

S18AFF – 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

S18AFF returns a value for the modified Bessel Function $I_1(x)$, via the routine name.

2 Specification

```
real FUNCTION S18AFF(X, IFAIL)
INTEGER          IFAIL
real           X
```

3 Description

This routine evaluates an approximation to the modified Bessel Function of the first kind $I_1(x)$.

Note. $I_1(-x) = -I_1(x)$, so the approximation need only consider $x \geq 0$.

The routine is based on three Chebyshev expansions:

For $0 < x \leq 4$,

$$I_1(x) = x \sum_{r=0}' a_r T_r(t), \text{ where } t = 2 \left(\frac{x}{4} \right)^2 - 1;$$

For $4 < x \leq 12$,

$$I_1(x) = e^x \sum_{r=0}' b_r T_r(t), \text{ where } t = \frac{x-8}{4};$$

For $x > 12$,

$$I_1(x) = \frac{e^x}{\sqrt{x}} \sum_{r=0}' c_r T_r(t), \text{ where } t = 2 \left(\frac{12}{x} \right) - 1.$$

For small x , $I_1(x) \simeq x$. This approximation is used when x is sufficiently small for the result to be correct to *machine precision*.

For large x , the routine must fail because $I_1(x)$ cannot be represented without overflow.

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.
- 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 is too large. On soft failure the routine returns the approximate value of $I_1(x)$ at the nearest valid argument.

7 Accuracy

Let δ and ϵ be the relative errors in the argument and result respectively.

If δ is somewhat larger than the *machine precision* (i.e., if δ is due to data errors etc.), then ϵ and δ are approximately related by:

$$\epsilon \simeq \left| \frac{xI_0(x) - I_1(x)}{I_1(x)} \right| \delta.$$

Figure 1 shows the behaviour of the error amplification factor

$$\left| \frac{xI_0(x) - I_1(x)}{I_1(x)} \right|,$$

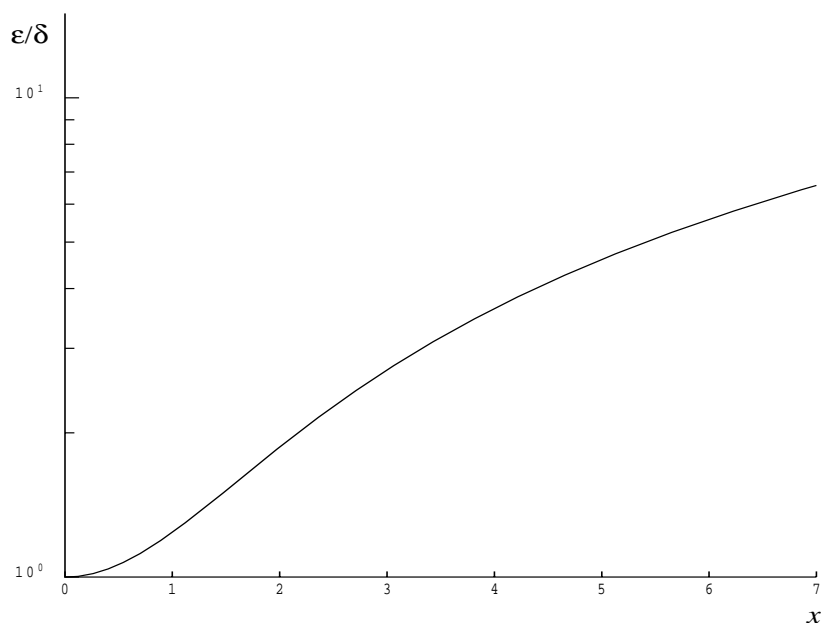


Figure 1

However, if δ is of the same order as *machine precision*, then rounding errors could make ϵ slightly larger than the above relation predicts.

For small x , $\epsilon \simeq \delta$ and there is no amplification of errors.

For large x , $\epsilon \simeq x\delta$ and we have strong amplification of errors. However the routine must fail for quite moderate values of x because $I_1(x)$ would overflow; hence in practice the loss of accuracy for large x is not excessive. Note that for large x , the errors will be dominated by those of the Fortran intrinsic function EXP.

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.

```

*      S18AFF 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             S18AFF
      EXTERNAL         S18AFF
*      .. Executable Statements ..
      WRITE (NOUT,*) 'S18AFF 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 = S18AFF(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

```

S18AFF Example Program Data
      0.0
      0.5
      1.0
      3.0
      6.0
      8.0
      10.0
      15.0
      20.0
      -1.0

```

9.3 Program Results

S18AFF Example Program Results

X	Y	IFAIL
0.000E+00	0.000E+00	0
5.000E-01	2.579E-01	0
1.000E+00	5.652E-01	0
3.000E+00	3.953E+00	0
6.000E+00	6.134E+01	0
8.000E+00	3.999E+02	0
1.000E+01	2.671E+03	0
1.500E+01	3.281E+05	0
2.000E+01	4.245E+07	0
-1.000E+00	-5.652E-01	0
