

## APPENDIX 7A

### PORTABLE C CODE FOR A PMMLCG

---

Here we present computer code in C to implement the PMMLCG defined by modulus  $m = m^* = 2^{31} - 1 = 2,147,483,647$  and multiplier  $a = a_2 = 630,360,016$ , discussed at the end of Sec. 7.2.2. The code shown here can be downloaded from [www.mhhe.com/law](http://www.mhhe.com/law). This code is based closely on the FORTRAN code of Marse and Roberts (1983), and it requires that integers between  $-m^*$  and  $m^*$  be represented and computed correctly. This generator has 100 different streams that are spaced 100,000 apart.

It is generally *not* recommended that this generator be used for serious real-world applications, since the combined MRG in App. 7B has much better statistical properties.

Figure 7.5 gives code for an ANSI-standard C (i.e., using function prototyping) version of this generator, in three functions, as detailed in the comments. Figure 7.6 gives a header file (`lcgrand.h`) that the user must `#include` to declare the functions. We have used this code on a variety of computers and compilers, and it was used in the C examples in Chaps. 1 and 2.

```
/* Prime modulus multiplicative linear congruential generator
Z[i] = (630360016 * Z[i-1]) (mod(pow(2,31) - 1)), based on Marse and Roberts'
portable FORTRAN random-number generator UNIRAN. Multiple (100) streams are
supported, with seeds spaced 100,000 apart. Throughout, input argument
"stream" must be an int giving the desired stream number. The header file
lcgrand.h must be included in the calling program (#include "lcgrand.h")
before using these functions.

Usage: (Three functions)

1. To obtain the next U(0,1) random number from stream "stream," execute
   u = lcgrand(stream);
   where lcgrand is a float function. The float variable u will contain the
   next random number.

2. To set the seed for stream "stream" to a desired value zset, execute
   lcgrandst(zset, stream);
   where lcgrandst is a void function and zset must be a long set to the
   desired seed, a number between 1 and 2147483646 (inclusive). Default
   seeds for all 100 streams are given in the code.

3. To get the current (most recently used) integer in the sequence being
   generated for stream "stream" into the long variable zget, execute
   zget = lcgrandgt(stream);
   where lcgrandgt is a long function. */

/* Define the constants. */

#define MODLUS 2147483647
#define MULT1  24112
#define MULT2  26143

/* Set the default seeds for all 100 streams. */
```

#### FIGURE 7.5

C code for the PMMLCG with  $m = 2^{31} - 1$  and  $a = 630,360,016$  based on Marse and Roberts (1983).

```

static long zrng[] =
{
    1,
    1973272912, 281629770, 20006270, 1280689831, 2096730329, 1933576050,
    913566091, 246780520, 1363774876, 604901985, 1511192140, 1259851944,
    824064364, 150493284, 242708531, 75253171, 1964472944, 1202299975,
    233217322, 1911216000, 726370533, 403498145, 993232223, 1103205531,
    762430696, 1922803170, 1385516923, 76271663, 413682397, 726466604,
    336157058, 1432650381, 1120463904, 595778810, 877722890, 1046574445,
    68911991, 2088367019, 748545416, 622401386, 2122378830, 640690903,
    1774806513, 2132545692, 2079249579, 78130110, 852776735, 1187867272,
    1351423507, 1645973084, 1997049139, 922510944, 2045512870, 898585771,
    243649545, 1004818771, 773686062, 403188473, 372279877, 1901633463,
    498067494, 2087759558, 493157915, 597104727, 1530940798, 1814496276,
    536444882, 1663153658, 855503735, 67784357, 1432404475, 619691088,
    119025595, 880802310, 176192644, 1116780070, 277854671, 1366580350,
    1142483975, 2026948561, 1053920743, 786262391, 1792203830, 1494667770,
    1923011392, 1433700034, 1244184613, 1147297105, 539712780, 1545929719,
    190641742, 1645390429, 264907697, 620389253, 1502074852, 927711160,
    364849192, 2049576050, 638580085, 547070247 };
/* Generate the next random number. */

float lcgrand(int stream)
{
    long zi, lowprd, hi31;

    zi = zrng[stream];
    lowprd = (zi & 65535) * MULT1;
    hi31 = (zi >> 16) * MULT1 + (lowprd >> 16);
    zi = ((lowprd & 65535) - MODLUS) +
        ((hi31 & 32767) << 16) + (hi31 >> 15);
    if (zi < 0) zi += MODLUS;
    lowprd = (zi & 65535) * MULT2;
    hi31 = (zi >> 16) * MULT2 + (lowprd >> 16);
    zi = ((lowprd & 65535) - MODLUS) +
        ((hi31 & 32767) << 16) + (hi31 >> 15);
    if (zi < 0) zi += MODLUS;
    zrng[stream] = zi;
    return (zi >> 7 | 1) / 16777216.0;
}

void lcgrandst (long zset, int stream) /* Set the current zrng for stream
                                     "stream" to zset. */
{
    zrng[stream] = zset;
}

long lcgrandgt (int stream) /* Return the current zrng for stream "stream". */
{
    return zrng[stream];
}

```

**FIGURE 7.5***(continued)*

```

/* The following 3 declarations are for use of the random-number generator
   lcgrand and the associated functions lcgrandst and lcgrandgt for seed
   management. This file (named lcgrand.h) should be included in any program
   using these functions by executing
       #include "lcgrand.h"
   before referencing the functions. */

float lcgrand(int stream);
void lcgrandst(long zset, int stream);
long lcgrandgt(int stream);

```

**FIGURE 7.6**

C header file (lcgrand.h) to accompany the C code in Fig. 7.5.