

NAG Library Routine Document

G05KJF

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

G05KJF allows for the generation of multiple, independent, sequences of pseudorandom numbers using the skip-ahead method.

2 Specification

```
SUBROUTINE G05KJF(N, STATE, IFAIL)
  INTEGER          N, STATE(lstate), IFAIL
```

3 Description

G05KJF adjusts a base generator to allow multiple, independent, sequences of pseudorandom numbers to be generated via the skip-ahead method (see the G05 Chapter Introduction for details).

If, prior to calling G05KJF the base generator defined by STATE would produce random numbers $x_1, x_2, x_3, \dots$, then after calling G05KJF the generator will produce random numbers $x_{n+1}, x_{n+2}, x_{n+3}, \dots$

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 G05KJF.

The skip-ahead algorithm can be used in conjunction with the NAG basic generator, both the Wichmann–Hill I and Wichmann–Hill II generators and the ACORN generator.

4 References

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

5 Parameters

1: N – INTEGER *Input*

On entry: n , the number of places to skip ahead.

Constraint: $N \geq 0$.

2: STATE(lstate) – INTEGER array *Communication Array*

Note: $lstate = LSTATE$, the dimension of STATE as supplied to the initialization routine 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.

3: 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.

On exit: IFAIL = 0 unless the routine detects an error (see Section 6).

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.**

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, $N < 0$.

$IFAIL = 2$

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

$IFAIL = 3$

On entry, cannot use the skip-ahead method with the base generator defined by STATE.

7 Accuracy

Not applicable.

8 Further Comments

Calling G05KJF and then generating a series of uniform values using G05SAF is more efficient than, but equivalent to, calling G05SAF and discarding the first n values. This may not be the case for distributions other than the uniform, as some distributional generators require more than one uniform variate to generate a single draw from the required distribution.

9 Example

This example initializes a base generator using G05KFF and then uses G05KJF to advance the sequence 50 places before generating five variates from a uniform distribution using G05SAF.

9.1 Program Text

```
*      G05KJF Example Program Text
*      Mark 22 Release. NAG Copyright 2006.
*      .. Parameters ..
      INTEGER          NOUT
      PARAMETER        (NOUT=6)
      INTEGER          MSTATE, MSEED, N, NV
      PARAMETER        (MSTATE=633,MSEED=1,N=50,NV=5)
*      .. Local Scalars ..
      INTEGER          GENID, I, IFAIL, LSEED, LSTATE, SUBID
*      .. Local Arrays ..
      DOUBLE PRECISION X(NV)
      INTEGER          SEED(MSEED), STATE(MSTATE)
*      .. External Subroutines ..
      EXTERNAL         G05KFF, G05KJF, G05SAF
*      .. Executable Statements ..
      WRITE (NOUT,*) 'G05KJF Example Program Results'
      WRITE (NOUT,*)
*      Initialize the seed
      SEED(1) = 1762543
*      GENID and SUBID identify the base generator
      GENID = 1
      SUBID = 1
      LSTATE = MSTATE
```

```

      LSEED = MSEED
*      Initialize the generator to a repeatable sequence
      IFAIL = 1
      CALL G05KFF(GENID,SUBID,SEED,LSEED,STATE,LSTATE,IFAIL)
      IF (IFAIL.NE.0) THEN
         WRITE (NOUT,99998) IFAIL
         GO TO 20
      END IF
*      Advance the sequence N places
      IFAIL = 1
      CALL G05KJF(N,STATE,IFAIL)
      IF (IFAIL.NE.0) THEN
         WRITE (NOUT,99997) IFAIL
         GO TO 20
      END IF
*      Generate a NV variates from a uniform distribution
      IFAIL = 1
      CALL G05SAF(NV,STATE,X,IFAIL)
      IF (IFAIL.NE.0) THEN
         WRITE (NOUT,99996) IFAIL
         GO TO 20
      END IF
*      Display the variates
      WRITE (NOUT,99999) (X(I),I=1,NV)
*
      20 CONTINUE
*
99999 FORMAT (1X,F10.4)
99998 FORMAT (1X,' ** G05KFF returned with IFAIL = ',I5)
99997 FORMAT (1X,' ** G05KJF returned with IFAIL = ',I5)
99996 FORMAT (1X,' ** G05SAF returned with IFAIL = ',I5)
      END

```

9.2 Program Data

None.

9.3 Program Results

G05KJF Example Program Results

```

0.2071
0.8413
0.8817
0.5494
0.5248

```
