Purpose

s14af returns the value of the $k$th derivative of the psi function $\psi(z)$ for complex $z$ and $k = 0, 1, \ldots, 4$, via the function name.

Syntax

\[
\text{[result, ifail]} = \text{s14af}(z, k)
\]

Description

s14af evaluates an approximation to the $k$th derivative of the psi function $\psi(z)$ given by

\[
\psi^{(k)}(z) = \frac{d^k}{dz^k} \psi(z) = \frac{d^k}{dz^k} \left( \frac{d}{dz} \log \Gamma(z) \right),
\]

where $z = x + iy$ is complex provided $y \neq 0$ and $k = 0, 1, \ldots, 4$. If $y = 0$, $z$ is real and thus $\psi^{(k)}(z)$ is singular when $z = 0, -1, -2, \ldots$.

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

s14af is based on a modification of the method proposed by Kölbig (1972).

To obtain the value of $\psi^{(k)}(z)$ when $z$ is real, s14ae can be used.

References


Köllbig K S (1972) Programs for computing the logarithm of the gamma function, and the digamma function, for complex arguments *Comp. Phys. Comm.* 4 221–226

Parameters

5.1 Compulsory Input Parameters

1: 

$z$ – complex scalar

The argument $z$ of the function.

*Constraint*: $\text{Re}(z)$ must not be ‘too close’ (see Section 6) to a nonpositive integer when $\text{Im}(z) = 0.0$.

2: 

$k$ – int32 scalar

The function $\psi^{(k)}(z)$ to be evaluated.

*Constraint*: $0 \leq k \leq 4$.

5.2 Optional Input Parameters

None.
5.3 Input Parameters Omitted from the MATLAB Interface

None.

5.4 Output Parameters

1: result – complex scalar
   The result of the function.

2: ifail – int32 scalar
   ifail = 0 unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail = 1
   On entry, k < 0,
   or k > 4,
   or Re(z) is ‘too close’ to a nonpositive integer when Im(z) = 0.0. That is,
   abs(Re(z) - nint(Re(z))) < machine precision * nint(abs(Re(z))).

ifail = 2
   The evaluation has been abandoned due to the likelihood of overflow. The result is returned as zero.

7 Accuracy

Empirical tests have shown that the maximum relative error is a loss of approximately two decimal places of precision.

8 Further Comments

None.

9 Example

```matlab
z = complex(-1.5, +2.5);
k = int32(1);
[result, ifail] = s14af(z, k)
```

result =
-0.1974 - 0.2427i
ifail =
0