NAG Library Routine Document

G05TJF

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

G05TJF generates a vector of pseudorandom integers from the discrete Poisson distribution with mean λ.

2 Specification

```
SUBROUTINE G05TJF (MODE, N, LAMBDA, R, LR, STATE, X, IFAIL)

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

3 Description

G05TJF generates n integers x_i from a discrete Poisson distribution with mean λ, where the probability of $x_i = I$ is

$$P(x_i = I) = \frac{\lambda^I \times e^{-\lambda}}{I!}, \qquad I = 0, 1, \ldots,$$

where $\lambda \geq 0$.

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 G05TJF with the same parameter values can then use this reference vector to generate further variates. The reference array is found using a recurrence relation if λ is less than 50 and by Stirling's formula otherwise.

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 G05TJF.

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: 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 G05TJF.

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: $\mathrm{N} \geq 0$.

3: LAMBDA – REAL (KIND=nag_wp) *Input*

On entry: λ, the mean of the Poisson distribution.

Constraint: $\mathrm{LAMBDA} \geq 0.0$.

4: R(LR) – REAL (KIND=nag_wp) array *Communication Array*

On entry: if $\mathrm{MODE} = 1$, the reference vector from the previous call to G05TJF.

If $\mathrm{MODE} = 3$, R is not referenced by G05TJF.

On exit: the reference vector.

5: LR – INTEGER *Input*

On entry: the dimension of the array R as declared in the (sub)program from which G05TJF is called.

Suggested value:

if $\mathrm{MODE} \neq 3$, $\mathrm{LR} = 30 + 20 \times \sqrt{\mathrm{LAMBDA}} + \mathrm{LAMBDA}$;
otherwise $\mathrm{LR} = 1$.

Constraints:

if $\mathrm{MODE} = 0$ or 2,
if $\sqrt{\mathrm{LAMBDA}} > 7.15$, $\mathrm{LR} > 9 + \mathrm{int}\left(8.5 + 14.3 \times \sqrt{\mathrm{LAMBDA}}\right)$;
otherwise $\mathrm{LR} > 9 + \mathrm{int}\left(\mathrm{LAMBDA} + 7.15 \times \sqrt{\mathrm{LAMBDA}} + 8.5\right)$;

if $\mathrm{MODE} = 1$, LR must remain unchanged from the previous call to G05TJF.

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 Poisson distribution.

8: 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: $\mathrm{IFAIL} = 0$ 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 $\neq 0$, 1, 2 or 3.

IFAIL $= 2$

On entry, N < 0.

IFAIL $= 3$

On entry, LAMBDA < 0.0.

MODE $= 0$ or 2 and LAMBDA is such that LR would have to be larger than the largest representable integer. Use MODE $= 3$ in this case.

IFAIL $= 4$

On entry, LAMBDA is not the same as when R was set up in a previous call to G05TJF with MODE $= 0$ or 2.

On entry, the R vector was not initialized correctly, or has been corrupted.

IFAIL $= 5$

On entry, LR is too small when MODE $= 0$ or 2.

IFAIL $= 6$

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

7 Accuracy

Not applicable.

8 Further Comments

None.

9 Example

This example prints 10 pseudorandom integers from a Poisson distribution with mean $\lambda = 20$, generated by a single call to G05TJF, after initialization by G05KFF.

9.1 Program Text

```
    Program g05tjfe

!     G05TJF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: g05kff, g05tjf, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter                :: lseed = 1, maxlr = 5000, nin = 5,    &
                                           nout = 6
```

```
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: lambda
      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, sqrt
!     .. Executable Statements ..
      Write (nout,*) 'G05TJF 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

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

!     Use suggested value for LR
      lr = int(3.0E1_nag_wp+2.0E1_nag_wp*sqrt(lambda)+lambda)

!     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 g05tjf(mode,n,lambda,r,lr,state,x,ifail)

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

99999 Format (1X,I12)
    End Program g05tjfe
```

9.2 Program Data

```
G05TJF Example Program Data
1  1  1762543     :: GENID,SUBID,SEED(1)
10                :: N
20.0              :: LAMBDA
```

9.3 Program Results

```
 G05TJF Example Program Results

           21
           15
           23
           24
           14
           20
           19
           23
           20
           22
```