S14ABF – NAG Fortran Library Routine Document

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

S14ABF returns a value for the logarithm of the Gamma function, $\ln\Gamma(x)$, via the routine name.

2 Specification

```
real FUNCTION S14ABF(X, IFAIL)
INTEGER              IFAIL
real                 X
```

3 Description

This routine evaluates an approximation to $\ln\Gamma(x)$. It is based on two Chebyshev expansions.

For $0 < x \leq x_{small}$, $\ln\Gamma(x) = -\ln x$ to within machine accuracy.

For $x_{small} < x \leq 15.0$, the recursive relation $\Gamma(1+x) = x\Gamma(x)$ is used to reduce the calculation to one involving $\Gamma(1+u)$, $0 \leq u < 1$ which is evaluated as:

$$\Gamma(1+u) = {\sum_{r=0}}' a_r T_r(t), \quad t = 2u - 1.$$

Once $\Gamma(x)$ has been calculated, the required result is produced by taking the logarithm.

For $15.0 < x \leq x_{big}$,

$$\ln\Gamma(x) = (x - \frac{1}{2})\ln x - x + \frac{1}{2}\ln 2\pi + y(x)/x$$

where $y(x) = {\sum_{r=0}}' b_r T_r(t)$, $t = 2\left(\frac{15}{x}\right)^2 - 1$.

For $x_{big} < x \leq x_{vbig}$ the term $y(x)/x$ is negligible and so its calculation is omitted.

For $x > x_{vbig}$ there is a danger of setting overflow so the routine must fail.

For $x \leq 0.0$ the function is not defined and the routine fails.

Note. x_{small} is calculated so that if $x < x_{small}$, $\Gamma(x) = 1/x$ to within machine accuracy. x_{big} is calculated so that if $x > x_{big}$,

$$\ln\Gamma(x) = (x - \frac{1}{2})\ln x - x + \frac{1}{2}\ln 2\pi$$

to within machine accuracy. x_{vbig} is calculated so that $\ln\Gamma(x_{vbig})$ is close to the value returned by X02ALF.

4 References

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

5 Parameters

1: X — ***real*** *Input*

On entry: the argument x of the function.

Constraint: X > 0.0.

2: IFAIL — INTEGER *Input/Output*

On entry: IFAIL must be set to 0, −1 or 1. For users not familiar with this parameter (described in Chapter P01) the recommended value is 0.

On exit: IFAIL = 0 unless the routine detects an error (see Section 6).

6 Error Indicators and Warnings

Errors detected by the routine:

IFAIL = 1

X ≤ 0.0, the function is undefined. On soft failure, the routine returns zero.

IFAIL = 2

X is too large, the function would overflow. On soft failure, the routine returns the value of the function at the largest permissible argument.

7 Accuracy

Let δ and ϵ be the relative errors in the argument and result respectively, and E be the absolute error in the result.

If δ is somewhat larger than the relative ***machine precision***, then

$$E \simeq |x \times \Psi(x)|\delta \text{ and } \epsilon \simeq \left|\frac{x \times \Psi(x)}{\ln \Gamma(x)}\right|\delta$$

where $\Psi(x)$ is the digamma function $\frac{\Gamma'(x)}{\Gamma(x)}$. Figure 1 and Figure 2 show the behaviour of these error amplification factors.

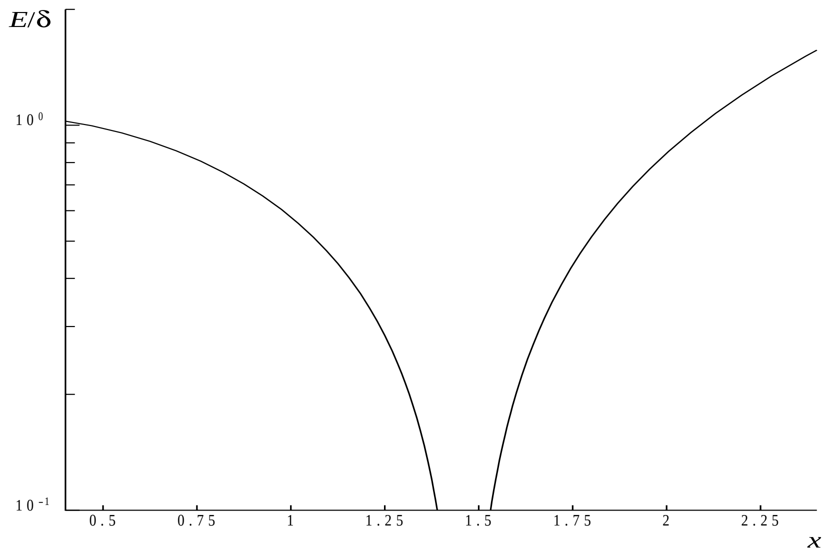

Figure 1

These show that relative error can be controlled, since except near $x = 1$ or 2 relative error is attenuated by the function or at least is not greatly amplified.

For large x, $\epsilon \simeq \left(1 + \frac{1}{\ln x}\right)\delta$ and for small x, $\epsilon \simeq \frac{1}{\ln x}\delta$.

The function $\ln \Gamma(x)$ has zeros at $x = 1$ and 2 and hence relative accuracy is not maintainable near those points. However absolute accuracy can still be provided near those zeros as is shown above.

If however, δ is of the order of the ***machine precision***, then rounding errors in the routine's internal arithmetic may result in errors which are slightly larger than those predicted by the equalities. It should be noted that even in areas where strong attenuation of errors is predicted the relative precision is bounded by the effective ***machine precision***.

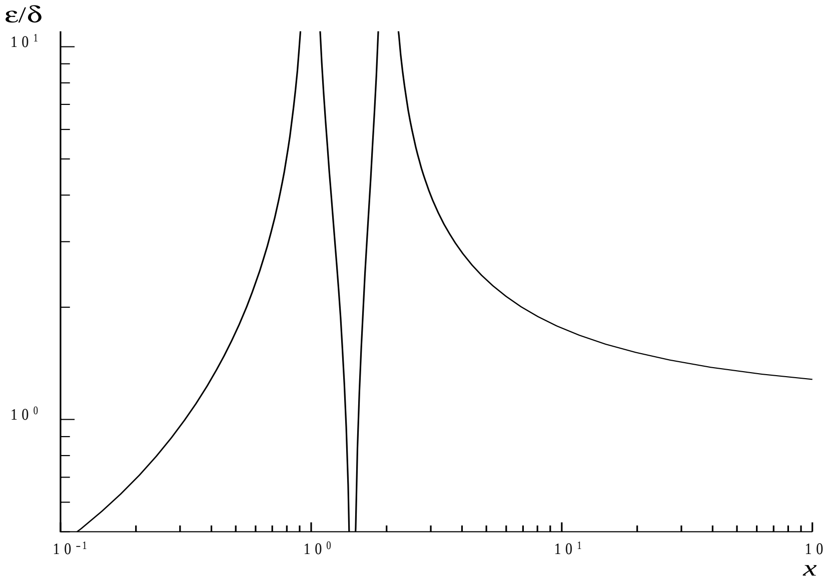

Figure 2

8 Further Comments

None.

9 Example

The example program reads values of the argument x from a file, evaluates the function at each value of x and prints the results.

9.1 Program Text

Note. The listing of the example program presented below uses bold italicised terms to denote precision-dependent details. Please read the Users' Note for your implementation to check the interpretation of these terms. As explained in the Essential Introduction to this manual, the results produced may not be identical for all implementations.

```
*     S14ABF Example Program Text
*     Mark 14 Revised.  NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
*     .. Local Scalars ..
      real             X, Y
      INTEGER          IFAIL
*     .. External Functions ..
      real             S14ABF
      EXTERNAL         S14ABF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'S14ABF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      WRITE (NOUT,*)
      WRITE (NOUT,*) '     X           Y        IFAIL'
      WRITE (NOUT,*)
   20 READ (NIN,*,END=40) X
      IFAIL = 1
*
      Y = S14ABF(X,IFAIL)
*
```

```
      WRITE (NOUT,99999) X, Y, IFAIL
      GO TO 20
   40 STOP
*
99999 FORMAT (1X,1P,2e12.3,I7)
      END
```

9.2 Program Data

```
S14ABF Example Program Data
             1.0
             1.25
             1.5
             1.75
             2.0
             5.0
             10.0
             20.0
             1000.0
             0.0
             -5.0
```

9.3 Program Results

```
 S14ABF Example Program Results

      X           Y        IFAIL

   1.000E+00   0.000E+00      0
   1.250E+00  -9.827E-02      0
   1.500E+00  -1.208E-01      0
   1.750E+00  -8.440E-02      0
   2.000E+00   0.000E+00      0
   5.000E+00   3.178E+00      0
   1.000E+01   1.280E+01      0
   2.000E+01   3.934E+01      0
   1.000E+03   5.905E+03      0
   0.000E+00   0.000E+00      1
  -5.000E+00   0.000E+00      1
```