NAG C Library Function Document

nag_rngs_chi_sq (g05lcc)

1 Purpose

nag_rngs_chi_sq (g05lcc) generates a vector of pseudo-random numbers taken from a $\chi^2$ distribution with $\nu$ degrees of freedom.

2 Specification

```c
void nag_rngs_chi_sq (Integer df, Integer n, double x[], Integer igen,
                   Integer iseed[], NagError *fail)
```

3 Description

The distribution has PDF (probability density function)

\[
f(x) = \frac{x^{\nu-1} \times e^{-x/2}}{2^\nu \times (\nu/2 - 1)!} \quad \text{if } x > 0;
\]

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

This is the same as a gamma distribution with parameters $\frac{1}{2} \nu$ and 2.

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_chi_sq (g05lcc).

4 References


5 Parameters

1: \( df \) – Integer \( \text{Input} \)

On entry: the number of degrees of freedom, $\nu$, of the distribution.

Constraint: $df \geq 1$.

2: \( n \) – Integer \( \text{Input} \)

On entry: the number, $n$, of pseudo-random numbers to be generated.

Constraint: $n \geq 0$.

3: \( x[dim] \) – double \( \text{Output} \)

Note: the dimension, $dim$, of the array $x$ must be at least max(1, $n$).

On exit: the $n$ pseudo-random numbers from the specified $\chi^2$ distribution.

4: \( igen \) – Integer \( \text{Input} \)

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).
5:  **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.

6:  **fail** – NagError *

    *Input/Output*

    The NAG error parameter (see the Essential Introduction).

### 6 Error Indicators and Warnings

**NE_INT**

- On entry, **n** = *(value)*.
  
  Constraint: **n** ≥ 0.

- On entry, **df** = *(value)*.
  
  Constraint: **df** ≥ 1.

**NE_BAD_PARAM**

- On entry, parameter *(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

The time taken by **nag_rngs_chi_sq (g05lcc)** increases with **ν**.

### 9 Example

The example program prints 5 pseudo-random numbers from a \( \chi^2 \) distribution with five degrees of freedom, generated by a single call to **nag_rngs_chi_sq (g05lcc)**, after initialisation by **nag_rngs_init_repeatable (g05kbc)**.

#### 9.1 Program Text

```c
/* nag_rngs_chi_sq(g05lcc) 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 i, igen, n;
    Integer exit_status=0;
    NagError fail;
```
/* Arrays */
double  *x=0;
Integer  iseed[4];

INIT_FAIL(fail);
Vprintf("g05lcc Example Program Results\n\n");
n = 5;

/* Allocate memory */
if ( !(x = NAG_ALLOC(n, 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);
g05lcc(5, n, x, igen, iseed, &fail);
if (fail.code != NE_NOERROR)
  {
    Vprintf("Error from g05lcc.\n%s\n", fail.message);
    exit_status = 1;
    goto END;
  }
for (i = 0; i < n; ++i)
  {
    Vprintf("%10.4f\n", x[i]);
  }

END:
if (x) NAG_FREE(x);
return exit_status;

9.2 Program Data
None.

9.3 Program Results

\n\begin{verbatim}
g05lcc Example Program Results

  2.0097
  13.6994
  3.8964
  3.0941
  8.0579
\end{verbatim}