

NAG Toolbox

nag_rand_permute (g05nc)

1 Purpose

nag_rand_permute (g05nc) performs a pseudorandom permutation of a vector of integers.

2 Syntax

```
[indx, state, ifail] = nag_rand_permute(indx, state, 'n', n)
[indx, state, ifail] = g05nc(indx, state, 'n', n)
```

3 Description

nag_rand_permute (g05nc) permutes the elements of an integer array without inspecting their values. Each of the $n!$ possible permutations of the n values may be regarded as being equally probable.

Even for modest values of n it is theoretically impossible that all $n!$ permutations may occur, as $n!$ is likely to exceed the cycle length of any of the base generators. For practical purposes this is irrelevant, as the time necessary to generate all possible permutations is many millenia.

One of the initialization functions nag_rand_init_repeat (g05kf) (for a repeatable sequence if computed sequentially) or nag_rand_init_nonrepeat (g05kg) (for a non-repeatable sequence) must be called prior to the first call to nag_rand_permute (g05nc).

4 References

Kendall M G and Stuart A (1969) *The Advanced Theory of Statistics (Volume 1)* (3rd Edition) Griffin
 Knuth D E (1981) *The Art of Computer Programming (Volume 2)* (2nd Edition) Addison–Wesley

5 Parameters

5.1 Compulsory Input Parameters

1: **indx(n)** – INTEGER array

The n integer values to be permuted.

2: **state(:)** – INTEGER array

Note: the actual argument supplied **must** be the array **state** supplied to the initialization routines nag_rand_init_repeat (g05kf) or nag_rand_init_nonrepeat (g05kg).

Contains information on the selected base generator and its current state.

5.2 Optional Input Parameters

1: **n** – INTEGER

Default: the dimension of the array **indx**.

The number of values to be permuted.

Constraint: $n \geq 1$.

5.3 Output Parameters

- 1: **indx(n)** – INTEGER array
The n permuted integer values.
- 2: **state(:)** – INTEGER array
Contains updated information on the state of the generator.
- 3: **ifail** – INTEGER
ifail = 0 unless the function detects an error (see Section 5).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail = 2

Constraint: $n \geq 1$.

ifail = 3

On entry, **state** vector has been corrupted or not initialized.

ifail = -99

An unexpected error has been triggered by this routine. Please contact NAG.

ifail = -399

Your licence key may have expired or may not have been installed correctly.

ifail = -999

Dynamic memory allocation failed.

7 Accuracy

Not applicable.

8 Further Comments

None.

9 Example

In the example program a vector containing the first eight positive integers in ascending order is permuted by a call to `nag_rand_permute (g05nc)` and the permutation is printed. This is repeated a total of ten times, after initialization by `nag_rand_init_repeat (g05kf)`.

9.1 Program Text

```
function g05nc_example

fprintf('g05nc example results\n\n');

% Initialize the seed and identify the base generator
seed = [nag_int(1762543)];
genid = nag_int(1);
subid = nag_int(1);
```

```

% Initialize the generator to a repeatable sequence
[state, ifail] = g05kf( ...
                    genid, subid, seed);

index = zeros(10, 8, nag_int_name);

% Set up the index vector
index1 = nag_int([1:8]);

fprintf(' 10 Permutations of first 8 integers\n');
% Permutate 10 times
for j = 1:10
    % Call the permutation routine
    [index(j,:), state, ifail] = g05nc( ...
                                    index1, state);
end

% Display variates
disp(double(index));

```

9.2 Program Results

g05nc example results

```

10 Permutations of first 8 integers
 6      2      4      8      1      3      5      7
 8      6      4      2      7      3      1      5
 4      2      8      7      5      6      3      1
 1      6      4      5      2      3      7      8
 1      7      3      8      4      2      5      6
 6      3      4      7      1      2      8      5
 6      4      1      8      2      5      3      7
 3      2      1      7      5      8      6      4
 4      2      1      5      3      6      8      7
 1      5      6      4      2      7      8      3

```
