

NAG Library Routine Document

G05TGF

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

G05TGF generates a sequence of n variates, each consisting of k pseudorandom integers, from the discrete multinomial distribution with k outcomes and m trials, where the outcomes have probabilities p_1, p_2, \dots, p_k respectively.

2 Specification

```
SUBROUTINE G05TGF (MODE, N, M, K, P, R, LR, STATE, X, LDX, IFAIL)
INTEGER          MODE, N, M, K, LR, STATE(*), X(LDX,K), LDX, IFAIL
REAL (KIND=nag_wp) P(K), R(LR)
```

3 Description

G05TGF generates a sequence of n groups of k integers $x_{i,j}$, for $j = 1, 2, \dots, k$ and $i = 1, 2, \dots, n$, from a multinomial distribution with m trials and k outcomes, where the probability of $x_{i,j} = I_j$ for each $j = 1, 2, \dots, k$ is

$$P(i_1 = I_1, \dots, i_k = I_k) = \frac{m!}{\prod_{j=1}^k I_j!} \prod_{j=1}^k p_j^{I_j} = \frac{m!}{I_1! I_2! \dots I_k!} p_1^{I_1} p_2^{I_2} \dots p_k^{I_k},$$

where

$$\sum_{j=1}^k p_j = 1 \quad \text{and} \quad \sum_{j=1}^k I_j = m.$$

A single trial can have several outcomes (k) and the probability of achieving each outcome is known (p_j). After m trials each outcome will have occurred a certain number of times. The k numbers representing the numbers of occurrences for each outcome after m trials is then a single sample from the multinomial distribution defined by the parameters k , m and p_j , for $j = 1, 2, \dots, k$. This routine returns n such samples.

When $k = 2$ this distribution is equivalent to the binomial distribution with parameters m and $p = p_1$ (see G05TAF).

The variates can be generated with or without using a search table and index. If a search table is used then it is stored with the index in a reference vector and subsequent calls to G05TGF with the same parameter values can then use this reference vector to generate further variates. The reference array is generated only for the outcome with greatest probability. The number of successes for the outcome with greatest probability is calculated first as for the binomial distribution (see G05TAF); the number of successes for other outcomes are calculated in turn for the remaining reduced multinomial distribution; the number of successes for the final outcome is simply calculated to ensure that the total number of successes is m .

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 G05TGF.

4 References

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

5 Parameters

- 1: MODE – INTEGER *Input*
On entry: a code for selecting the operation to be performed by the routine.
 MODE = 0
 Set up reference vector only.
 MODE = 1
 Generate variates using reference vector set up in a prior call to G05TGF.
 MODE = 2
 Set up reference vector and generate variates.
 MODE = 3
 Generate variates without using the reference vector.
Constraint: MODE = 0, 1, 2 or 3.
- 2: N – INTEGER *Input*
On entry: n , the number of pseudorandom numbers to be generated.
Constraint: $N \geq 0$.
- 3: M – INTEGER *Input*
On entry: m , the number of trials of the multinomial distribution.
Constraint: $M \geq 0$.
- 4: K – INTEGER *Input*
On entry: k , the number of possible outcomes of the multinomial distribution.
Constraint: $K \geq 2$.
- 5: P(K) – REAL (KIND=nag_wp) array *Input*
On entry: contains the probabilities p_j , for $j = 1, 2, \dots, k$, of the k possible outcomes of the multinomial distribution.
Constraint: $0.0 \leq P(j) \leq 1.0$ and $\sum_{j=1}^k P(j) = 1.0$.
- 6: R(LR) – REAL (KIND=nag_wp) array *Communication Array*
On entry: if MODE = 1, the reference vector from the previous call to G05TGF.
 If MODE = 3, R is not referenced.
On exit: if MODE \neq 3, the reference vector.
- 7: LR – INTEGER *Input*
Note: for convenience p_{max} will be used here to denote the expression $p_{max} = \max_j(P(j))$.
On entry: the dimension of the array R as declared in the (sub)program from which G05TGF is called.
Suggested value:
 if MODE \neq 3, $LR = 30 + 20 \times \sqrt{M \times p_{max} \times (1 - p_{max})}$;
 otherwise LR = 1.

Constraints:

if $\text{MODE} = 0$ or 2 ,

$$\text{LR} > \min(\text{M}, \text{INT} [\text{M} \times p_m ax + 7.25 \times \sqrt{\text{M} \times p_m ax \times (1 - p_m ax)} + 8.5]),$$

$$- \max(0, \text{INT} [\text{M} \times p_m ax - 7.25 \times \sqrt{\text{M} \times p_m ax \times (1 - p_m ax)}]) + 9,$$
 if $\text{MODE} = 1$, LR must remain unchanged from the previous call to G05TGF.

8: 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.

9: X(LDX, K) – INTEGER array *Output*

On exit: the first n rows of $X(i, j)$ each contain k pseudorandom numbers representing a k -dimensional variate from the specified multinomial distribution.

10: LDX – INTEGER *Input*

On entry: the first dimension of the array X as declared in the (sub)program from which G05TGF is called.

Constraint: $\text{LDX} \geq \text{N}$.

11: 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 = 1

On entry, $\text{MODE} = \langle \text{value} \rangle$.
 Constraint: $\text{MODE} = 0, 1, 2$ or 3 .

IFAIL = 2

On entry, $\text{N} = \langle \text{value} \rangle$.
 Constraint: $\text{N} \geq 0$.

IFAIL = 3

On entry, $\text{M} = \langle \text{value} \rangle$.
 Constraint: $\text{M} \geq 0$.

IFAIL = 4

On entry, $K = \langle value \rangle$.
Constraint: $K \geq 2$.

IFAIL = 5

On entry, at least one element of the vector P is less than 0.0 or greater than 1.0.
On entry, the sum of the elements of P do not equal one.

IFAIL = 6

On entry, some of the elements of the array R have been corrupted or have not been initialized.
The value of M or K is not the same as when R was set up in a previous call.
Previous value of M = $\langle value \rangle$ and M = $\langle value \rangle$.
Previous value of K = $\langle value \rangle$ and K = $\langle value \rangle$.

IFAIL = 7

On entry, LR is too small when MODE = 0 or 2: LR = $\langle value \rangle$, minimum length required = $\langle value \rangle$.

IFAIL = 8

On entry, STATE vector has been corrupted or not initialized.

IFAIL = 10

On entry, LDX = $\langle value \rangle$ and N = $\langle value \rangle$.
Constraint: LDX \geq N.

IFAIL = 210

On entry, LDX = $\langle value \rangle$ and K = $\langle value \rangle$.
Constraint: LDX \geq K.

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

G05TGF 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 reference vector for only one outcome can be set up because the conditional distributions cannot be known in advance of the generation of variates. The outcome with greatest probability of success is chosen for the reference vector because it will have the greatest spread of likely values.

10 Example

This example prints 20 pseudorandom k -dimensional variates from a multinomial distribution with $k = 4$, $m = 6000$, $p_1 = 0.08$, $p_2 = 0.1$, $p_3 = 0.8$ and $p_4 = 0.02$, generated by a single call to G05TGF, after initialization by G05KFF.

10.1 Program Text

```

Program g05tgfe

!      G05TGF Example Program Text

!      Mark 25 Release. NAG Copyright 2014.

!      .. Use Statements ..
Use nag_library, Only: g05kff, g05tgf, nag_wp, x04eaf
!      .. Implicit None Statement ..
Implicit None
!      .. Parameters ..
Integer, Parameter          :: lseed = 1, maxlr = 5000, nin = 5,      &
                               nout = 6
!      .. Local Scalars ..
Real (Kind=nag_wp)         :: pmax
Integer                    :: genid, ifail, k, ldx, lr, lstate, m, &
                               mode, n, subid
!      .. Local Arrays ..
Real (Kind=nag_wp), Allocatable :: p(:), r(:)
Integer                        :: seed(lseed)
Integer, Allocatable          :: state(:), x(:, :)
!      .. Intrinsic Procedures ..
Intrinsic                    :: int, maxval, real, sqrt
!      .. Executable Statements ..
Write (nout,*) 'G05TGF Example Program Results'
Write (nout,*)
Flush (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

!      Read in the distribution parameters
      Read (nin,*) m, k

      ldx = n
      Allocate (x(ldx,k),p(k))

!      Read in probabilities
      Read (nin,*) p(1:k)

!      Use suggested value for LR
      pmax = maxval(p(1:k))
      lr = int(3.0E1_nag_wp+2.0E1_nag_wp*sqrt(real(m, &
        kind=nag_wp)*pmax*(1.0E0_nag_wp-pmax)))

!      If R is a reasonable size use MODE = 2
!      else do not reference R and use MODE = 3
      If (lr<maxlr) Then
        mode = 2
      Else
        mode = 3
        lr = 0
      End If

      Allocate (r(lr))

!      Generate the variates
      ifail = 0
      Call g05tgf(mode,n,m,k,p,r,lr,state,x,ldx,ifail)

!      Display the variates
      ifail = 0
      Call x04eaf('General',' ',n,k,x,ldx,' ',ifail)

      End Program g05tgfe

```

10.2 Program Data

```

G05TGF Example Program Data
1 1 1762543      :: GENID,SUBID,SEED(1)
20              :: N
6000 4          :: M,K
0.08 0.1 0.8 0.02 :: P

```

10.3 Program Results

G05TGF Example Program Results

	1	2	3	4
1	468	603	4811	118
2	490	630	4761	119
3	482	575	4821	122
4	495	591	4826	88
5	512	611	4761	116
6	474	601	4800	125
7	485	595	4791	129
8	468	582	4825	125
9	485	598	4800	117
10	485	573	4814	128
11	501	634	4749	116
12	482	618	4780	120
13	470	584	4810	136
14	479	642	4750	129
15	476	608	4807	109

16	473	631	4782	114
17	509	596	4778	117
18	450	565	4877	108
19	484	556	4840	120
20	466	615	4802	117
