

## NAG C Library Function Document

### nag\_rngs\_weibull (g05lmc)

#### 1 Purpose

nag\_rngs\_weibull (g05lmc) generates a vector of pseudo-random numbers from a two parameter Weibull distribution with shape parameter  $a$  and scale parameter  $b$ .

#### 2 Specification

```
void nag_rngs_weibull (double a, double b, Integer n, double x[], Integer igen,
    Integer iseed[], NagError *fail)
```

#### 3 Description

The distribution has PDF (probability density function)

$$f(x) = \frac{a}{b} x^{a-1} e^{-x^a/b} \quad \text{if } x > 0,$$

$$f(x) = 0 \quad \text{otherwise.}$$

nag\_rngs\_weibull (g05lmc) returns the value  $(-b \ln y)^{1/a}$ , where  $y$  is a pseudo-random number from a uniform distribution over (0,1).

One of the initialisation functions nag\_rngs\_init\_repeatable (g05kbc) (for a repeatable sequence if computed sequentially) or nag\_rngs\_init\_nonrepeatable (g05kcc) (for a non-repeatable sequence) must be called prior to the first call to nag\_rngs\_weibull (g05lmc).

#### 4 References

Knuth D E (1981) *The Art of Computer Programming (Volume 2)* (2nd Edition) Addison–Wesley

Kendall M G and Stuart A (1969) *The Advanced Theory of Statistics (Volume 1)* (3rd Edition) Griffin

#### 5 Parameters

- |    |  |               |
|----|--|---------------|
| 1: | <b>a</b> – double  | <i>Input</i>  |
|    | <i>On entry:</i> the shape parameter, $a$ , of the distribution.                                   |               |
|    | <i>Constraint:</i> <b>a</b> > 0.0.   |               |
| 2: | <b>b</b> – double  | <i>Input</i>  |
|    | <i>On entry:</i> the scale parameter, $b$ , of the distribution.                                   |               |
|    | <i>Constraint:</i> <b>b</b> > 0.0.   |               |
| 3: | <b>n</b> – Integer   | <i>Input</i>  |
|    | <i>On entry:</i> the number, $n$ , of pseudo-random numbers to be generated.                       |               |
|    | <i>Constraint:</i> <b>n</b> ≥ 0.   |               |
| 4: | <b>x</b> [ <i>dim</i> ] – double   | <i>Output</i> |
|    | <b>Note:</b> the dimension, <i>dim</i> , of the array <b>x</b> must be at least max(1, <b>n</b> ). |               |
|    | <i>On exit:</i> the $n$ pseudo-random numbers from the specified Weibull distribution.             |               |

- 5: **igen** – Integer *Input*  
*On entry:* must contain the identification number for the generator to be used to return a pseudo-random number and should remain unchanged following initialisation by a prior call to one of the functions `nag_rngs_init_repeatable` (g05kbc) or `nag_rngs_init_nonrepeatable` (g05kcc).
- 6: **iseed**[4] – Integer *Input/Output*  
*On entry:* contains values which define the current state of the selected generator.  
*On exit:* contains updated values defining the new state of the selected generator.
- 7: **fail** – NagError \* *Input/Output*  
The NAG error parameter (see the Essential Introduction).

## 6 Error Indicators and Warnings

### NE\_INT

On entry, **n** =  $\langle value \rangle$ .  
Constraint: **n**  $\geq$  0.

### NE\_REAL

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

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

### NE\_BAD\_PARAM

On entry, parameter  $\langle value \rangle$  had an illegal value.

### 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.

## 7 Accuracy

Not applicable.

## 8 Further Comments

None.

## 9 Example

The example program prints the first five pseudo-random real numbers from a Weibull distribution with shape parameter 1.0 and scale parameter 2.0, generated by a single call to `nag_rngs_weibull` (g05lmc), after initialisation by `nag_rngs_init_repeatable` (g05kbc).

### 9.1 Program Text

```
/* nag_rngs_weibull(g05lmc) Example Program.
 *
 * Copyright 2001 Numerical Algorithms Group.
 *
 * Mark 7, 2001.
 */
```

```

#include <stdio.h>
#include <nag.h>
#include <nag_stdlib.h>
#include <nagg05.h>

int main(void)
{
    /* Scalars */
    Integer  igen, j, m;
    Integer  exit_status=0;
    NagError fail;

    /* Arrays */
    double  *x=0;
    Integer  iseed[4];

    INIT_FAIL(fail);
    Vprintf("g05lmc Example Program Results\n\n");

    m = 5;
    /* Allocate memory */
    if ( !(x = NAG_ALLOC(m, double)) )
    {
        Vprintf("Allocation failure\n");
        exit_status = -1;
        goto END;
    }

    /* Initialise the seed to a repeatable sequence */
    iseed[0] = 1762543;
    iseed[1] = 9324783;
    iseed[2] = 42344;
    iseed[3] = 742355;
    /* igen identifies the stream. */
    igen = 1;
    g05kbc(&igen, iseed);

    g05lmc(1.0, 2.0, m, x, igen, iseed, &fail);
    if (fail.code != NE_NOERROR)
    {
        Vprintf("Error from g05lmc.\n%s\n", fail.message);
        exit_status = 1;
        goto END;
    }
    for (j = 0; j < m; ++j)
    {
        Vprintf("%10.4f\n", x[j]);
    }
    END:
    if (x) NAG_FREE(x);
    return exit_status;
}

```

## 9.2 Program Data

None.

## 9.3 Program Results

g05lmc Example Program Results

```

4.8310
0.1006
1.8009
0.5936
0.1031

```