NAG Fortran Library Routine Document

G05KAF

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

G05KAF returns a pseudo-random number taken from a uniform distribution between 0 and 1.

2 Specification

```
real FUNCTION G05KAF(IGEN, ISEED)
INTEGER             IGEN, ISEED(4)
```

3 Description

This routine returns the next pseudo-random number from a uniform (0,1) generator.

The particular generator used to generate random numbers is selected by the value set for the input parameter IGEN. Consult the G05 Chapter Introduction for details of the algorithms that can be used.

The current state of the chosen generator is saved in the integer array ISEED which should not be altered between successive calls. Initial states are set or re-initialised by a call to G05KBF (for a repeatable sequence if computed sequentially) or G05KCF (for a non-repeatable sequence).

G05LGF may be used to generate a vector of n pseudo-random numbers which, if computed sequentially using the same generator, are exactly the same as n successive values of G05KAF. On many machines G05LGF is likely to be much faster.

4 References

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

5 Parameters

1: IGEN – INTEGER *Input*

On entry: must contain the identification number for the generator to be used to return a pseudo-random number and should remain unchanged following initialisation by a prior call to one of the routines G05KBF or G05KCF.

2: ISEED(4) – INTEGER array *Input/Output*

On entry: contains values which define the current state of the selected generator.

On exit: contains updated values defining the new state of the selected generator.

6 Error Indicators and Warnings

None.

7 Accuracy

Not applicable.

8 Further Comments

The generator with the smallest period that can be selected is the basic generator. The period of the basic generator is 2^{57}.

Its performance has been analysed by the Spectral Test, see Section 3.3.4 of Knuth (1981), yielding the following results in the notation of Knuth (1981).

n	ν_n	Upper bound for ν_n
2	3.44×10^8	4.08×10^8
3	4.29×10^5	5.88×10^5
4	1.72×10^4	2.32×10^4
5	1.92×10^3	3.33×10^3
6	593	939
7	198	380
8	108	197
9	67	120

The right-hand column gives an upper bound for the values of ν_n attainable by any multiplicative congruential generator working modulo 2^{59}.

An informal interpretation of the quantities ν_n is that consecutive n-tuples are statistically uncorrelated to an accuracy of $1/\nu_n$. This is a theoretical result; in practice the degree of randomness is usually much greater than the above figures might support. More details are given in Knuth (1981), and in the references cited therein.

Note that the achievable accuracy drops rapidly as the number of dimensions increases. This is a property of all multiplicative congruential generators and is the reason why very long periods are needed even for samples of only a few random numbers.

9 Example

The example program prints the first five pseudo-random numbers from a uniform distribution between 0 and 1, generated by G05KAF after initialisation by G05KBF.

9.1 Program Text

Note: the listing of the example program presented below uses ***bold italicised*** terms to denote precision-dependent details. Please read the Users' Note for your implementation to check the interpretation of these terms. As explained in the Essential Introduction to this manual, the results produced may not be identical for all implementations.

```
*     G05KAF Example Program Text
*     Mark 20 Release. NAG Copyright 2001.
*     .. Parameters ..
      INTEGER          NOUT
      PARAMETER        (NOUT=6)
*     .. Local Scalars ..
      real             X
      INTEGER          I, IGEN
*     .. Local Arrays ..
      INTEGER          ISEED(4)
*     .. External Functions ..
      real             G05KAF
      EXTERNAL         G05KAF
*     .. External Subroutines ..
      EXTERNAL         G05KBF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'G05KAF Example Program Results'
      WRITE (NOUT,*)
*     Initialise the seed
      ISEED(1) = 1762543
      ISEED(2) = 9324783
      ISEED(3) = 42344
      ISEED(4) = 742355
*     IGEN identifies the stream.
      IGEN = 1
```

```
      CALL G05KBF(IGEN,ISEED)
*
      DO 20 I = 1, 5
         X = G05KAF(IGEN,ISEED)
         WRITE (NOUT,99999) X
   20 CONTINUE
      STOP
*
99999 FORMAT (1X,F10.4)
      END
```

9.2 Program Data

None.

9.3 Program Results

```
 G05KAF Example Program Results

     0.0893
     0.9510
     0.4064
     0.7432
     0.9498
```