Stochastic Simulation
Random number generation

Bo Friis Nielsen

Applied Mathematics and Computer Science

Technical University of Denmark

2800 Kgs. Lyngby – Denmark

Email: bfn@imm.dtu.dk

Random number generation

- Uniform distribution
- Number theory
- Testing of random numbers
- Recommendations of random number generators

Summary

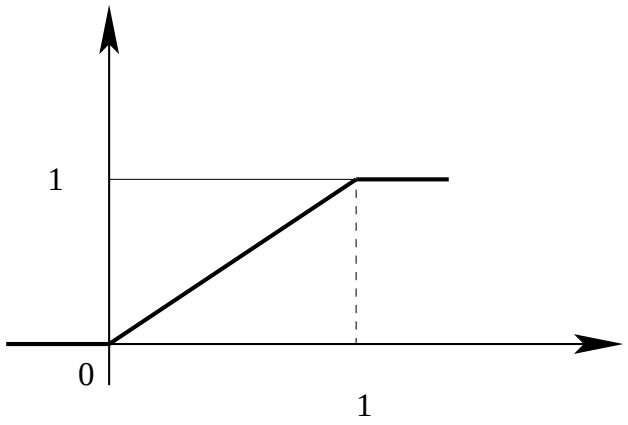
- We talk about generating **pseudo**random numbers
- There exists a large number of RNG's
- ... of varying quality
- Don't implement your own, except for fun or as a research project.
- Built-in RNG's should be checked before use
- ... at least in general-purpose development environments.
- Scientific computing environments typically have state-of-the-art RNG's that can be trusted.
- Any RNG will fail, if the circumstances are extreme enough.

History/background

- The need for random numbers evident
- Tables
- Physical generators. Lottery machines
- Need for computer generated numbers

Definition

- Uniform distribution $[0;\ 1]$.
- Randomness (independence).
- **Random numbers:** A sequence of independent random variable, U_i, uniformly distributed on $]0,\ 1[$



- Generate a sequence of independently and identically distributed $U(0, 1)$ numbers.
- One basic problem is computers do not work in $\mathbb{R}$

Random generation

Mechanics devices:

- Coin (head or tail)
- Dice (1-6)
- Monte-Carlo (Roulette) wheel
- Wheel of fortune
- Deck of cards
- Lotteries (Dansk tipstjeneste)

Other devices:

- electronic noise in a diode or resistor
- tables of random numbers

Definition of a RNG

An RNG is a computer algorithm that outputs a sequence of reals or integers, which appear to be

- Uniformly distributed on $[0; 1]$ or $\{0, \ldots, N-1\}$
- Statistically independent.

Caveats:

- "Appear to be" means: The sequence must have the same **relevant** statistical properties as I.I.D. uniformly distributed random variables
- With any finite precision format such as `double`, uniform on $[0; 1]$ can never be achieved.

1. Four digit integer
   (output divide by 10000)
2. square it.
3. Take the middle four digits
4. repeat

i	Z_i	U_i	Z_i^2
0	7182	0.7182	51,581,124
1	5811	0.5811	33,767,721
2	7677	0.7677	58,936,329
3	9363	0.9363	87,665,769
4	6657	0.6657	44,315,649
5	3156	0.3156	09,960,336
⋮	⋮	⋮	⋮

Might seem plausible - but rather dubious

Fibonacci

Leonardo of Pisa (pseudonym: Fibonacci) dealt in the book "Liber Abaci"(1202) with the integer sequence defined by:

$$x_i = x_{i-1} + x_{i-2} \quad i \geq 2 \quad x_0 = 1 \quad x_1 = 1$$

Fibonacci generator. Also called an additive congruential method.

$$\boxed{x_i = mod(\ x_{i-1} + x_{i-2},\ M\ )} \quad U_i = \frac{x_i}{M}$$

where $x = mod(\ y, M\ )$ is the modulus after division ie. $y - nM$ where $n = \lfloor y/M \rfloor$ Notice $x_i \in [0,\ M-1]$. Consequently, there is $M^2 - 1$ possible starting values.

Maximal length of period is $M^2 - 1$ which is only achieved for $M = 2,\ 3$.

Congruential Generator

The generator

$$U_i = mod(\ aU_{i-1},\ 1\ ) \quad U_i \in [0,\ 1]$$

illustrates the principle provided a is large, the last digits are retained.

Can be implemented as (x_i is an integer)

$$x_i = mod(\ ax_{i-1},\ M\ ) \quad U_i = \frac{x_i}{M}$$

Examples are $a = 23$ and $M = 10^8 + 1$.

Mid conclusion

- Initial state determine the whole sequence
- Potentially many different cycles
- Length of each cycle

If x_i can take N values, then the maximum length of a cycle is N.

Properties for a Random number generator

- Cycle length
- Randomness
- Speed
- Reproducible
- Portable

Linear Congruential Generator

LCG are defined as

$$x_i = mod(\ ax_{i-1} + c,\ M\ ) \qquad U_i = \frac{x_i}{M}$$

for a *multiplier* a, *shift* c and *modulus* M.

We will take a, c and x_0 such x_i lies in $(0,\ 1,\ \ldots\ , M-1)$ and it looks random.

Example: $M = 16,\ a = 5,\ c = 1$

With $x_0 = 3$: 0 1 6 15 12 13 2 11 8 9 14 7 4 5 10 3

Theorem 1

Maximum cycle length The LCG has full length if (and only if)

- M and c are relative prime.
- For each prime factor p of M, $mod(a, p) = 1$.
- if 4 is a factor of M, then $mod(a, 4) = 1$. Notice, If M is a prime, full period is attained only if $a = 1$.

Shuffling

eg. XOR between several generators.

- To enlarge period
- Improve randomness
- But not well understood
- LCGs widespread use, generally to be recommended

Mersenne Twister

Matsumoto and Nishimura, 1998

- A large structured linear feedback shift register
- Uses 19,937 bits of memory
- Has maximum period, i.e. $2^{19937} - 1$
- Has right distribution
- ... also joint distribution of 623 subsequent numbers
- Probably the best PRNG so far for stochastic simulation (not for cryptography).

RNGs in common environments

R: The Mersenne Twister is the default, many others can be chosen.

Python: Mersenne Twister chosen.

S-plus: XOR-shuffling between a congruential generator and a (Tausworthe) feedback shift register generator. The period is about $2^{62} \approx 4 \cdot 10^{18}$, but seed dependent (!).

Matlab 7.4 and higher: By default, the Mersenne Twister. Also one other available.

Characteristics

Definition: A sequence of *pseudo-random* numbers U_i is a deterministic sequence of numbers in $]0,\ 1[$ having the same relevant statistical properties as a sequence of random numbers.

The question is what are relevant statistical properties.

- Distribution type
- Randomness (independence, whiteness)