NAG C Library Function Document

nag_rngs_f (g05ldc)

1 Purpose

nag_rngs_f (g05ldc) generates a vector of pseudo-random numbers taken from a F (or Fisher's variance ratio) distribution with μ and ν degrees of freedom.

2 Specification

```
void nag_rngs_f (Integer df1, Integer df2, Integer n, double x[], Integer igen,
     Integer iseed[], NagError *fail)
```

3 Description

The distribution has PDF (probability density function)

$$f(x) = \frac{\left(\frac{\mu+\nu-2}{2}\right)!\, x^{\frac{1}{2}\mu-1}}{\left(\frac{1}{2}\mu-1\right)!\left(\frac{1}{2}\nu-1\right)!\left(1+\frac{\mu}{\nu}x\right)^{\frac{1}{2}(\mu+\nu)}} \times \left(\frac{\mu}{\nu}\right)^{\frac{1}{2}\mu} \quad \text{if } x > 0,$$

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

nag_rngs_f (g05ldc) calculates the values

$$\frac{\nu y_i}{\mu z_i}, \quad i = 1, \ldots, n,$$

where y_i and z_i are generated by nag_rngs_gamma (g05lfc) from gamma distributions with parameters $(\frac{1}{2}\mu, 2)$ and $(\frac{1}{2}\nu, 2)$ respectively (i.e., from χ^2 distributions with μ and ν degrees of freedom).

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_f (g05ldc).

4 References

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

5 Parameters

1: **df1** – Integer *Input*

On entry: the number of degrees of freedom, μ, of the distribution.

Constraint: **df1** ≥ 1.

2: **df2** – Integer *Input*

On entry: the number of degrees of freedom, ν, of the distribution.

Constraint: **df2** ≥ 1.

3: **n** – Integer *Input*

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

Constraint: **n** ≥ 0.

4: **x**[dim] – double *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 F 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** = ⟨*value*⟩.
Constraint: **n** ≥ 0.

On entry, **df2** = ⟨*value*⟩.
Constraint: **df2** ≥ 1.

On entry, **df1** = ⟨*value*⟩.
Constraint: **df1** ≥ 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_f (g05ldc) increases with μ and ν.

9 Example

The example program prints five pseudo-random numbers from a F-distribution with two and three degrees of freedom, generated by a single call to nag_rngs_f (g05ldc), after initialisation by nag_rngs_init_repeatable (g05kbc).

9.1 Program Text

```
/* nag_rngs_f(g05ldc) 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("g05ldc 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);
  g05ldc(2, 3, n, x, igen, iseed, &fail);
  if (fail.code != NE_NOERROR)
    {
      Vprintf("Error from g05ldc.\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

```
g05ldc Example Program Results

   14.2359
    0.8889
    0.4055
    2.3299
    0.0689
```