# NAG Fortran Library Routine Document

# G05CAF

**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

G05CAF returns a pseudo-random number taken from a uniform distribution between 0 and 1.

## 2 Specification

```
real FUNCTION G05CAF(X)
real                 X
```

## 3 Description

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

The particular mechanism used to generate random numbers can be selected by a prior call to G05ZAF. Consult the G05 Chapter Introduction for details of the algorithms that can be used and refer to the G05ZAF routine document on how to select a generator mechanism. If a prior call to G05ZAF is not made the default mechanism is used; to determine the default mechanism chosen for your implementation, refer to the Users' Note.

The current state of each generator used is saved internally in the code. Initial states are set by default but the sequence may be re-initialised by a call to G05CBF (for a repeatable sequence if computed sequentially) or G05CCF (for a non-repeatable sequence). The current state may be saved by a call to G05CFF, and restored by a call to G05CGF.

G05FAF 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 G05CAF. On many machines G05FAF 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: X – ***real*** *Dummy*

A dummy argument (originally required by ANSI Fortran 66 syntax).

## 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).

| $n$ | $\nu_n$ | Upper bound for $\nu_n$ |
|---|---|---|
| 2 | $3.44 \times 10^8$ | $4.08 \times 10^8$ |
| 3 | $4.29 \times 10^5$ | $5.88 \times 10^5$ |
| 4 | $1.72 \times 10^4$ | $2.32 \times 10^4$ |
| 5 | $1.92 \times 10^3$ | $3.33 \times 10^3$ |
| 6 | 593 | 939 |
| 7 | 198 | 380 |
| 8 | 108 | 197 |
| 9 | 67 | 120 |

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

An informal interpretation of the quantities $\nu_n$ 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 G05CAF after initialisation by G05CBF.

The generator mechanism used is selected by an initial call to G05ZAF.

## 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.

```
*     G05CAF Example Program Text
*     Mark 20 Revised. NAG Copyright 2001.
*     .. Parameters ..
      INTEGER          NOUT
      PARAMETER        (NOUT=6)
*     .. Local Scalars ..
      real             X
      INTEGER          I
*     .. External Functions ..
      real             G05CAF
      EXTERNAL         G05CAF
*     .. External Subroutines ..
      EXTERNAL         G05CBF, G05ZAF
*     .. Executable Statements ..
      CALL G05ZAF('O')
      WRITE (NOUT,*) 'G05CAF Example Program Results'
      WRITE (NOUT,*)
      CALL G05CBF(0)
      DO 20 I = 1, 5
*
         X = G05CAF(X)
*
         WRITE (NOUT,99999) X
   20 CONTINUE
      STOP
*
99999 FORMAT (1X,F10.4)
      END
```

## 9.2 Program Data

None.

## 9.3 Program Results

```
 G05CAF Example Program Results

     0.7951
     0.2257
     0.3713
     0.2250
     0.8787
```