Author(s) : CDC: H. von Eicken, IBM: T. Lindelöf
Submitter :
Language : Assembler

Library: KERNLIB
Submitted: 07.12.1970
Revised: 15.09.1978

Uniform Random Numbers

OBSOLETE

Please note that this routine has been obsoleted in CNL 215. Users are advised not to use it any longer and to replace it in older programs. No maintenance for it will take place and it will eventually disappear.

Suggested replacement:
`RANMAR` (V113) or `RANECU` (V114) or `RANLUX` (V115)

`RNDM` generates uniformly distributed pseudo-random numbers in the interval (0,1) in type `REAL` and in the interval $(1, 2^{47} - 1)$ (CDC) or $(1, 2^{31} - 1)$ (IBM) in type `INTEGER`. The CDC version has a period of more than 10^{13}. The IBM period, however, is only about 5×10^8 which may not be good enough for some calculations. In that case `RNDM2` (V107) should be used instead.

Structure:

`SUBROUTINE` subprogram
User Entry Names: `IRNDM`, `RNDM`, `RDMIN`, `RDMOUT`

Usage:

```
Y = RNDM(X)
```

where `X` is a dummy argument (see **Notes**), sets `Y` to a pseudo-random number in the interval (0,1). `X` and `Y` are of type `REAL`.

```
I = IRNDM(X)
```

where `X` is a dummy argument (see **Notes**), sets `I` to an integer pseudo-random number in the interval $(1, 2^{47} - 1)$ on CDC, $(1, 2^{31} - 1)$ on IBM. `X` is of type `REAL` and `I` is of type `INTEGER`.

```
CALL RDMOUT(SEED)
```

replaces `SEED` by the current value of the integer pseudo-random number. This `SEED` may then be used to restart the sequence at this point, by a call to `RDMIN`. `SEED` is of type `REAL`.

```
CALL RDMIN(SEED)
```

replaces the current value of the integer pseudo-random number by the value of the variable `SEED`. `SEED` is of type `REAL`. The value of `SEED` should not be chosen by the user but should be obtained by a previous call to `RDMOUT`. If this is not complied with, the numbers generated may have serious defects in their randomness.

Method:

CDC:
Consider the sequence:

$$\begin{aligned} r_i &= \alpha r_{i-1} (\bmod 2^{47}) \text{ for } i = 1, 2, \ldots \\ \text{with } r_0 &= 2000\,0000\,0110\,6047\,1625_8 \\ \text{and } \alpha &= 2000\,0000\,3432\,7724\,4615_8 \end{aligned}$$

where r_0 and α are the unnormalised floating-point representation of the starting number and 5^{15} respectively. The j-th floating-point number R_j is obtained by packing r_j with an exponent (-47) and normalising it. This ensures that R_j falls in the interval (0,1).

The product αr_{j-1} is generated in a 96 bit accumulator. The integer number N_j returned is the low order 47 bits of the contents of this accumulator, except that the right-most 11 bits are replaced by those occupying bit positions 48-58. This replacement is done in order to increase the time period of the low order bits.

IBM: See write-up for `RNDM2` (V107).

Notes:

While the argument is dummy, in the sense that the generator makes no use of it, it must be noted that if a reference to `RNDM` occurs

- more than once within a Fortran statement, the argument to it should be different in each case;
- in a `DO`-loop, the argument must depend either directly or indirectly on the index of this loop.

These rules must be observed since the compilers, in their attempt to optimise the object code, assume that functions called with identical arguments return the same function value.

•