NAG Library Function Document

nag_incomplete_gamma (s14bac)

1 Purpose

nag_incomplete_gamma (s14bac) computes values for the incomplete gamma functions $P(a,x)$ and $Q(a,x)$.

2 Specification

```
#include <nag.h>
#include <nags.h>
void nag_incomplete_gamma (double a, double x, double tol, double *p,
     double *q, NagError *fail)
```

3 Description

nag_incomplete_gamma (s14bac) evaluates the incomplete gamma functions in the normalized form

$$P(a,x) = \frac{1}{\Gamma(a)}\int_0^x t^{a-1}e^{-t}\,dt,$$

$$Q(a,x) = \frac{1}{\Gamma(a)}\int_x^\infty t^{a-1}e^{-t}\,dt,$$

with $x \geq 0$ and $a > 0$, to a user-specified accuracy. With this normalization, $P(a,x) + Q(a,x) = 1$.

Several methods are used to evaluate the functions depending on the arguments a and x, the methods including Taylor expansion for $P(a,x)$, Legendre's continued fraction for $Q(a,x)$, and power series for $Q(a,x)$. When both a and x are large, and $a \simeq x$, the uniform asymptotic expansion of Temme (1987) is employed for greater efficiency – specifically, this expansion is used when $a \geq 20$ and $0.7a \leq x \leq 1.4a$.

Once either P or Q is computed, the other is obtained by subtraction from 1. In order to avoid loss of relative precision in this subtraction, the smaller of P and Q is computed first.

This function is derived from the function GAM in Gautschi (1979b).

4 References

Gautschi W (1979a) A computational procedure for incomplete gamma functions *ACM Trans. Math. Software* **5** 466–481

Gautschi W (1979b) Algorithm 542: Incomplete gamma functions *ACM Trans. Math. Software* **5** 482–489

Temme N M (1987) On the computation of the incomplete gamma functions for large values of the parameters *Algorithms for Approximation* (eds J C Mason and M G Cox) Oxford University Press

5 Arguments

1: **a** – double *Input*

On entry: the argument a of the functions.

Constraint: **a** > 0.0.

2: **x** – double *Input*

On entry: the argument x of the functions.

Constraint: **x** ≥ 0.0.

3: **tol** – double *Input*

On entry: the relative accuracy required by you in the results. If nag_incomplete_gamma (s14bac) is entered with **tol** greater than 1.0 or less than ***machine precision***, then the value of ***machine precision*** is used instead.

4: **p** – double * *Output*

5: **q** – double * *Output*

On exit: the values of the functions $P(a,x)$ and $Q(a,x)$ respectively.

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

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

6 Error Indicators and Warnings

NE_ALG_NOT_CONV

Algorithm fails to terminate in ⟨*value*⟩ iterations.

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*⟩.
Constraint: **a** > 0.0.

NE_REAL_ARG_LT

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

7 Accuracy

There are rare occasions when the relative accuracy attained is somewhat less than that specified by argument **tol**. However, the error should never exceed more than one or two decimal places. Note also that there is a limit of 18 decimal places on the achievable accuracy, because constants in the function are given to this precision.

8 Parallelism and Performance

Not applicable.

9 Further Comments

The time taken for a call of nag_incomplete_gamma (s14bac) depends on the precision requested through **tol**, and also varies slightly with the input arguments a and x.

10 Example

This example reads values of the argument a and x from a file, evaluates the function and prints the results.

10.1 Program Text

```
/* nag_incomplete_gamma (s14bac) Example Program.
 *
 * Copyright 1990 Numerical Algorithms Group.
 *
 * Mark 2 revised, 1992.
 */

#include <nag.h>
#include <stdio.h>
#include <nag_stdlib.h>
#include <nags.h>
#include <nagx02.h>

int main(void)
{
  Integer  exit_status = 0;
  double   a, p, q, tol, x;
  NagError fail;

  INIT_FAIL(fail);

  /* Skip heading in data file */
  scanf("%*[^\n]");
  printf("nag_incomplete_gamma (s14bac) Example Program Results\n");
  /* nag_machine_precision (x02ajc).
   * The machine precision
   */
  tol = nag_machine_precision;
  printf("        a           x           p           q\n");
  while (scanf("%lf %lf", &a, &x) != EOF)
    {
      /* nag_incomplete_gamma (s14bac).
       * Incomplete Gamma functions P(ax) and Q(ax)
       */
      nag_incomplete_gamma(a, x, tol, &p, &q, &fail);
      if (fail.code != NE_NOERROR)
        {
          printf("Error from nag_incomplete_gamma (s14bac).\n%s\n",
                  fail.message);
          exit_status = 1;
          goto END;
        }
      printf("%12.4f%12.4f%12.4f%12.4f\n", a, x, p, q);
    }

 END:
  return exit_status;
}
```

10.2 Program Data

```
nag_incomplete_gamma (s14bac) Example Program Data
  2.0   3.0
  7.0   1.0
  0.5  99.0
 20.0  21.0
 21.0  20.0
```

10.3 Program Results

```
nag_incomplete_gamma (s14bac) Example Program Results
        a           x           p           q
      2.0000      3.0000      0.8009      0.1991
      7.0000      1.0000      0.0001      0.9999
      0.5000     99.0000      1.0000      0.0000
     20.0000     21.0000      0.6157      0.3843
     21.0000     20.0000      0.4409      0.5591
```