NAG Library Routine Document

G05NDF

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

G05NDF selects a pseudorandom sample without replacement from an integer vector.

2 Specification

```
SUBROUTINE G05NDF (IPOP, N, ISAMPL, M, STATE, IFAIL)
INTEGER IPOP(N), N, ISAMPL(M), M, STATE(*), IFAIL
```

3 Description

G05NDF selects m elements from a population vector IPOP of length n and places them in a sample vector ISAMPL. Their order in IPOP will be preserved in ISAMPL. Each of the $\binom{n}{m}$ possible combinations of elements of ISAMPL may be regarded as being equally probable.

For moderate or large values of n it is theoretically impossible that all combinations of size m may occur, unless m is near 1 or near n. This is because $\binom{n}{m}$ exceeds the cycle length of any of the base generators. For practical purposes this is irrelevant, as the time taken to generate all possible combinations 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 G05NDF.

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

1: IPOP(N) – INTEGER array *Input*

On entry: the population to be sampled.

2: N – INTEGER *Input*

On entry: the number of elements in the population vector to be sampled.

Constraint: N $\geq$ 1.

3: ISAMPL(M) – INTEGER array *Output*

On exit: the selected sample.

4: M – INTEGER *Input*

On entry: the sample size.

Constraint: 1 $\leq$ M $\leq$ N.

5: 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.

6: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, −1 or 1. If you are unfamiliar with this parameter you should refer to Section 3.3 in the Essential Introduction 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 parameter, 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 = ⟨*value*⟩.
Constraint: N ≥ 1.

IFAIL = 4

On entry, M = ⟨*value*⟩ and N = ⟨*value*⟩.
Constraint: 1 ≤ M ≤ N.

IFAIL = 5

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 3.8 in the Essential Introduction for further information.

IFAIL = −399

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

See Section 3.7 in the Essential Introduction for further information.

IFAIL = −999

Dynamic memory allocation failed.

See Section 3.6 in the Essential Introduction for further information.

7 Accuracy

Not applicable.

8 Parallelism and Performance

G05NDF 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

The time taken by G05NDF is of order n.

In order to sample other kinds of vectors, or matrices of higher dimension, the following technique may be used:

(a) set IPOP$(i) = i$, for $i = 1, 2, \ldots, n$;

(b) use G05NDF to take a sample from IPOP and put it into ISAMPL;

(c) use the contents of ISAMPL as a set of indices to access the relevant vector or matrix.

In order to divide a population into several groups, G05NCF is more efficient.

10 Example

In the example program random samples of size $1, 2, \ldots, 8$ are selected from a vector containing the first eight positive integers in ascending order. The samples are generated and printed for each sample size by a call to G05NDF after initialization by G05KFF.

10.1 Program Text

```
    Program g05ndfe

!     G05NDF Example Program Text

!     Mark 25 Release. NAG Copyright 2014.

!     .. Use Statements ..
      Use nag_library, Only: g05kff, g05ndf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: lseed = 1, nin = 5, nout = 6
!     .. Local Scalars ..
      Integer                          :: genid, i, ifail, lstate, m, n, pm,   &
                                          subid
!     .. Local Arrays ..
      Integer, Allocatable             :: ipop(:), isampl(:), state(:)
      Integer                          :: seed(lseed)
!     .. Executable Statements ..
      Write (nout,*) 'G05NDF 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 initialiser 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
      Read (nin,*) n

      Allocate (ipop(n))

!     Display title
      Write (nout,99999) ' Samples from the first ', n, ' integers'
      Write (nout,*)
      Write (nout,*) ' Sample size     Values'

!     Initialize the population
      Do i = 1, n
        ipop(i) = i
      End Do

!     Dummy allocation
      Allocate (isampl(1))

!     Loop over different sample sizes
      pm = 0
d_lp: Do
        Read (nin,*,Iostat=ifail) m
        If (ifail/=0) Then
          Exit d_lp
        End If

!       Reallocate ISAMPL
        If (pm/=m) Then
          Deallocate (isampl)
          Allocate (isampl(m))
          pm = m
        End If

!       Generate sample
        ifail = 0
        Call g05ndf(ipop,n,isampl,m,state,ifail)

!       Display the results
        Write (nout,99998) m, isampl(1:m)
      End Do d_lp

99999 Format (1X,A,I0,A)
99998 Format (1X,I6,9X,8(1X,I3))
    End Program g05ndfe
```

10.2 Program Data

```
G05NDF Example Program Data
1  1  1762543     :: GENID,SUBID,SEED(1)
8                 :: N
1                 :: List of sample sizes (M)
2
3
4
5
6
7
8
```

10.3 Program Results

```
 G05NDF Example Program Results

  Samples from the first 8 integers

  Sample size     Values
      1            2
      2            3   6
      3            1   5   7
      4            2   6   7   8
      5            1   2   3   4   8
      6            1   3   4   5   6   7
      7            1   3   4   5   6   7   8
      8            1   2   3   4   5   6   7   8
```