NAG Library Function Document

nag_gamma_dist (g01efc)

1 Purpose

nag_gamma_dist (g01efc) returns the lower or upper tail probability of the gamma distribution, with parameters α and β.

2 Specification

```
#include <nag.h>
#include <nagg01.h>
double nag_gamma_dist (Nag_TailProbability tail, double g, double a,
       double b, NagError *fail)
```

3 Description

The lower tail probability for the gamma distribution with parameters α and β, $P(G \leq g)$, is defined by:

$$P(G \leq g;\alpha,\beta) = \frac{1}{\beta^{\alpha}\Gamma(\alpha)}\int_0^g G^{\alpha-1}e^{-G/\beta}\,dG, \quad \alpha > 0.0,\ \beta > 0.0.$$

The mean of the distribution is $\alpha\beta$ and its variance is $\alpha\beta^2$. The transformation $Z = \dfrac{G}{\beta}$ is applied to yield the following incomplete gamma function in normalized form,

$$P(G \leq g;\alpha,\beta) = P(Z \leq g/\beta : \alpha, 1.0) = \frac{1}{\Gamma(\alpha)}\int_0^{g/\beta} Z^{\alpha-1}e^{-Z}\,dZ.$$

This is then evaluated using nag_incomplete_gamma (s14bac).

4 References

Hastings N A J and Peacock J B (1975) *Statistical Distributions* Butterworth

5 Arguments

1: **tail** – Nag_TailProbability *Input*

On entry: indicates whether an upper or lower tail probability is required.

tail = Nag_LowerTail
The lower tail probability is returned, that is $P(G \leq g : \alpha, \beta)$.

tail = Nag_UpperTail
The upper tail probability is returned, that is $P(G \geq g : \alpha, \beta)$.

Constraint: **tail** = Nag_LowerTail or Nag_UpperTail.

2: **g** – double *Input*

On entry: g, the value of the gamma variate.

Constraint: **g** ≥ 0.0.

3: **a** – double *Input*

On entry: the parameter α of the gamma distribution.

Constraint: **a** > 0.0.

4: **b** – double *Input*

On entry: the parameter β of the gamma distribution.

Constraint: **b** > 0.0.

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

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

6 Error Indicators and Warnings

On any of the error conditions listed below except **fail**.**code** = NE_ALG_NOT_CONV nag_gamma_dist (g01efc) returns 0.0.

NE_ALG_NOT_CONV

The algorithm has failed to converge in ⟨*value*⟩ iterations. The probability returned should be a reasonable approximation to the solution.

NE_BAD_PARAM

On entry, argument ⟨*value*⟩ had an illegal 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 contact NAG for assistance.

NE_REAL_ARG_LE

On entry, **a** = ⟨*value*⟩ and **b** = ⟨*value*⟩.
Constraint: **a** > 0.0 and **b** > 0.0.

NE_REAL_ARG_LT

On entry, **g** = ⟨*value*⟩.
Constraint: **g** ≥ 0.0.

7 Accuracy

The result should have a relative accuracy of ***machine precision***. There are rare occasions when the relative accuracy attained is somewhat less than ***machine precision*** but the error should not exceed more than 1 or 2 decimal places. Note also that there is a limit of 18 decimal places on the achievable accuracy, because constants in nag_incomplete_gamma (s14bac) are given to this precision.

8 Parallelism and Performance

Not applicable.

9 Further Comments

The time taken by nag_gamma_dist (g01efc) varies slightly with the input arguments **g**, **a** and **b**.

10 Example

This example reads in values from a number of gamma distributions and computes the associated lower tail probabilities.

10.1 Program Text

```
/* nag_gamma_dist (g01efc) Example Program.
 *
 * Copyright 1990 Numerical Algorithms Group.
 *
 * Mark 1, 1990.
 */

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

int main(void)
{
  Integer  exit_status = 0;
  double   a, b, g, p;
  NagError fail;

  INIT_FAIL(fail);

  /* Skip heading in data file */
  scanf("%*[^\n]");
  printf("nag_gamma_dist (g01efc) Example Program Results\n");
  printf(" Gamma deviate    Alpha     Beta    Lower tail prob.\n\n");
  while (scanf("%lf %lf %lf", &g, &a, &b) != EOF)
    {
      /* nag_gamma_dist (g01efc).
       * Probabilities for the gamma distribution
       */
      p = nag_gamma_dist(Nag_LowerTail, g, a, b, &fail);
      if (fail.code != NE_NOERROR)
        {
          printf("Error from nag_gamma_dist (g01efc).\n%s\n",
                  fail.message);
          exit_status = 1;
          goto END;
        }
      printf(" %9.2f%13.2f%9.2f%14.4f\n", g, a, b, p);
    }

 END:
  return exit_status;
}
```

10.2 Program Data

```
nag_gamma_dist (g01efc) Example Program Data
 15.5   4.0   2.0
  0.5   4.0   1.0
 10.0   1.0   2.0
  5.0   2.0   2.0
```

10.3 Program Results

```
nag_gamma_dist (g01efc) Example Program Results
 Gamma deviate    Alpha     Beta    Lower tail prob.

     15.50         4.00     2.00        0.9499
      0.50         4.00     1.00        0.0018
     10.00         1.00     2.00        0.9933
```

5.00	2.00	2.00	0.7127