

NAG C Library Function Document

nag_real_polygamma (s14aec)

1 Purpose

nag_real_polygamma (s14aec) returns the value of the k th derivative of the psi function $\psi(x)$ for real x and $k = 0, 1, \dots, 6$.

2 Specification

```
double nag_real_polygamma (double x, Integer k, NagError *fail)
```

3 Description

This routine evaluates an approximation to the k th derivative of the psi function $\psi(x)$ given by

$$\psi^{(k)}(x) = \frac{d^k}{dx^k} \psi(x) = \frac{d^k}{dx^k} \left(\frac{d}{dx} \log_e \Gamma(x) \right),$$

where x is real with $x \neq 0, -1, -2, \dots$ and $k = 0, 1, \dots, 6$. For negative non-integer values of x , the recurrence relationship

$$\psi^{(k)}(x+1) = \psi^{(k)}(x) + \frac{d^k}{dx^k} \left(\frac{1}{x} \right)$$

is used. The value of $\frac{(-1)^{k+1} \psi^{(k)}(x)}{k!}$ is obtained by a call to a routine based on PSIFN in Amos (1983).

Note that $\psi^{(k)}(x)$ is also known as the *polygamma* function. Specifically, $\psi^{(0)}(x)$ is often referred to as the *digamma* function and $\psi^{(1)}(x)$ as the *trigamma* function in the literature. Further details can be found in Abramowitz and Stegun (1972).

4 Parameters

- | | | |
|----|--|---------------------|
| 1: | x – double | <i>Input</i> |
| | <i>On entry:</i> the argument x of the function. | |
| | <i>Constraint:</i> x must not be ‘too close’ (see Section 5) to a non-positive integer. | |
| 2: | k – Integer | <i>Input</i> |
| | <i>On entry:</i> the function $\psi^{(k)}(z)$ to be evaluated. | |
| | <i>Constraint:</i> $0 \leq \mathbf{k} \leq 6$. | |
| 3: | fail – NagError * | <i>Input/Output</i> |
| | The NAG error parameter (see the Essential Introduction). | |

5 Error Indicators and Warnings

NE_INT

On entry, **k** = <value>.

Constraint: $0 \leq \mathbf{k} \leq 6$.

NE_REAL

On entry, $x = \langle \text{value} \rangle$.

Constraint: x must not be ‘too close’ to a non-positive integer. That is, $|x - \text{nint}(x)| \geq \text{machine precision} \times \text{nint}(x)$

NE_UNDERFLOW_LIKELY

The evaluation has been abandoned due to the likelihood of underflow. The result is returned as zero.

NE_OVERFLOW_LIKELY

The evaluation has been abandoned due to the likelihood of overflow. The result is returned as zero.

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 consult NAG for assistance.

6 Further Comments

6.1 Accuracy

All constants in the underlying functions are given to approximately 18 digits of precision. If t denotes the number of digits of precision in the floating-point arithmetic being used, then clearly the maximum number in the results obtained is limited by $p = \min(t, 18)$. Empirical tests by Amos (1983) have shown that the maximum relative error is a loss of approximately two decimal places of precision. Further tests with the function $-\psi^{(0)}(x)$ have shown somewhat improved accuracy, except at points near the positive zero of $\psi^{(0)}(x)$ at $x = 1.46\dots$, where only absolute accuracy can be obtained.

6.2 References

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

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

7 See Also

None.

8 Example

The example program evaluates $\psi^{(2)}(x)$ at $x = 2.5$, and prints the results.

8.1 Program Text

```
/* nag_real_polygamma (s14aec) Example Program.
 *
 * Copyright 2000 Numerical Algorithms Group.
 *
 * NAG C Library
 *
 * Mark 6, 2000.
 */

#include <stdio.h>
#include <nag.h>
```

```

#include <nag_stdlib.h>
#include <nags.h>

int main(void)
{
    double x, y;
    Integer exit_status=0;
    NagError fail;
    Integer k;

    INIT_FAIL(fail);

    Vprintf("s14aec Example Program Results\n\n");
    /* Skip heading in data file */
    Vscanf("%*[^\\n]");
    Vprintf("\\n      X      K      (D^K/DX^K)psi(X)\\n\\n\\n");

    while(scanf("%lf %ld%*[^\\n]", &x, &k) != EOF)
    {
        y = s14aec (x, k, &fail);
        if (fail.code == NE_NOERROR)
            Vprintf("%5.1f %5ld      %12.4e\\n", x, k, y);
        else
        {
            Vprintf("Error from s14aec.\\n%s\\n", fail.message);
            exit_status = 1;
            goto END;
        }
    }
    END:
    return exit_status;
}

```

8.2 Program Data

```

s14aec Example Program Data
1.0  0
0.5  1
-3.6 2
8.0  3
2.9  4
-4.7 5
-5.4 6 : Values of x and k

```

8.3 Program Results

```
s14aec Example Program Results
```

X	K	(D^K/DX^K)psi(X)
1.0	0	-5.7722e-01
0.5	1	4.9348e+00
-3.6	2	-2.2335e+01
8.0	3	4.6992e-03
2.9	4	-1.5897e-01
-4.7	5	1.6566e+05
-5.4	6	4.1378e+05