

NAG Library Routine Document

G05TCF

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

G05TCF generates a vector of pseudorandom integers from the discrete geometric distribution with probability p of success at a trial.

2 Specification

```
SUBROUTINE G05TCF (MODE, N, P, R, LR, STATE, X, IFAIL)
  INTEGER          MODE, N, LR, STATE(*), X(N), IFAIL
  REAL (KIND=nag_wp) P, R(LR)
```

3 Description

G05TCF generates n integers x_i from a discrete geometric distribution, where the probability of $x_i = I$ (a first success after $I + 1$ trials) is

$$P(x_i = I) = p \times (1 - p)^I, \quad I = 0, 1, \dots$$

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 G05TCF with the same parameter value can then use this reference vector to generate further variates. If the search table is not used (as recommended for small values of p) then a direct transformation of uniform variates is used.

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 G05TCF.

4 References

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

5 Arguments

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 G05TCF.

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: P – REAL (KIND=nag_wp) *Input*
On entry: the parameter p of the geometric distribution representing the probability of success at a single trial.
Constraint: **machine precision** $\leq P \leq 1.0$ (see X02AJF).
- 4: R(LR) – REAL (KIND=nag_wp) array *Communication Array*
On entry: if $\text{MODE} = 1$, the reference vector from the previous call to G05TCF.
 If $\text{MODE} = 3$, R is not referenced.
On exit: if $\text{MODE} \neq 3$, the reference vector.
- 5: LR – INTEGER *Input*
On entry: the dimension of the array R as declared in the (sub)program from which G05TCF is called.
Suggested value:
 if $\text{MODE} \neq 3$, $\text{LR} = 8 + 42/P$ approximately (see Section 9);
 otherwise $\text{LR} = 1$.
Constraints:
 if $\text{MODE} = 0$ or 2 , $\text{LR} \geq 30/P + 8$;
 if $\text{MODE} = 1$, LR should remain unchanged from the previous call to G05TCF.
- 6: 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.
- 7: X(N) – INTEGER array *Output*
On exit: the n pseudorandom numbers from the specified geometric distribution.
- 8: 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:

$IFAIL = 1$

On entry, $MODE = \langle value \rangle$.

Constraint: $MODE = 0, 1, 2$ or 3 .

$IFAIL = 2$

On entry, $N = \langle value \rangle$.

Constraint: $N \geq 0$.

$IFAIL = 3$

On entry, $P = \langle value \rangle$.

Constraint: *machine precision* $\leq P \leq 1.0$.

P is so small that LR would have to be larger than the largest representable integer. Use $MODE = 3$ instead. $P = \langle value \rangle$

$IFAIL = 4$

On entry, some of the elements of the array R have been corrupted or have not been initialized.

P is not the same as when R was set up in a previous call.

Previous value of $P = \langle value \rangle$ and $P = \langle value \rangle$.

$IFAIL = 5$

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

$IFAIL = 6$

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

G05TCF 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 to set up the reference vector, if used, increases with the length of array R. However, if the reference vector is used, the time taken to generate numbers decreases as the space allotted to the index part of R increases. Nevertheless, there is a point, depending on the distribution, where this improvement becomes very small and the suggested value for the length of array R is designed to approximate this point.

If P is very small then the storage requirements for the reference vector and the time taken to set up the reference vector becomes prohibitive. In this case it is recommended that the reference vector is not used. This is achieved by selecting $\text{MODE} = 3$.

10 Example

This example prints 10 pseudorandom integers from a geometric distribution with parameter $p = 0.001$, generated by a single call to G05TCF, after initialization by G05KFF.

10.1 Program Text

```

Program g05tcfe

!      G05TCF Example Program Text

!      Mark 26 Release. NAG Copyright 2016.

!      .. Use Statements ..
      Use nag_library, Only: g05kff, g05tcf, nag_wp, x02amf
!      .. Implicit None Statement ..
      Implicit None
!      .. Parameters ..
      Integer, Parameter          :: lseed = 1, maxlr = 5000, nin = 5,      &
                                   nout = 6
!      .. Local Scalars ..
      Real (Kind=nag_wp)          :: p
      Integer                     :: genid, ifail, lr, lstate, mode, n,      &
                                   subid
!      .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable :: r(:)
      Integer                         :: seed(lseed)
      Integer, Allocatable           :: state(:), x(:)
!      .. Intrinsic Procedures ..
      Intrinsic                     :: int
!      .. Executable Statements ..
      Write (nout,*) 'G05TCF 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
!      Read (nin,*) n

!      Read in the distribution parameters
!      Read (nin,*) p

!      Use suggested value for LR
!      If (p<x02amf()) Then
!      P is too close to 0.0 to calculate LR, so
!      set to MAXLR, which means we will use MODE = 3
!      lr = maxlr
!      Else
!      lr = int(8.0E0_nag_wp+4.2E1_nag_wp/p)
!      End If

!      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 (x(n),r(lr))

!      Generate the variates
!      ifail = 0
!      Call g05tcf(mode,n,p,r,lr,state,x,ifail)

!      Display the variates
!      Write (nout,99999) x(1:n)

99999 Format (1X,I12)
      End Program g05tcfe

```

10.2 Program Data

G05TCF Example Program Data

1	1	1762543	:: GENID,SUBID,SEED(1)
10			:: N
0.001			:: P

10.3 Program Results

G05TCF Example Program Results

```

451
2238
292
225
2256
708
955
239
696
397

```
