

NAG C Library Function Document

nag_prob_non_central_f_dist (g01gdc)

1 Purpose

`nag_prob_non_central_f_dist (g01gdc)` returns the probability associated with the lower tail of the non-central F or variance-ratio distribution.

2 Specification

```
#include <nag.h>
#include <nagg01.h>

double nag_prob_non_central_f_dist(double f, double df1, double df2,
                                   double lambda, double tol, Integer max_iter, NagError *fail)
```

3 Description

The lower tail probability of the non-central F -distribution with ν_1 and ν_2 degrees of freedom and non-centrality parameter λ , $P(F \leq f : \nu_1, \nu_2; \lambda)$, is defined by

$$P(F \leq f : \nu_1, \nu_2; \lambda) = \int_0^x p(F : \nu_1, \nu_2; \lambda) dF$$

where

$$\begin{aligned} P(F : \nu_1, \nu_2; \lambda) = \sum_{j=0}^{\infty} e^{-\lambda/2} \frac{(\lambda/2)^j}{j!} \times \frac{(\nu_1 + 2j)^{(\nu_1+2j)/2} \nu_2^{\nu_2/2}}{B((\nu_1 + 2j)/2, \nu_2/2)} \\ \times u^{(\nu_1+2j-2)/2} [\nu_2 + (\nu_1 + 2j)u]^{-(\nu_1+2j+\nu_2)/2} \end{aligned}$$

and $B(\cdot, \cdot)$ is the beta function.

The probability is computed by means of a transformation to a non-central beta distribution;

$$P(F \leq f : \nu_1, \nu_2; \lambda) = P_\beta(X \leq x : a, b; \lambda)$$

where $x = \frac{\nu_1 f}{\nu_1 f + \nu_2}$ and $P_\beta(X \leq x : a, b; \lambda)$ is the lower tail probability integral of the non-central beta distribution with parameters a , b , and λ .

If ν_2 is very large, greater than 10^6 , then a χ^2 approximation is used.

4 Parameters

- | | |
|--|--|
| <p>1: f – double
 <i>On entry:</i> the deviate from the non-central F-distribution, f.
 <i>Constraint:</i> $\mathbf{f} > 0$.</p> <p>2: df1 – double
 <i>On entry:</i> the degrees of freedom of the numerator variance, ν_1.
 <i>Constraint:</i> $0.0 < \mathbf{df1} \leq 1.0e6$.</p> <p>3: df2 – double
 <i>On entry:</i> the degrees of freedom of the denominator variance, ν_2.
 <i>Constraint:</i> $\mathbf{df2} > 0.0$.</p> | <i>Input</i>
<i>Input</i>
<i>Input</i> |
|--|--|

4:	lambda – double	<i>Input</i>
<i>On entry:</i> the non-centrality parameter, λ .		
<i>Constraint:</i> $0.0 \leq \text{lambda} \leq -2.0 \times \log(U)$ where U is the safe range parameter as defined by nag_real_safe_small_number (X02AMC).		
5:	tol – double	<i>Input</i>
<i>On entry:</i> the relative accuracy required by the user in the results. If nag_prob_non_central_f_dist is entered with tol greater than or equal to 1.0 or less than $10 \times \text{machine precision}$ (see nag_machine_precision (X02AJC)), then the value of $10 \times \text{machine precision}$ is used instead.		
6:	max_iter – Integer	<i>Input</i>
<i>On entry:</i> the maximum number of iterations to be used.		
<i>Suggested value:</i> 500. See nag_prob_non_central_chi_sq (g01gcc) and nag_prob_non_central_beta_dist (g01gec) for further details.		
<i>Constraint:</i> max_iter ≥ 1 .		
7:	fail – NagError *	<i>Input/Output</i>
<i>The NAG error parameter (see the Essential Introduction).</i>		

5 Error Indicators and Warnings

NE_REAL_ARG_CONS

On entry, **df1** = <value>. This parameter must satisfy $0.0 < \text{df1} \leq 1.0e6$.

On entry, **lambda** = <value>. This parameter must satisfy $0.0 \leq \text{lambda} \leq -2.0 * \log(X02AMC)$.

NE_REAL_ARG_LE

On entry, **df2** must not be less than or equal to 0.0: **df2** = <value>.

On entry, **f** must not be less than or equal to 0.0: **f** = <value>.

NE_INT_ARG_LT

On entry, **max_iter** must not be less than 1: **max_iter** = <value>.

NE_CONV

The solution has failed to converge in <value> iterations, consider increasing **max_iter** or **tol**.

NE_PROB_F

The required probability cannot be computed accurately. This may happen if the result would be very close to zero or one. Alternatively the values of **df1** and **f** may be too large. In the latter case the user could try using a normal approximation, see Abramowitz and Stegun (1972).

NE_PROB_F_INIT

The required accuracy was not achieved when calculating the initial value of the central F or χ^2 probability. The user should try a larger value of **tol**. If the χ^2 approximation is being used then nag_prob_non_central_f_dist returns zero otherwise the value returned should be an approximation to the correct value.

NE_INTERNAL_ERROR

An internal error has occurred in this function. Check the function call and any array sizes. If the call is correct then please consult NAG for assistance.

6 Further Comments

When both ν_1 and ν_2 are large a normal approximation may be used and when only ν_1 is large a χ^2 approximation may be used. In both cases λ is required to be of the same order as ν_1 . See Abramowitz and Stegun Abramowitz and Stegun (1972) for further details.

6.1 Accuracy

The relative accuracy should be as specified by **tol**. For further details see nag_prob_non_central_chi_sq (g01gcc) and nag_prob_non_central_beta_dist (g01gec).

6.2 References

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

7 See Also

nag_prob_non_central_chi_sq (g01gcc)
nag_prob_non_central_beta_dist (g01gec)

8 Example

Values from, and degrees of freedom for F -distributions are read, the lower-tail probabilities computed, and all these values printed, until the end of data is reached.

8.1 Program Text

```
/* nag_prob_non_central_f_dist (g01gdc) Example Program.
*
* Copyright 2000 Numerical Algorithms Group.
*
* NAG C Library
*
* Mark 6, 2000.
*/
#include <stdio.h>
#include <nag.h>
#include <nagg01.h>

int main(void)
{
    double df1, df2, f, prob, lambda, tol;
    Integer max_iter;
    Integer exit_status=0;
    NagError fail;

    INIT_FAIL(fail);
    Vprintf("g01gdc Example Program Results\n");

    /* Skip heading in data file */
    Vscanf("%*[^\n]");

```

```

Vprintf ("\n      f      df1      df2      lambda      prob\n\n");
tol = 5e-6;
max_iter = 50;
while ((scanf("%lf %lf %lf %lf %*[^\n]", &f, &df1, &df2, &lambda)) != EOF)
{
    prob = g01gdc(f, df1, df2, lambda, tol, max_iter, &fail);
    if (fail.code != NE_NOERROR)
    {
        Vprintf("Error from g01gdc.\n%s\n", fail.message);
        exit_status=1;
        goto END;
    }
    Vprintf("%8.3f %8.3f %8.3f %8.3f %8.4f\n", f, df1, df2, lambda, prob);
}
END:
return exit_status;
}

```

8.2 Program Data

```

g01gdc Example Program Data
 5.5   1.5   25.5   3.0          :f df1 lambda
39.9   1.0   1.0    2.0          :f df1 lambda
 2.5  20.25  1.0    0.0          :f df1 lambda

```

8.3 Program Results

```

g01gdc Example Program Results

```

f	df1	df2	lambda	prob
5.500	1.500	25.500	3.000	0.8214
39.900	1.000	1.000	2.000	0.8160
2.500	20.250	1.000	0.000	0.5342
