NAG Library Routine Document

G05NCF

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of ***bold italicised*** terms and other implementation-dependent details.

1 Purpose

G05NCF performs a pseudorandom permutation of a vector of integers.

2 Specification

```
SUBROUTINE G05NCF (INDX, N, STATE, IFAIL)
INTEGER INDX(N), N, STATE(*), IFAIL
```

3 Description

G05NCF 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 routines G05KFF (for a repeatable sequence if computed sequentially) or G05KGF (for a non-repeatable sequence) must be called prior to the first call to G05NCF.

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 Arguments

1: INDX(N) – INTEGER array *Input/Output*

On entry: the n integer values to be permuted.

On exit: the n permuted integer values.

2: N – INTEGER *Input*

On entry: the number of values to be permuted.

Constraint: N ≥ 1.

3: STATE($*$) – INTEGER array *Communication Array*

Note: the actual argument supplied **must** be the array STATE supplied to the initialization routines G05KFF or G05KGF.

On entry: contains information on the selected base generator and its current state.

On exit: contains updated information on the state of the generator.

4: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this argument you should refer to Section 3.4 in How to Use the NAG Library and its Documentation for details.

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, if you are not familiar with this argument, the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

On exit: IFAIL $= 0$ unless the routine detects an error or a warning has been flagged (see Section 6).

6 Error Indicators and Warnings

If on entry IFAIL $= 0$ or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

IFAIL $= 2$

On entry, N $= \langle value \rangle$.
Constraint: N ≥ 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.

See Section 3.9 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -399$

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

See Section 3.8 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -999$

Dynamic memory allocation failed.

See Section 3.7 in How to Use the NAG Library and its Documentation for further information.

7 Accuracy

Not applicable.

8 Parallelism and Performance

G05NCF is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this routine. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

9 Further Comments

None.

10 Example

In the example program a vector containing the first eight positive integers in ascending order is permuted by a call to G05NCF and the permutation is printed. This is repeated a total of ten times, after initialization by G05KFF.

10.1 Program Text

```
    Program g05ncfe

!     G05NCF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: g05kff, g05ncf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: lseed = 1, nin = 5, nout = 6
!     .. Local Scalars ..
      Integer                          :: genid, i, ifail, j, lstate, n,       &
                                          nperm, subid
!     .. Local Arrays ..
      Integer, Allocatable             :: indx(:), state(:)
      Integer                          :: seed(lseed)
!     .. Executable Statements ..
      Write (nout,*) 'G05NCF Example Program Results'
      Write (nout,*)

!     Skip heading in data file
      Read (nin,*)

!     Read in the base generator information and seed
      Read (nin,*) genid, subid, seed(1)

!     Initial call to initializer to get size of STATE array
      lstate = 0
      Allocate (state(lstate))
      ifail = 0
      Call g05kff(genid,subid,seed,lseed,state,lstate,ifail)

!     Reallocate STATE
      Deallocate (state)
      Allocate (state(lstate))

!     Initialize the generator to a repeatable sequence
      ifail = 0
      Call g05kff(genid,subid,seed,lseed,state,lstate,ifail)

!     Read in sample size and number of permutations
      Read (nin,*) n, nperm

      Allocate (indx(n))

      Write (nout,99998) nperm, ' Permutations of first ', n, ' integers'
      Write (nout,*)

!     Permutate NPERM times
      Do j = 1, nperm
!       Set up the index vector
        Do i = 1, n
          indx(i) = i
        End Do

!       Call the permutation routine
        ifail = 0
        Call g05ncf(indx,n,state,ifail)
```

```
!       Display the results
        Write (nout,99999) indx(1:n)
      End Do

99999 Format (1X,8I3)
99998 Format (1X,I0,A,I0,A)
    End Program g05ncfe
```

10.2 Program Data

```
G05NCF Example Program Data
1  1  1762543     :: GENID,SUBID,SEED(1)
8  10             :: N,NPERM
```

10.3 Program Results

```
 G05NCF Example Program 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
```