NAG C Library Function Document

nag_rngs_cauchy (g05llc)

1 Purpose

nag_rngs_cauchy (g05llc) generates a vector of pseudo-random numbers from a Cauchy distribution with median \( a \) and semi-interquartile range \( b \).

2 Specification

```c
void nag_rngs_cauchy (double xmed, double semiqr, Integer n, double x[],
                   Integer igen, Integer iseed[], NagError *fail)
```

3 Description

The distribution has PDF (probability density function)

\[
f(x) = \frac{1}{\pi b \left(1 + \left(\frac{x-a}{b}\right)^2\right)}.
\]

nag_rngs_cauchy (g05llc) returns the value

\[
a + b \frac{2y_1 - 1}{y_2},
\]

where \( y_1 \) and \( y_2 \) are a pair of consecutive pseudo-random numbers from a uniform distribution over (0,1), such that

\[
(2y_1 - 1)^2 + y_2^2 \leq 1.
\]

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_cauchy (g05llc).

4 References


5 Parameters

1: \( \text{xmed} \) – double
   
   \text{Input}
   
   \text{On entry:} the median, \( a \), of the distribution.

2: \( \text{semiqr} \) – double
   
   \text{Input}
   
   \text{On entry:} the semi-interquartile range, \( b \), of the distribution.
   
   \text{Constraint:} \( \text{semiqr} \geq 0.0 \).

3: \( \text{n} \) – Integer
   
   \text{Input}
   
   \text{On entry:} the number, \( n \), of pseudo-random numbers to be generated.
   
   \text{Constraint:} \( \text{n} \geq 0 \).

4: \( \text{x}[\text{dim}] \) – double
   
   \text{Output}
   
   \text{Note:} the dimension, \( \text{dim} \), of the array \( \text{x} \) must be at least \( \max(1, \text{n}) \).
On exit: the \( n \) pseudo-random numbers from the specified Cauchy 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).


*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, \( n = (\text{value}) \).

Constraint: \( n \geq 0 \).

**NE_REAL**

On entry, \( \text{semiqr} = (\text{value}) \).

Constraint: \( \text{semiqr} \geq 0.0 \).

**NE_BAD_PARAM**

On entry, parameter \( (\text{value}) \) 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 Cauchy distribution with median 1.0 and semi-interquartile range 2.0, generated by a single call to nag_rngs_cauchy (g05llc), after initialisation by nag_rngs_init_repeatable (g05kbc).

9.1 Program Text

```c
/* nag_rngs_cauchy(g05llc) Example Program. */
/* Copyright 2001 Numerical Algorithms Group. */
/* * Mark 7, 2001. */
#include <stdio.h>
```
```c
#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("g05llc 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);
    g05llc(1.0, 2.0, m, x, igen, iseed, &fail);
    if (fail.code != NE_NOERROR)
      {
        Vprintf("Error from g05llc.\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

<table>
<thead>
<tr>
<th>g05llc Example Program Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4962</td>
</tr>
<tr>
<td>1.8604</td>
</tr>
<tr>
<td>0.2698</td>
</tr>
<tr>
<td>1.0859</td>
</tr>
<tr>
<td>-3.9829</td>
</tr>
</tbody>
</table>
```

[NP3645/7]