

NAG Library Function Document

nag_log_beta (s14cbc)

1 Purpose

nag_log_beta (s14cbc) returns the value of the logarithm of the beta function, $\ln B(a, b)$, via the routine name.

2 Specification

```
#include <nag.h>
#include <nags.h>
double nag_log_beta (double a, double b, NagError *fail)
```

3 Description

nag_log_beta (s14cbc) calculates values for $\ln B(a, b)$ where B is the beta function given by

$$B(a, b) = \int_0^1 t^{a-1} (1-t)^{b-1} dt$$

or equivalently

$$B(a, b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$$

and $\Gamma(x)$ is the gamma function. Note that the beta function is symmetric, so that $B(a, b) = B(b, a)$.

In order to efficiently obtain accurate results several methods are used depending on the parameters a and b .

Let $a_0 = \min(a, b)$ and $b_0 = \max(a, b)$. Then:

for $a_0 \geq 8$,

$$\ln B = 0.5 \ln(2\pi) - 0.5 \ln(b_0) + \Delta(a_0) + \Delta(b_0) - \Delta(a_0 + b_0) - u - v;$$

where

$$\Delta(a_0) = \ln \Gamma(a_0) - (a_0 - 0.5) \ln a_0 + a_0 - 0.5 \ln(2\pi),$$

$$u = -(a_0 - 0.5) \ln \left[\frac{a_0}{a_0 + b_0} \right] \quad \text{and}$$

$$v = b_0 \ln \left(1 + \frac{a_0}{b_0} \right).$$

for $a_0 < 1$,

for $b_0 \geq 8$,

$$\ln B = \ln \Gamma(a_0) + \ln \frac{\Gamma(b_0)}{\Gamma(a_0 + b_0)};$$

for $b_0 < 8$,

$$\ln B = \ln \Gamma(a_0) + \ln \Gamma(b_0) - \ln \Gamma(a_0 + b_0);$$

for $2 < a_0 < 8$, a_0 is reduced to the interval $[1, 2]$ by $B(a, b) = \frac{a_0-1}{a_0+b_0-1} B(a_0-1, b_0)$;

for $1 \leq a_0 \leq 2$,
 for $b_0 \geq 8$,

$$\ln B = \ln \Gamma(a_0) + \ln \frac{\Gamma(b_0)}{\Gamma(a_0 + b_0)};$$

for $2 < b_0 < 8$, b_0 is reduced to the interval $[1, 2]$;
 for $b_0 \leq 2$,

$$\ln B = \ln \Gamma(a_0) + \ln \Gamma(b_0) - \ln \Gamma(a_0 + b_0).$$

nag_log_beta (s14cbc) is derived from BETALN in DiDonato and Morris (1992).

4 References

DiDonato A R and Morris A H (1992) Algorithm 708: Significant digit computation of the incomplete beta function ratios *ACM Trans. Math. Software* **18** 360–373

5 Arguments

- 1: **a** – double *Input*
On entry: the parameter a of the function.
Constraint: **a** > 0.0.
- 2: **b** – double *Input*
On entry: the parameter b of the function.
Constraint: **b** > 0.0.
- 3: **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_REAL

On entry, **a** = $\langle value \rangle$.
 Constraint: **a** > 0.0.

On entry, **b** = $\langle value \rangle$.
 Constraint: **b** > 0.0.

7 Accuracy

nag_log_beta (s14cbc) should produce full relative accuracy for all input arguments.

8 Parallelism and Performance

Not applicable.

9 Further Comments

None.

10 Example

This example reads values of the parameters a and b from a file, evaluates the function and prints the results.

10.1 Program Text

```

/* nag_log_beta (s14cbc) Example Program.
 *
 * Copyright 2014 Numerical Algorithms Group.
 */

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

int main(void)
{
    Integer    exit_status = 0;
    double    a, b, lb;
    NagError  fail;

    INIT_FAIL(fail);

    /* Skip heading in data file */
#ifdef _WIN32
    scanf_s("%*[\n]");
#else
    scanf("%*[\n]");
#endif
    printf("nag_log_beta (s14cbc) Example Program Results\n");
    printf("  a      b      ln(beta(a,b))\n");
#ifdef _WIN32
    while (scanf_s("%lf %lf", &a, &b) != EOF)
#else
    while (scanf("%lf %lf", &a, &b) != EOF)
#endif
    {
        /* nag_log_beta (s14cbc).
         * Log Beta function ln(beta(a,b))
         */
        lb = nag_log_beta(a, b, &fail);
        if (fail.code != NE_NOERROR)
        {
            printf("Error from nag_log_beta (s14cbc).\n%s\n",
                fail.message);
            exit_status = 1;
            goto END;
        }
        printf("%5.2f %5.2f %12.4e\n", a, b, lb);
    }
}

```

```
    }  
  
    END:  
    return exit_status;  
}
```

10.2 Program Data

```
nag_log_beta (s14cbc) Example Program Data  
0.2  1.0  
0.6  1.0  
1.0  0.2  
1.0  1.0  
2.0  2.0  
5.0  5.0  
6.0  2.0  
6.0  3.0
```

10.3 Program Results

```
nag_log_beta (s14cbc) Example Program Results  
a      b      ln(beta(a,b))  
0.20  1.00    1.6094e+00  
0.60  1.00    5.1083e-01  
1.00  0.20    1.6094e+00  
1.00  1.00    0.0000e+00  
2.00  2.00   -1.7918e+00  
5.00  5.00   -6.4457e+00  
6.00  2.00   -3.7377e+00  
6.00  3.00   -5.1240e+00
```
