LCGs, Uniform<0,1>, etc.

Mihir Arjunwadkar

Centre for Modeling and Simulation
Savitribai Phule Pune University

Connotations of *randomness*

A sequence of numbers which

- lacks predictability
- lacks (easily) recognizable pattern or rule
- contains information which cannot be compressed into an equivalent but shorter representation
- is indistinguishable from realizations of a truly random process
- has statistical independence (in some contexts)
- ...

Imprecise as definition,
uses circular reasoning,
...
(but you know what is meant)

Mais quand une regle est fort composée, ce qui luy est conforme, passe pour irrégulier.

But when a rule is extremely complex, that which conforms to it passes for random.

Gottfried Wilhelm 

Discourse on Metaphysics (1686)

as quoted at `http://en.wikiquote.org/wiki/Randomness`

We may, therefore, use **deterministic** means to generate a sequence of numbers that **appear** random ...

$$X_{n+1} = (aX_n + c) \mod M$$

X_n, M, a, c: Integer

Seed $0 \leq X_0 < M$
Modulus $M \geq 1$
Multiplier $1 < a < M$
Increment $0 \leq c < M$

Completely deterministic: Given M, a, c,

same seed $\implies$ same sequence

Integer arithmetic

- X_n is a periodic sequence with *maximum* cycle length $= M$ because of the modulo operation.

- Bad choice of a, c can lead to the "bad" sequences

  For example, $a = c = 1$, any M, X_0. $M = 5, X_0 = 3 \implies$

  $3, 4, 0, 1, 2, 3, 4, 0, 1, 2, 3, 4, 0, 1, 2, 3, 4, 0, 1, 2, 3, 4, 0, 1, 2, \ldots$

- Shorter-than-M cycles

  For example, $M = 10, a = c = X_0 = 7 \implies$

  $7, 6, 9, 0, 7, 6, 9, 0, 7, 6, 9, 0, 7, 6, 9, 0, 7, 6, 9, 0, 7, 6, 9, 0, 7, \ldots$

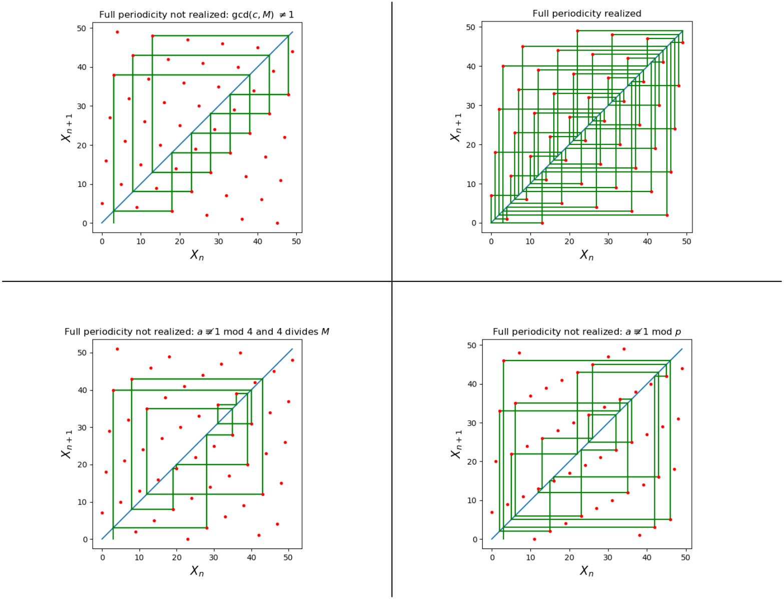
How are the parameters a, c, M chosen?

A LCG has full period M iff

1. $\gcd(c, M) = 1$.
2. $a \equiv 1 \mod p$ for each prime factor p of M.
3. $a \equiv 1 \mod 4$ if 4 divides M.

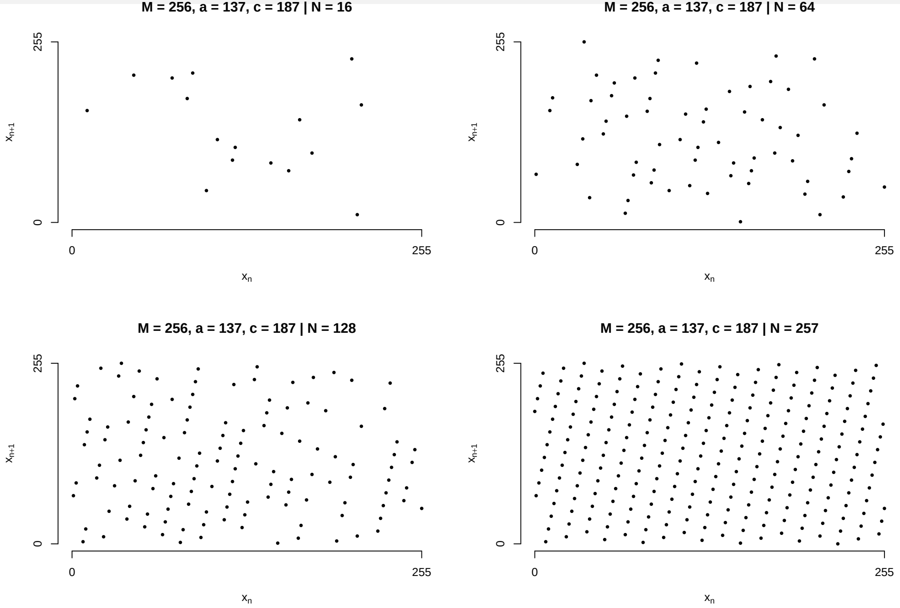
Brian D. Ripley, *Stochastic Simulation*, Wiley (1987)

How are the parameters a, c, M chosen?

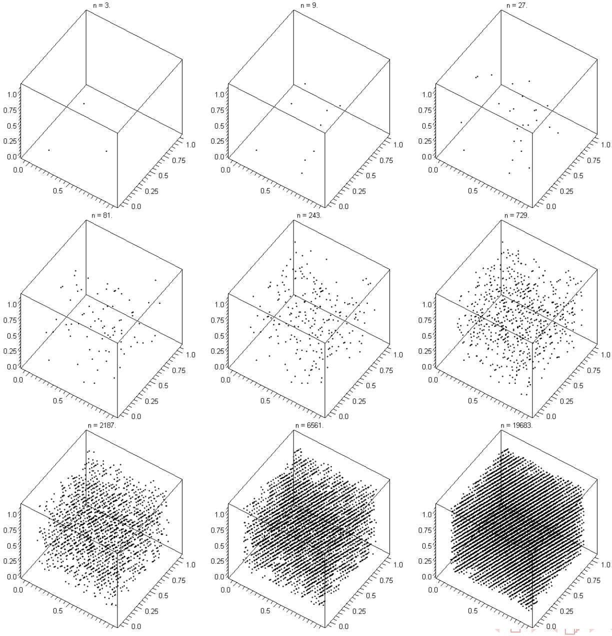


Courtesy: Snehal Shekatkar

Serial correlations: Marsaglia lattice



Serial correlations: Marsaglia lattice



http://en.wikipedia.org/wiki/File:Lcg_3d.gif

Breaking serial correlations

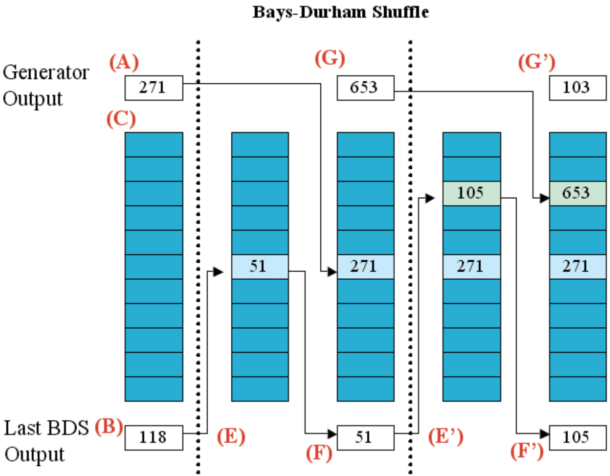
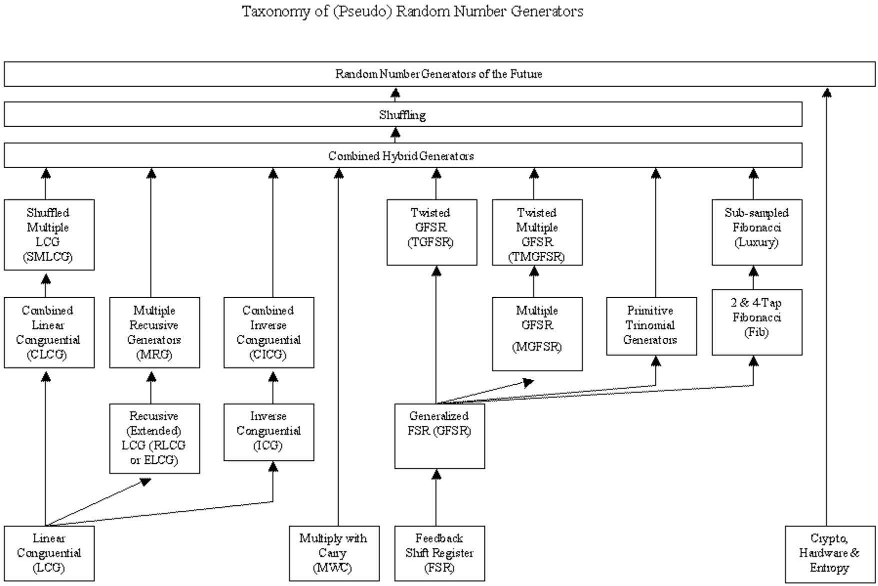


Figure 1 – Two cycles of a Bays-Durham Shuffle. Internal state consists of the last output of the Random Number Generator (A), the last output of the Bays-Durham Shuffle (B), and a table of random values (C). To generate a new output, an index (E) is created from the last output (B). The value in the table at this index becomes the next output (F). The value at the index is replaced with the current output of the generator (G), and the generator is updated to its next state.

http://www.sml.ee.upatras.gr/UploadedFiles/07-RNG0-!!!!!random_number_generators.pdf

A taxonomy of RNGs



http://www.sml.ee.upatras.gr/UploadedFiles/07-RNGO-!!!!!random_number_generators.pdf

The initial seed

- `/dev/random` and `/dev/urandom`:
  - `http://sourceware.org/ml/gsl-discuss/2004-q1/msg00071.html`
  - `http://www.2uo.de/myths-about-urandom/`
  - `http://en.wikipedia.org/?title=/dev/random`
- Concoct a seed from the machine clock or current time

Distribution of numbers in the LCG sequence

Assuming a full-period LCG, the numbers $0, \ldots, M-1$ occur with equal propensity $\implies$ uniform PMF over the set $0, \ldots, M-1$.

Why? Here is a suggestive argument:

- Except the modulo-M operation, the relationship between X_n and X_{n+1} is linear. If X_n has a uniform PMF, so will X_{n+1}.
- Geometrically, the module-M operation turns the interval $[0, M)$ into a circle by identifying M with 0. The LCG sequence can be thought of as going round-and-round over this circle with a constant speed and an additive increment at every step.

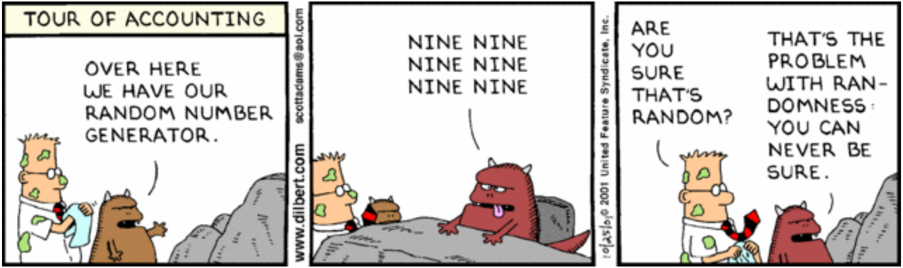
This can be numerically verified, but this is not a rigorous proof.

Uniform PMF over $0, \ldots, M$ can be transformed to a uniform distribution over $\langle 0, 1\rangle$ as

- Over $[0, 1)$: $U = X/M$
- Over $[0, 1]$: $U = X/(M - 1)$
- Over $(0, 1]$: $U = (X + 1)/M$
- Over $(0, 1)$: $U = (X + 1)/(M + 1)$

Note

- *Division operation* $/ \equiv$ *real division.*
- *Although we pretend to have a continuous interval* $\langle 0, 1\rangle$, U *is discrete.*
- *In practice,* U *is also represented as a representable floating-point number, and rounding may modulate this discreteness further.*


TOUR OF ACCOUNTING
OVER HERE WE HAVE OUR RANDOM NUMBER GENERATOR.
NINE NINE NINE NINE NINE NINE
ARE YOU SURE THAT'S RANDOM?
THAT'S THE PROBLEM WITH RAN-DOMNESS: YOU CAN NEVER BE SURE.
www.dilbert.com
scottadams@aol.com
© 2001 United Feature Syndicate, Inc.

General Perspective

- There is no end to conceivable tests
- A RNG that passes a sequences of tests $T_1, \ldots, T_N$ need not necessarily pass a new test T_{N+1}
- Conversely, passing some N tests does not guarantee randomness.
- An extensive literature survey:
  `http://www.ciphersbyritter.com/RES/RANDTEST.HTM`

- Knuth's spectral test
- DIEHARD
  http://stat.fsu.edu/pub/diehard/
- Monkey Tests
  http://dieharder.googlecode.com/svn/trunk/doc/monkey_tests.pdf
- dieharder
  http://www.phy.duke.edu/~rgb/General/dieharder.php
- TestU01
  http://simul.iro.umontreal.ca/testu01/tu01.html
- NIST RNG tools
  http://csrc.nist.gov/groups/ST/toolkit/rng/index.html
- http://www.cs.fsu.edu/~mascagni/research/testing.html
- ...

Empirical/Statistical Tests for Uniformity

- χ^2 tests on sequence, pairs, k-tuples, ...
- Kolmogorov-Smirnov
- ...

Empirical/Statistical Tests for Independence

- Runs test
- Gaps test
- Permutation tests
- Test for the Pearson correlation coefficient
  $H_0 : \rho = 0$ vs. $H_1 : \rho \neq 0$

LCG implementations

- Mersenne Twister

  http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/emt.html

  http://en.wikipedia.org/wiki/Mersenne_twister
- GNU Scientific Library

  http://www.gnu.org/s/gsl/
- Rmath standalone library
- SPRNG

  http://sprng.cs.fsu.edu/
- Unix/linux RNG: `drand48`
- Numerical Recipes

  Test NR codes thoroughly before using them!

Recommendations

- Use RNGs that have well-understood properties.
- Complex methods are not necessarily better.
- Use well-tested implementations.
- To see if a RNG is good enough for your application:
  - Run your application with two very different RNGs and see if they produce the same result.
  - Does your application produce results which can be traced back to any patterns related to the RNG?
  - Run as many tests as possible for the intended sequence length: `http://burtleburtle.net/bob/rand/testsfor.html`

References

- Brian D. Ripley, *Stochastic Simulation*, Wiley (1987).
- James E. Gentle, *Random Number Generation and Monte Carlo Methods*, Springer (2003).
- Luc Devroye, *Non-uniform Random Variate Generation*, Springer (1986). `http://luc.devroye.org/rnbookindex.html`
- Donald Knuth, *The Art of Computer Programming, Vol 2: Seminumerical Algorithms*, Addison-Wesley (1981), or any newer edition.