Author(s) : F. James
Library: MATHLIB
Submitter :
Submitted: 15.03.1994
Language : Fortran
Revised:

Uniform Random Numbers of Guaranteed Quality

`RANLUX` generates pseudorandom numbers uniformly distributed in the interval (0,1), the end points excluded. Each call produces an array of single-precision real numbers of which 24 bits of mantissa are random. The user can choose a **luxury level** which guarantees the quality required for his application. The lowest luxury level (zero) gives a fast generator which will fail some sophisticated tests of randomness; The highest level (four) is about five times slower but guarantees complete randomness. In all cases the period is greater than 10^{165}. Independent subsequences can be generated. Entries are provided for initialization and checkpointing.

Structure:

`SUBROUTINE` Subprograms
User Entry Names: `RANLUX`, `RLUXGO`, `RLUXAT`, `RLUXIN`, `RLUXUT`

Usage:

```
CALL RANLUX(RVEC,LEN)
```

returns a vector `RVEC` of `LEN` 32-bit random floating point numbers in the interval (0,1), the end points excluded. `RVEC` is an array of type `REAL` and of length `LEN` at least.

Luxury levels:

For simplicity, five standard luxury levels may be chosen (t is the time factor relative to level zero; for the definition of p, see **References**). Ref. 1. explains the method, Ref. 2. describes the Fortran implementation in more detail.

Level	p	t	
0	24	1	Equivalent to the original `RCARRY` of Marsaglia and Zaman, very long period, but fails many tests.
1	48	1.5	Considerable improvement in quality over level 0, now passes the gap test, but still fails spectral test.
2	97	2	Passes all known tests, but theoretically still defective.
3	223	3	`DEFAULT VALUE`. Any theoretically possible correlations have very small chance of being observed.
4	389	5	Highest possible luxury, all 24 bits chaotic.

As a rough indication of timing, `RNDM` (V104) is about t=0.5, `RANMAR` (V113) t=1, and `RANECU` (V114) t=2. Concerning the quality scale, `RNDM` is maybe good enough for moving fish around on a screen saver (if you are not afraid of getting some diagonal lines on your screen), `RANMAR` and `RANECU` both have quality which probably corresponds to a luxury level between 1 and 2, but this is based only on empirical testing and true quality may be lower.

No initialization is necessary if the user wants default values. Otherwise the following are available:

```
CALL RLUXGO(LUX,INT,K1,K2)
```

When K1 $=$ K2 $=$ 0, this call initializes the RANLUX generator from one 32-bit integer INT and sets the Luxury Level. If LUX is an integer between 0 and 4, it sets the luxury level as defined above. If LUX $>$ 24, it is taken as the value of p, which then can take on other values than those given in the table. If INT $=$ 0, default initialization is used and only the luxury level is set by LUX. Otherwise, every possible value of INT gives rise to a valid, independent sequence which will not overlap any sequence initialized with any other value of INT. The integers K1 and K2 are used for restarting the generator from a break point saved by RLUXAT.

```
CALL RLUXAT(LUX,INT,K1,K2)
```

dumps the four integers which can be used to restart the generator at this point by calling RLUXGO. RANLUX will then skip over K1 $+ 10^9 *$K2 numbers to reach the break point. A more efficient but less convenient method for restarting is offered by RLUXIN and RLUXUT.

```
CALL RLUXIN(IVEC)
```

restarts the generator from vector IVEC of 25 32-bit integers (see RLUXUT). IVEC is an array of type INTEGER and of length 25 at least.

```
CALL RLUXUT(IVEC)
```

outputs the current values of the 25 32-bit integer seeds, to be used for restarting.

References:

1. M. Lüscher, A portable high-quality random number generator for lattice field theory simulations, Computer Phys. Commun. **79** (1994), 100–110.
2. F. James, RANLUX: A Fortran implementation of the high-quality pseudorandom number generator of Lüscher, Computer Phys. Commun. **79** (1994) 111–114.

•