NAG Library Function Document

nag_complex_log_gamma (s14agc)

1 Purpose

nag_complex_log_gamma (s14agc) returns the value of the logarithm of the gamma function $\ln \Gamma(z)$ for complex z, .

2 Specification

```
#include <nag.h>
#include <nags.h>
Complex nag_complex_log_gamma (Complex z, NagError *fail)
```

3 Description

nag_complex_log_gamma (s14agc) evaluates an approximation to the logarithm of the gamma function $\ln \Gamma(z)$ defined for $\mathrm{Re}(z) > 0$ by

$$\ln \Gamma(z) = \ln \int_0^\infty e^{-t} t^{z-1} dt$$

where $z = x + iy$ is complex. It is extended to the rest of the complex plane by analytic continuation unless $y = 0$, in which case z is real and each of the points $z = 0, -1, -2, \ldots$ is a singularity and a branch point.

nag_complex_log_gamma (s14agc) is based on the method proposed by Kölbig (1972) in which the value of $\ln \Gamma(z)$ is computed in the different regions of the z plane by means of the formulae

$$\begin{aligned} \ln \Gamma(z) &= \left(z - \tfrac{1}{2}\right) \ln z - z + \tfrac{1}{2} \ln 2\pi + z \sum_{k=1}^{K} \frac{B_{2k}}{2k(2k-1)} z^{-2k} + R_K(z) && \text{if } x \geq x_0 \geq 0, \\ &= \ln \Gamma(z+n) - \ln \prod_{\nu=0}^{n-1} (z+\nu) && \text{if } x_0 > x \geq 0, \\ &= \ln \pi - \ln \Gamma(1-z) - \ln(\sin \pi z) && \text{if } x < 0, \end{aligned}$$

where $n = [x_0] - [x]$, $\{B_{2k}\}$ are Bernoulli numbers (see Abramowitz and Stegun (1972)) and $[x]$ is the largest integer $\leq x$. Note that care is taken to ensure that the imaginary part is computed correctly, and not merely modulo 2π.

The function uses the values $K = 10$ and $x_0 = 7$. The remainder term $R_K(z)$ is discussed in Section 7.

To obtain the value of $\ln \Gamma(z)$ when z is real and positive, nag_log_gamma (s14abc) can be used.

4 References

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

Kölbig 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

5 Arguments

1: **z** – Complex *Input*

On entry: the argument z of the function.

Constraint: $\mathbf{z}.re$ must not be 'too close' (see Section 6) to a non-positive integer when $\mathbf{z}.im = 0.0$.

2: **fail** – NagError * *Input/Output*

The NAG error argument (see Section 3.6 in the Essential Introduction).

6 Error Indicators and Warnings

NE_ALLOC_FAIL

Dynamic memory allocation failed.
See Section 3.2.1.2 in the Essential Introduction for further information.

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 contact NAG for assistance.

An unexpected error has been triggered by this function. Please contact NAG.
See Section 3.6.6 in the Essential Introduction for further information.

NE_NO_LICENCE

Your licence key may have expired or may not have been installed correctly.
See Section 3.6.5 in the Essential Introduction for further information.

NE_TOO_CLOSE_INTEGER

On entry, $\mathbf{z}.re$ is 'too close' to a non-positive integer when $\mathbf{z}.im = 0.0$: $\mathbf{z}.re = \langle value \rangle$, $\mathrm{nint}(\mathbf{z}.re) = \langle value \rangle$.

7 Accuracy

The remainder term $R_K(z)$ satisfies the following error bound:

$$\begin{aligned} |R_K(z)| &\leq \frac{|B_{2K}|}{|(2K-1)|} z^{1-2K} \\ &\leq \frac{|B_{2K}|}{|(2K-1)|} x^{1-2K} \text{if } x \geq 0. \end{aligned}$$

Thus $|R_{10}(7)| < 2.5 \times 10^{-15}$ and hence in theory the function is capable of achieving an accuracy of approximately 15 significant digits.

8 Parallelism and Performance

Not applicable.

9 Further Comments

None.

10 Example

This example evaluates the logarithm of the gamma function $\ln \Gamma(z)$ at $z = -1.5 + 2.5i$, and prints the results.

10.1 Program Text

```
/* nag_complex_log_gamma (s14agc) Example Program.
 *
 * Copyright 2014 Numerical Algorithms Group.
 *
 * Mark 7, 2002.
 */

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

int main(void)
{
  Integer  exit_status = 0;
  Complex  y, z;
  NagError fail;

  INIT_FAIL(fail);

  /* Skip heading in data file */
#ifdef _WIN32
  scanf_s("%*[^\n]");
#else
  scanf("%*[^\n]");
#endif
  printf("nag_complex_log_gamma (s14agc) Example Program Results\n");
  printf("     z                  ln(Gamma(z))\n");
#ifdef _WIN32
  while (scanf_s(" (%lf,%lf)%*[^\n] ", &z.re, &z.im) != EOF)
#else
  while (scanf(" (%lf,%lf)%*[^\n] ", &z.re, &z.im) != EOF)
#endif
    {
      /* nag_complex_log_gamma (s14agc).
       * Logarithm of the Gamma function ln Gamma(z)
       */
      y = nag_complex_log_gamma(z, &fail);
      if (fail.code == NE_NOERROR)
        printf("(%5.1f,%5.1f)   (%13.4e,%13.4e)\n", z.re, z.im, y.re,
                y.im);
      else
        {
          printf("Error from nag_complex_log_gamma (s14agc).\n%s\n",
                  fail.message);
          exit_status = 1;
          goto END;
        }
    }

 END:

  return exit_status;
}
```

10.2 Program Data

```
nag_complex_log_gamma (s14agc) Example Program Data
 (-1.5, 2.5)  : Value of z
```

10.3 Program Results

```
nag_complex_log_gamma (s14agc) Example Program Results
     z                  ln(Gamma(z))
( -1.5,  2.5)   ( -5.0140e+00, -4.0718e+00)
```