      lout = max(ltail,lp,ldf1,ldf2)
      Allocate (f(lout),ivalid(lout))

!     Calculate deviates (inverse CDF)
      ifail = -1
      Call g01tdf(ltail,tail,lp,p,ldf1,df1,ldf2,df2,f,ivalid,ifail)

      If (ifail==0 .Or. ifail==1) Then
!       Display titles
        Write (nout,*) &
          '   TAIL      P        DF1       DF2       F      IVALID'
        Write (nout,*) repeat('-',57)

!       Display results
        Do i = 1, lout
          Write (nout,99999) tail(mod(i-1,ltail)+1), p(mod(i-1,lp)+1), &
            df1(mod(i-1,ldf1)+1), df2(mod(i-1,ldf2)+1), f(i), ivalid(i)
        End Do
      End If

99999 Format (5X,A1,4X,F6.3,2(4X,F6.2),3X,F7.3,4X,I3)
    End Program g01tdfe
```

9.2 Program Data

```
G01TDF Example Program Data
1                       :: LTAIL
'L'                     :: TAIL
3                       :: LP
0.984 0.9 0.534         :: P
3                       :: LDF1
10.0 1.0 20.25          :: DF1
3                       :: LDF2
25.5 1.0 1.0            :: DF2
```

9.3 Program Results

```
 G01TDF Example Program Results

    TAIL      P        DF1       DF2       F      IVALID
 ------------------------------------------------------------
     L     0.984     10.00     25.50     2.847      0
     L     0.900      1.00      1.00    39.863      0
     L     0.534     20.25      1.00     2.498      0
```