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

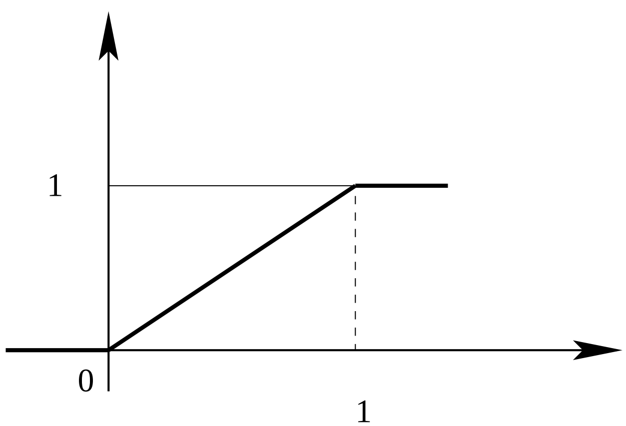
- We talk about generating **pseudo**random numbers
- There exist a large number of RNG's
- ... of varying quality
- Don't implement your own, except for fun or as a research project.
- Built-in RNG's should be checked before use
- ... at least in general-purpose development environments.
- Scientific 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).
- One basic problem is computers do not work in $\mathbb{R}$ **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.

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
$\vdots$	$\vdots$	$\vdots$	$\vdots$

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.

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

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

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

mimics the magnification effect of a roulette wheel provided a is large.

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
- How many different cycles
- Length of each cycle

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

Let us see when this occur

Properties for a Random generator

- Cycle length
- Randomness
- Speed
- Reproducible
- Portable

Linear Congruential Generator

LCG are defined as

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

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 $a = 1$.

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.

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.

Shuffling

eg. XOR between several generators.

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

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)

Testing random number generators

- Test for distribution type
  - ◇ Visual tests/plots
  - ◇ χ^2 test
  - ◇ Kolmogorov Smirnov test
- Test for independence
  - ◇ Visual tests/plots
  - ◇ Run test up/down
  - ◇ Run test length of runs
  - ◇ Test of correlation coefficients

Significance test

- We assume (known) model - *The hypothesis*
- We identify a certain characterising random variable - *The test statistic*
- We reject the hypothesis if the test statistic is an abnormal observation under the hypothesis

Key terms

- Hypothesis/Alternative
- Test statistic
- Significance level
- Accept/Critical area
- Power
- p-value

Test for distribution type χ^2 test

The general form of the test statistic is

$$T = \sum_{i=1}^{n_{\text{classes}}} \frac{(n_{\text{observed},i} - n_{\text{expected},i})^2}{n_{\text{expected},i}}$$

- The test statistic is to be evaluated with a χ^2 distribution with df degrees of freedom. df is generally $n_{\text{classes}} - 1 - m$ where m is the number of estimated parameters.

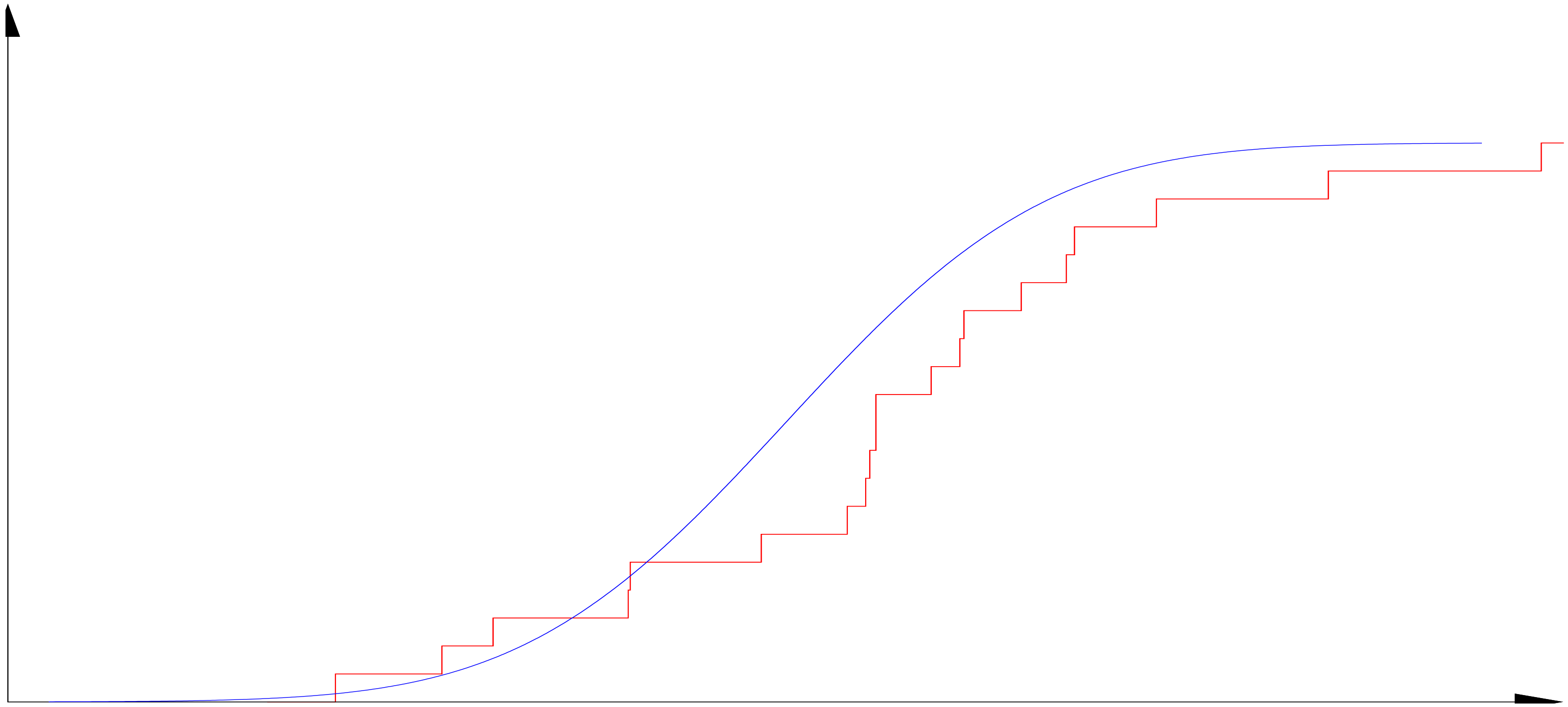
Test for distribution type Kolmogorov Smirnov test

- Compare empirical distribution function $F_n(x)$ with hypothesized distribution $F(x)$.
- For known parameters the test statistic does not depend on $F(x)$
- No grouping considerations needed
- Works only for completely specified distributions in the original version

Empirical distribution

20 $N(0,1)$ variates (sorted):

-2.20, -1.68, -1.43, -0.77, -0.76, -0.12, 0.30, 0.39, 0.41, 0.44, 0.44, 0.71, 0.85, 0.87, 1.15, 1.37, 1.41, 1.81, 2.65, 3.69



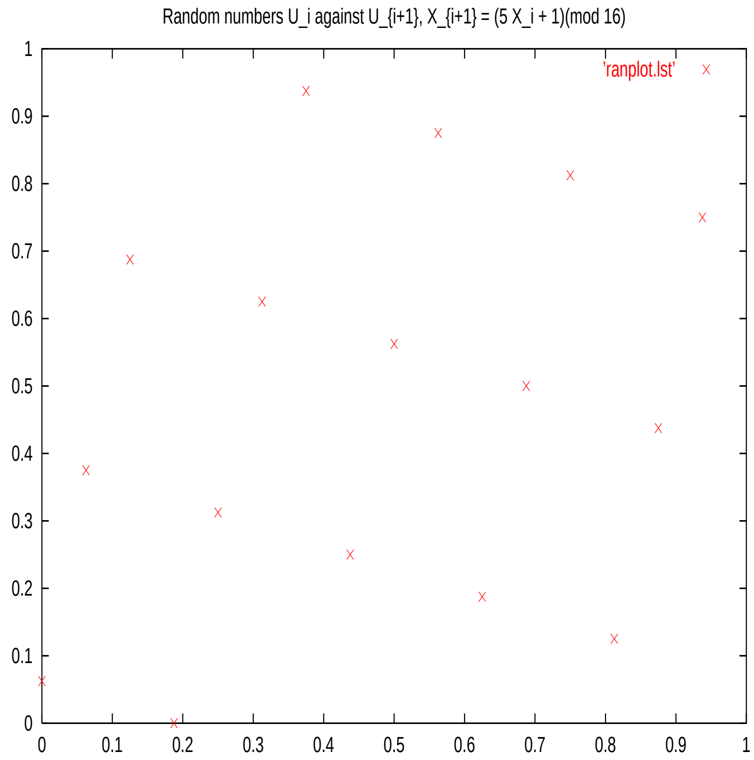
$$D_n = \sup_x \{|F_n(x) - F(x)|\}$$

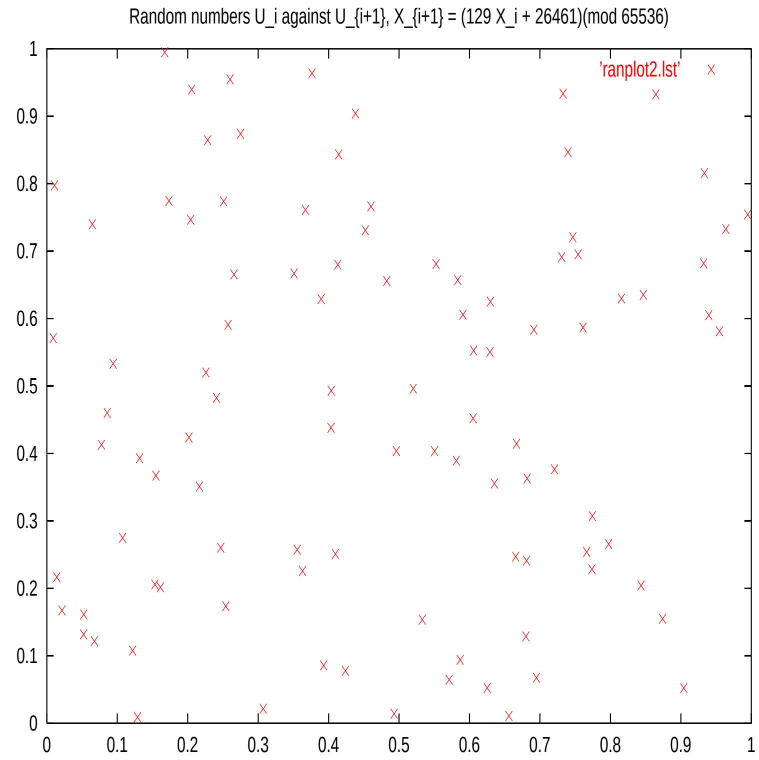
Test statistic and significance levels

		Level of significance $(1-\alpha)$				
Case	Adjusted test statistic	0.850	0.900	0.950	0.975	0.990
All parameters known	$\left(\sqrt{n}+0.12+\frac{0.11}{\sqrt{n}}\right)D_n$	1.138	1.224	1.358	1.480	1.628
$N(\bar{X}(n), S^2(n))$	$\left(\sqrt{n}-0.01+\frac{0.85}{\sqrt{n}}\right)D_n$	0.775	0.819	0.895	0.955	1.035
$\exp(\bar{X}(n))$	$\left(\sqrt{n}+0.26+\frac{0.5}{\sqrt{n}}\right)\left(D_n-\frac{0.2}{n}\right)$	0.926	0.990	1.094	1.190	1.308

Test for correlation - Visual tests

- Plot of U_{i+1} versus U_i





Run test I

Above/below

- The run test given in Conradsen, can be used by e.g. comparing with the median.
- The number of runs (above/below the median) is (asymptotically) distributed as

$$\mathbb{N}\left(2\frac{n_1 n_2}{n_1+n_2}+1, 2\frac{n_1 n_2(2n_1 n_2-n_1-n_2)}{(n_1+n_2)^2(n_1+n_2-1)}\right)$$

where n_1 is the number of samples above and n_2 is the number below.

Run tests II

Up/Down A test specifically designed for testing random number generators is the UP/DOWN run test, see e.g. Donald E. Knuth, The Art of Computer Programming Volume 2, 1998, pp. 66-.

The sequence:

0.54,0.67,0.13,0.89,0.33,0.45,0.90,0.01,0.45,0.76,0.82,0.24, 0.17

has runs of length 2,2,3,4,1, ...

Run tests II

Generate n random numbers.The observed number of runs of length $1, \ldots, 5$ and $\geq$6 are
recorded in the vector $\boldsymbol{R}$. The test statistic is calculated by:

$$Z = \frac{1}{n-6}(\boldsymbol{R} - n\boldsymbol{B})^T A(\boldsymbol{R} - n\boldsymbol{B})$$

$$A = \begin{bmatrix} 4529.4 & 9044.9 & 13568 & 18091 & 22615 & 27892 \\ 9044.9 & 18097 & 27139 & 36187 & 45234 & 55789 \\ 13568 & 27139 & 40721 & 54281 & 67852 & 83685 \\ 18091 & 36187 & 54281 & 72414 & 90470 & 111580 \\ 22615 & 45234 & 67852 & 90470 & 113262 & 139476 \\ 27892 & 55789 & 83685 & 111580 & 139476 & 172860 \end{bmatrix} \qquad \boldsymbol{B} = \begin{bmatrix} \frac{1}{6} \\ \frac{5}{24} \\ \frac{11}{120} \\ \frac{19}{720} \\ \frac{29}{5040} \\ \frac{1}{840} \end{bmatrix}$$

The test statistic is compared with a $\chi^2(6)$ distribution. $n > 4000$

Correlation coefficients

- the estimated correlation

$$c_h = \frac{1}{n-h} \sum_{i=1}^{n-h} U_i U_{i+h} \in \mathbb{N}\left(0.25, \frac{7}{144n}\right)$$

Exercise 1

- Write a program generating 10.000 (pseudo-) random numbers and present these numbers in a histogramme (e.g. 10 classes).
  - First implement the LCG yourself by experimenting with different values of "a", "b" and "c".
  - Evaluate the quality of the generators by graphical descriptive statistices (histogrammes, scatter plots) and statistical tests (χ^2,Kolmogorov-Smirnov, run-tests, and correlation test.
  - Then apply a system available generator (e.g. drand48() C, and C++) and perform the various statistical tests for this also.