NAG Library Function Document

nag_prob_non_central_chi_sq (g01gcc)

1 Purpose

nag_prob_non_central_chi_sq (g01gcc) returns the probability associated with the lower tail of the noncentral \( \chi^2 \)-distribution.

2 Specification

```c
#include <nag.h>
#include <nagg01.h>
double nag_prob_non_central_chi_sq (double x, double df, double lambda,
   double tol, Integer max_iter, NagError *fail)
```

3 Description

The lower tail probability of the noncentral \( \chi^2 \)-distribution with \( \nu \) degrees of freedom and noncentrality parameter \( \lambda \), \( P(X \leq x : \nu; \lambda) \), is defined by

\[
P(X \leq x : \nu; \lambda) = \sum_{j=0}^{\infty} e^{-\lambda/2} \left( \frac{\lambda}{2j} \right)^j P(X \leq x : \nu + 2j; 0),
\]

(1)

where \( P(X \leq x : \nu + 2j; 0) \) is a central \( \chi^2 \)-distribution with \( \nu + 2j \) degrees of freedom.

The value of \( j \) at which the Poisson weight, \( e^{-\lambda/2} \left( \frac{\lambda}{2j} \right)^j \), is greatest is determined and the summation (1) is made forward and backward from that value of \( j \).

The recursive relationship:

\[
P(X \leq x : a + 2; 0) = P(X \leq x : a; 0) - \frac{(x^a/2)e^{-x/2}}{\Gamma(a + 1)}
\]

(2)

is used during the summation in (1).

4 References


5 Arguments

1: x – double

*Input*

*On entry:* the deviate from the noncentral \( \chi^2 \)-distribution with \( \nu \) degrees of freedom and noncentrality parameter \( \lambda \).

*Constraint:* \( x \geq 0.0 \).

2: df – double

*Input*

*On entry:* \( \nu \), the degrees of freedom of the noncentral \( \chi^2 \)-distribution.

*Constraint:* \( df \geq 0.0 \).
3:  **lambda** – double  \hspace{1cm} Input  
   *On entry:* \( \lambda \), the noncentrality parameter of the noncentral \( \chi^2 \)-distribution.  
   *Constraint:* \( \lambda \geq 0.0 \) if \( df > 0.0 \) or \( \lambda > 0.0 \) if \( df = 0.0 \).

4:  **tol** – double  \hspace{1cm} Input  
   *On entry:* the required accuracy of the solution. If `nag_prob_non_central_chi_sq (g01gcc)` 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.

5:  **max_iter** – Integer  \hspace{1cm} Input  
   *On entry:* the maximum number of iterations to be performed.  
   *Suggested value:* 100. See Section 9 for further discussion.  
   *Constraint:* \( \text{max}_{\text{iter}} \geq 1 \).

6:  **fail** – NagError *  \hspace{1cm} Input/Output  
   The NAG error argument (see Section 3.6 in the Essential Introduction).

### 6 Error Indicators and Warnings

**NE_2_REAL_ARG_CONS**  
On entry, \( df = 0.0 \) and \( \lambda = 0.0 \).  
*Constraint:* \( \lambda > 0.0 \) if \( df = 0.0 \).

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

**NE_CHI_PROB**  
The calculations for the central chi-square probability has failed to converge. A larger value of \( tol \) should be used.

**NE_CONV**  
The solution has failed to converge in \( (\text{value}) \) iterations. Consider increasing \( \text{max}_{\text{iter}} \) or \( tol \).

**NE_INT_ARG_LT**  
On entry, \( \text{max}_{\text{iter}} = (\text{value}) \).  
*Constraint:* \( \text{max}_{\text{iter}} \geq 1 \).

**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.
The initial value of the Poisson weight used in the summation of (1) (see Section 3) was too small to be calculated. The computed probability is likely to be zero.

On entry, \( df = \text{value} \).
Constraint: \( df \geq 0.0 \).

On entry, \( \lambda = \text{value} \).
Constraint: \( \lambda \geq 0.0 \).

On entry, \( x = \text{value} \).
Constraint: \( x \geq 0.0 \).

The value of a term required in (2) (see Section 3) is too large to be evaluated accurately. The most likely cause of this error is both \( x \) and \( \lambda \) are too large.

Accuracy
The summations described in Section 3 are made until an upper bound on the truncation error relative to the current summation value is less than \( tol \).

Parallelism and Performance
Not applicable.

Further Comments
The number of terms in (1) required for a given accuracy will depend on the following factors:

(i) The rate at which the Poisson weights tend to zero. This will be slower for larger values of \( \lambda \).

(ii) The rate at which the central \( \chi^2 \) probabilities tend to zero. This will be slower for larger values of \( \nu \) and \( x \).

Example
This example reads values from various noncentral \( \chi^2 \)-distributions, calculates the lower tail probabilities and prints all these values until the end of data is reached.

Program Text
/* nag_prob_non_central_chi_sq (g01gcc) Example Program. */
* Copyright 2014 Numerical Algorithms Group.
* Mark 6a revised, 2001.
*/
#include <stdio.h>
#include <nag.h>
#include <nagg01.h>
int main(void)
{    Integer exit_status = 0, max_iter;
    NagError fail;
    double df, lambda, prob, tol, x;

    INIT_FAIL(fail);
printf("nag_prob_non_central_chi_sq (g01gcc) Example Program Results\n\n");

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

printf("n x df lambda prob\n\n");
tol = 5e-6;
max_iter = 50;
#ifdef _WIN32
    while ((scanf_s(" %lf %lf %lf %*[\n] ", &x, &df, &lambda)) != EOF)
#else
    while ((scanf(" %lf %lf %lf %*[\n] ", &x, &df, &lambda)) != EOF)
#endif
{
    /* nag_prob_non_central_chi_sq (g01gcc).
    * Computes probabilities for the non-central chi^2
    * distribution
    */
    prob = nag_prob_non_central_chi_sq(x, df, lambda, tol, max_iter, &fail);
    if (fail.code != NE_NOERROR)
    {
      printf("Error from nag_prob_non_central_chi_sq (g01gcc).\n%s\n", fail.message);
      exit_status = 1;
      goto END;
    }
    printf("%8.3f %8.3f %8.3f %8.4f\n", x, df, lambda, prob);
}
END:
return exit_status;
}

10.2 Program Data

nag_prob_non_central_chi_sq (g01gcc) Example Program Data
8.26 20.0 3.5 :x df lambda
6.2 7.5 2.0 :x df lambda
55.76 45.0 1.0 :x df lambda

10.3 Program Results

nag_prob_non_central_chi_sq (g01gcc) Example Program Results

<table>
<thead>
<tr>
<th>x</th>
<th>df</th>
<th>lambda</th>
<th>prob</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.260</td>
<td>20.000</td>
<td>3.500</td>
<td>0.0032</td>
</tr>
<tr>
<td>6.200</td>
<td>7.500</td>
<td>2.000</td>
<td>0.2699</td>
</tr>
<tr>
<td>55.760</td>
<td>45.000</td>
<td>1.000</td>
<td>0.8443</td>
</tr>
</tbody>
</table>