NAG Library Routine Document

S14ADF

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

S14ADF returns a sequence of values of scaled derivatives of the psi function $\psi(x)$ (also known as the digamma function).

2 Specification

```
SUBROUTINE S14ADF (X, N, M, ANS, IFAIL)
INTEGER            N, M, IFAIL
REAL (KIND=nag_wp) X, ANS(M)
```

3 Description

S14ADF computes m values of the function

$$w(k,x) = \frac{(-1)^{k+1}\psi^{(k)}(x)}{k!},$$

for $x > 0$, $k = n,\ n+1, \ldots, n+m-1$, where ψ is the psi function

$$\psi(x) = \frac{d}{dx}\ln\Gamma(x) = \frac{\Gamma'(x)}{\Gamma(x)},$$

and $\psi^{(k)}$ denotes the kth derivative of ψ.

The routine is derived from the routine PSIFN in Amos (1983). The basic method of evaluation of $w(k,x)$ is the asymptotic series

$$w(k,x) \sim \epsilon(k,x) + \frac{1}{2x^{k+1}} + \frac{1}{x^k}\sum_{j=1}^{\infty} B_{2j}\frac{(2j+k-1)!}{(2j)!k!x^{2j}}$$

for large x greater than a machine-dependent value $x_{\min}$, followed by backward recurrence using

$$w(k,x) = w(k,x+1) + x^{-k-1}$$

for smaller values of x, where $\epsilon(k,x) = -\ln x$ when $k = 0$, $\epsilon(k,x) = \dfrac{1}{kx^k}$ when $k > 0$, and B_{2j}, $j = 1, 2, \ldots$, are the Bernoulli numbers.

When k is large, the above procedure may be inefficient, and the expansion

$$w(k,x) = \sum_{j=1}^{\infty}\frac{1}{(x+j)^{k+1}},$$

which converges rapidly for large k, is used instead.

4 References

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

Amos D E (1983) Algorithm 610: A portable FORTRAN subroutine for derivatives of the psi function *ACM Trans. Math. Software* **9** 494–502

5 Parameters

1: X – REAL (KIND=nag_wp) *Input*

On entry: the argument x of the function.

Constraint: X > 0.0.

2: N – INTEGER *Input*

On entry: the index of the first member n of the sequence of functions.

Constraint: N ≥ 0.

3: M – INTEGER *Input*

On entry: the number of members m required in the sequence $w(k,x)$, for $k = n, \ldots, n+m-1$.

Constraint: M ≥ 1.

4: ANS(M) – REAL (KIND=nag_wp) array *Output*

On exit: the first m elements of ANS contain the required values $w(k,x)$, for $k = n, \ldots, n+m-1$.

5: 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: 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 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:

IFAIL $= 1$

On entry, X ≤ 0.0.

IFAIL $= 2$

On entry, N < 0.

IFAIL $= 3$

On entry, M < 1.

IFAIL $= 4$

No results are returned because underflow is likely. Either X or N + M − 1 is too large. If possible, reduce the value of M and call S14ADF again.

IFAIL $= 5$

No results are returned because overflow is likely. Either X is too small, or $\mathrm{N} + \mathrm{M} - 1$ is too large. If possible, reduce the value of M and call S14ADF again.

IFAIL $= 6$

No results are returned because there is not enough internal workspace to continue computation. $\mathrm{N} + \mathrm{M} - 1$ may be too large. If possible, reduce the value of M and call S14ADF again.

IFAIL $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

See Section 3.8 in the Essential Introduction for further information.

IFAIL $= -399$

Your licence key may have expired or may not have been installed correctly.

See Section 3.7 in the Essential Introduction for further information.

IFAIL $= -999$

Dynamic memory allocation failed.

See Section 3.6 in the Essential Introduction for further information.

7 Accuracy

All constants in S14ADF are given to approximately 18 digits of precision. Calling the number of digits of precision in the floating-point arithmetic being used t, then clearly the maximum number of correct digits in the results obtained is limited by $p = \min(t, 18)$. Empirical tests of S14ADF, taking values of x in the range $0.0 < x < 50.0$, and n in the range $1 \leq n \leq 50$, have shown that the maximum relative error is a loss of approximately two decimal places of precision. Tests with $n = 0$, i.e., testing the function $-\psi(x)$, have shown somewhat better accuracy, except at points close to the zero of $\psi(x)$, $x \simeq 1.461632$, where only absolute accuracy can be obtained.

8 Parallelism and Performance

Not applicable.

9 Further Comments

The time taken for a call of S14ADF is approximately proportional to m, plus a constant. In general, it is much cheaper to call S14ADF with m greater than 1 to evaluate the function $w(k, x)$, for $k = n, \ldots, n + m - 1$, rather than to make m separate calls of S14ADF.

10 Example

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

10.1 Program Text

```
    Program s14adfe

!     S14ADF Example Program Text

!     Mark 25 Release. NAG Copyright 2014.

!     .. Use Statements ..
      Use nag_library, Only: nag_wp, s14adf
```

```
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: m = 4, nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: x
      Integer                          :: i, ifail, ioerr, n
!     .. Local Arrays ..
      Real (Kind=nag_wp)               :: ans(m)
!     .. Executable Statements ..
      Write (nout,*) 'S14ADF Example Program Results'

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

      Write (nout,*)
      Write (nout,*) '      X           ANS(1)        ANS(2)        ', &
        'ANS(3)        ANS(4)'
      Write (nout,*)

      n = 0

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

        If (ioerr<0) Then
          Exit data
        End If

        ifail = 0
        Call s14adf(x,n,m,ans,ifail)

        Write (nout,99999) x, (ans(i),i=1,m)
      End Do data

99999 Format (1X,1P,5(E12.4,2X))
    End Program s14adfe
```

10.2 Program Data

```
S14ADF Example Program Data
0.1
0.5
3.6
8.0
```

10.3 Program Results

```
 S14ADF Example Program Results

       X           ANS(1)        ANS(2)        ANS(3)        ANS(4)

   1.0000E-01    1.0424E+01    1.0143E+02    1.0009E+03    1.0001E+04
   5.0000E-01    1.9635E+00    4.9348E+00    8.4144E+00    1.6235E+01
   3.6000E+00   -1.1357E+00    3.1988E-01    5.0750E-02    1.0653E-02
   8.0000E+00   -2.0156E+00    1.3314E-01    8.8498E-03    7.8321E-04
```