NAG 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 noncentral 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 noncentral F-distribution with ν_1 and ν_2 degrees of freedom and noncentrality 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

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

$$\times u^{(\nu_1+2j-2)/2} [\nu_2 + (\nu_1 + 2j)u]^{-(\nu_1+2j+\nu_2)/2}$$

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

The probability is computed by means of a transformation to a noncentral beta distribution:

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

where $x = \dfrac{\nu_1 f}{\nu_1 f + \nu_2}$ and $P_\beta(X \leq x : a, b; \lambda)$ is the lower tail probability integral of the noncentral 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 References

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

5 Arguments

1: **f** – double *Input*

On entry: f, the deviate from the noncentral F-distribution.

Constraint: **f** > 0.0.

2: **df1** – double *Input*

On entry: the degrees of freedom of the numerator variance, ν_1.

Constraint: $0.0 < \mathbf{df1} \leq 10^6$.

3: **df2** – double *Input*

On entry: the degrees of freedom of the denominator variance, ν_2.

Constraint: **df2** > 0.0.

4: **lambda** – double *Input*

On entry: λ, the noncentrality parameter.

Constraint: $0.0 \leq \mathbf{lambda} \leq -2.0\log(U)$ where U is the safe range parameter as defined by nag_real_safe_small_number (X02AMC).

5: **tol** – double *Input*

On entry: the relative accuracy required by you in the results. If nag_prob_non_central_f_dist (g01gdc) is entered with **tol** greater than or equal to 1.0 or less than $10 \times$ ***machine precision*** (see nag_machine_precision (X02AJC)), then the value of $10 \times$ ***machine precision*** is used instead.

6: **max_iter** – Integer *Input*

On entry: the maximum number of iterations to be used.

Suggested value: 500. See nag_prob_non_central_chi_sq (g01gcc) and nag_prob_non_central_beta_dist (g01gec) for further details.

Constraint: **max_iter** ≥ 1.

7: **fail** – NagError * *Input/Output*

The NAG error argument (see Section 3.6 in the Essential Introduction).

6 Error Indicators and Warnings

NE_ALLOC_FAIL

Dynamic memory allocation failed.
See Section 3.2.1.2 in the Essential Introduction for further information.

NE_CONV

The solution has failed to converge in ⟨*value*⟩ iterations. Consider increasing **max_iter** or **tol**.

NE_INT_ARG_LT

On entry, **max_iter** $= \langle value \rangle$.
Constraint: **max_iter** ≥ 1.

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 contact NAG for assistance.

An unexpected error has been triggered by this function. Please contact NAG.
See Section 3.6.6 in the Essential Introduction for further information.

NE_NO_LICENCE

Your licence key may have expired or may not have been installed correctly.
See Section 3.6.5 in the Essential Introduction for further information.

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 you 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. You should try a larger value of **tol**. If the χ^2 approximation is being used then nag_prob_non_central_f_dist (g01gdc) returns zero otherwise the value returned should be an approximation to the correct value.

NE_REAL_ARG_CONS

On entry, **df1** $= \langle value \rangle$.
Constraint: $0.0 <$ **df1** $\leq 10^6$.

On entry, **df1** $= \langle value \rangle$.
Constraint: **df1** > 0.0.

On entry, **lambda** $= \langle value \rangle$.
Constraint: $0.0 \leq$ **lambda** $\leq -2.0 \times \log(U)$, where U is the safe range parameter as defined by nag_real_safe_small_number (X02AMC).

NE_REAL_ARG_LE

On entry, **df2** $= \langle value \rangle$.
Constraint: **df2** > 0.0.

On entry, **f** $= \langle value \rangle$.
Constraint: **f** > 0.0.

7 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).

8 Parallelism and Performance

Not applicable.

9 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 (1972) for further details.

10 Example

This example reads values from, and degrees of freedom for, F-distributions, computes the lower tail probabilities and prints all these values until the end of data is reached.

10.1 Program Text

```
/* nag_prob_non_central_f_dist (g01gdc) Example Program.
 *
 * Copyright 2014 Numerical Algorithms Group.
 *
 * NAG C Library
 *
 * Mark 6, 2000.
```

```
 */

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

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

  INIT_FAIL(fail);

  printf(
          "nag_prob_non_central_f_dist (g01gdc) Example Program Results\n");

  /* Skip heading in data file */
#ifdef _WIN32
  scanf_s("%*[^\n]");
#else
  scanf("%*[^\n]");
#endif

  printf("\n     f       df1     df2     lambda    prob\n\n");
  tol = 5e-6;
  max_iter = 50;
#ifdef _WIN32
  while ((scanf_s("%lf %lf %lf %lf %*[^\n]",
                  &f, &df1, &df2, &lambda)) != EOF)
    {
#else
  while ((scanf("%lf %lf %lf %lf %*[^\n]",
                  &f, &df1, &df2, &lambda)) != EOF)
    {
#endif
      /* nag_prob_non_central_f_dist (g01gdc).
       * Computes probabilities for the non-central F-distribution
       */
      prob = nag_prob_non_central_f_dist(f, df1, df2, lambda, tol, max_iter,
                                         &fail);
      if (fail.code != NE_NOERROR)
        {
          printf(
                  "Error from nag_prob_non_central_f_dist (g01gdc).\n%s\n",
                  fail.message);
          exit_status = 1;
          goto END;
        }
      printf("%8.3f %8.3f %8.3f %8.3f %8.4f\n", f, df1, df2, lambda,
              prob);
    }
 END:
  return exit_status;
}
```

10.2 Program Data

```
nag_prob_non_central_f_dist (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
```

10.3 Program Results

```
nag_prob_non_central_f_dist (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
```