# NAG Toolbox for MATLAB

# g05ka

## 1 Purpose

g05ka returns a pseudo-random number taken from a uniform distribution between 0 and 1.

## 2 Syntax

```
[result, iseed] = g05ka(igen, iseed)
```

## 3 Description

g05ka 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-initialized by a call to g05kb (for a repeatable sequence if computed sequentially) or g05kc (for a non-repeatable sequence).

g05lg 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 this function. On many machines g05lg 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

### 5.1 Compulsory Input Parameters

1: **igen – int32 scalar**

Must contain the identification number for the generator to be used to return a pseudo-random number and should remain unchanged following initialization by a prior call to g05kb or g05kc.

2: **iseed**$(4)$ **– int32 array**

Contains values which define the current state of the selected generator.

### 5.2 Optional Input Parameters

None.

### 5.3 Input Parameters Omitted from the MATLAB Interface

None.

### 5.4 Output Parameters

1: **result – double scalar**

The result of the function.

2: **iseed(4) – int32 array**

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.

| $n$ | $\nu_n$ | 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

```
igen = int32(1);
iseed = [int32(1762543);
     int32(9324783);
     int32(42344);
     int32(742355)];
[result, iseedOut] = g05ka(igen, iseed)
```

```
result =
    0.0893
iseedOut =
     2299464
      907816
     5377688
     9683645
```