NAG Library Routine Document

S14BAF

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of ***bold italicised*** terms and other implementation-dependent details.

1 Purpose

S14BAF computes values for the incomplete gamma functions $P(a,x)$ and $Q(a,x)$.

2 Specification

```
SUBROUTINE S14BAF (A, X, TOL, P, Q, IFAIL)

INTEGER             IFAIL
REAL (KIND=nag_wp)  A, X, TOL, P, Q
```

3 Description

S14BAF 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 routine is derived from the subroutine 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 Parameters

1: A – REAL (KIND=nag_wp) *Input*

On entry: the argument a of the functions.

Constraint: A > 0.0.

2: X – REAL (KIND=nag_wp) *Input*

On entry: the argument x of the functions.

Constraint: $\mathrm{X} \geq 0.0$.

3: TOL – REAL (KIND=nag_wp) *Input*

On entry: the relative accuracy required by you in the results. If S14BAF is entered with TOL greater than 1.0 or less than ***machine precision***, then the value of ***machine precision*** is used instead.

4: P – REAL (KIND=nag_wp) *Output*
5: Q – REAL (KIND=nag_wp) *Output*

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

6: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this parameter you should refer to Section 3.3 in the Essential Introduction for details.

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, if you are not familiar with this parameter, the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

On exit: $\mathrm{IFAIL} = 0$ unless the routine detects an error or a warning has been flagged (see Section 6).

6 Error Indicators and Warnings

If on entry $\mathrm{IFAIL} = 0$ or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

$\mathrm{IFAIL} = 1$

On entry, $\mathrm{A} \leq 0.0$.

$\mathrm{IFAIL} = 2$

On entry, $\mathrm{X} < 0.0$.

$\mathrm{IFAIL} = 3$

Convergence of the Taylor series or Legendre continued fraction fails within 600 iterations. This error is extremely unlikely to occur; if it does, contact NAG.

7 Accuracy

There are rare occasions when the relative accuracy attained is somewhat less than that specified by parameter 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 routine are given to this precision.

8 Further Comments

The time taken for a call of S14BAF depends on the precision requested through TOL, and also varies slightly with the input arguments a and x.

9 Example

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

9.1 Program Text

```
    Program s14bafe

!     S14BAF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: nag_wp, s14baf, x02ajf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: a, p, q, tol, x
      Integer                          :: ifail, ioerr
!     .. Executable Statements ..
      Write (nout,*) 'S14BAF Example Program Results'

!     Skip heading in data file
      Read (nin,*)

      Write (nout,*)
      Write (nout,*) '         A           X           P           Q'

      tol = x02ajf()

data: Do
        Read (nin,*,Iostat=ioerr) a, x

        If (ioerr<0) Then
          Exit data
        End If

        ifail = 0
        Call s14baf(a,x,tol,p,q,ifail)

        Write (nout,99999) a, x, p, q
      End Do data

99999 Format (1X,4F12.4)
    End Program s14bafe
```

9.2 Program Data

```
S14BAF Example Program Data
  2.0   3.0
  7.0   1.0
  0.5  99.0
 20.0  21.0
 21.0  20.0
```

9.3 Program Results

```
 S14BAF 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
```