RANDOMNESS

Pau Fonseca i Casas, pau@fib.upc.edu

Randomness

Understanding the thinks that we cannot understand.

Non determinist system

- We cannot predict the system behavior
  - The system depends on uncontrollable factors named randomness.

Indeterministic world

- “In extreme conditions, like in the early universe, or in high energy particle collisions, there could be significant loss of information. This would lead to unpredictability, in the evolution of the universe.“
- **STEPHEN HAWKING**

Deterministic world

- Einstein, when considering light as bundles of energy (photons), impelled the birth of quantum physics: It extended the theory of Planck's how much to the energy of the emitted light. Although he was one of the pioneers of quantum, in 1927 he entered into debate and conflict with the physicists who developed it, especially with Werner Heisenberg and Niels Bohr, members of the Copenhagen School.
- Both the classical worldview or Newtonian and relativistic are deterministic. What do that mean? It means that the physical world is completely predictable, that is, if identical experiments are made, the results are expected to be identical as well.
- In the quantum world that is not exactly the case, the events are not so predictable; It does not dominate the principle of classical determination, but the principle of indetermination. Einstein didn't want to accept that the probability is a characteristic of the behavior of nature, contrary to causes and effects occur with an absolute logic. It does not accept that chance and indetermination are a characteristic of the physical world. In this sense, his phrase is famous: **"God does not play dice with the Universe"**.

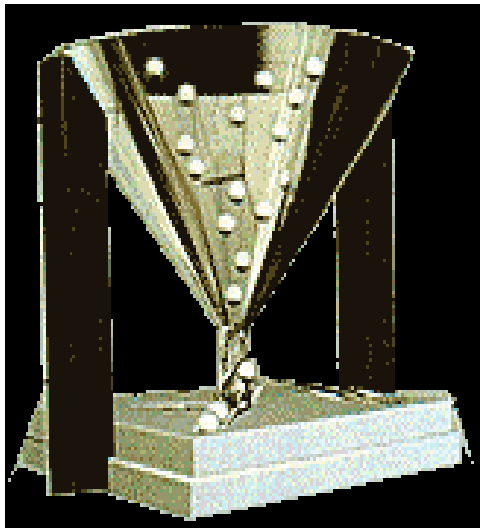
Deterministic world

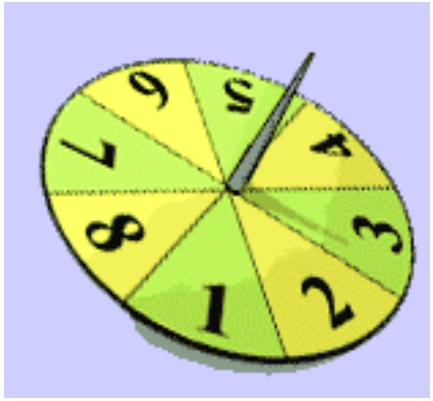
- Einstein and Bohr

RNG

Random number generation

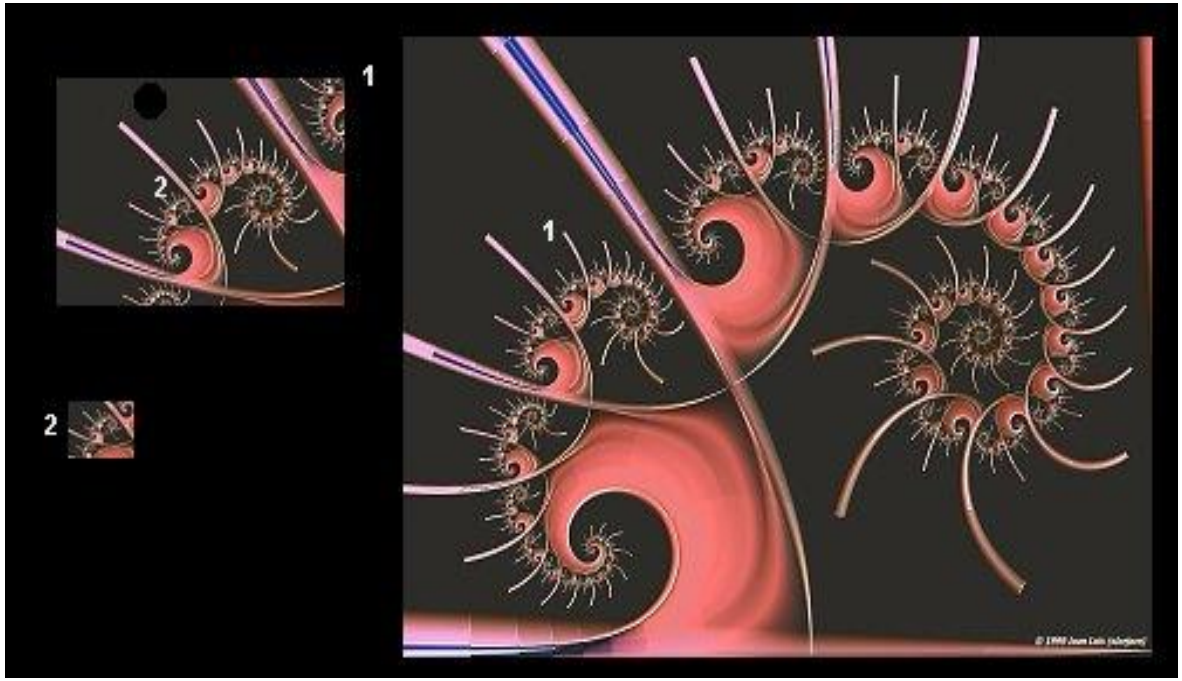
Random numbers





Complex representations

- Julia sets



On the nature

- Have the lungs Hausdorff 2½ dimension?
- https://en.wikipedia.org/wiki/List_of_fractals_by_Hausdorff_dimension

Random number generator

Using in science

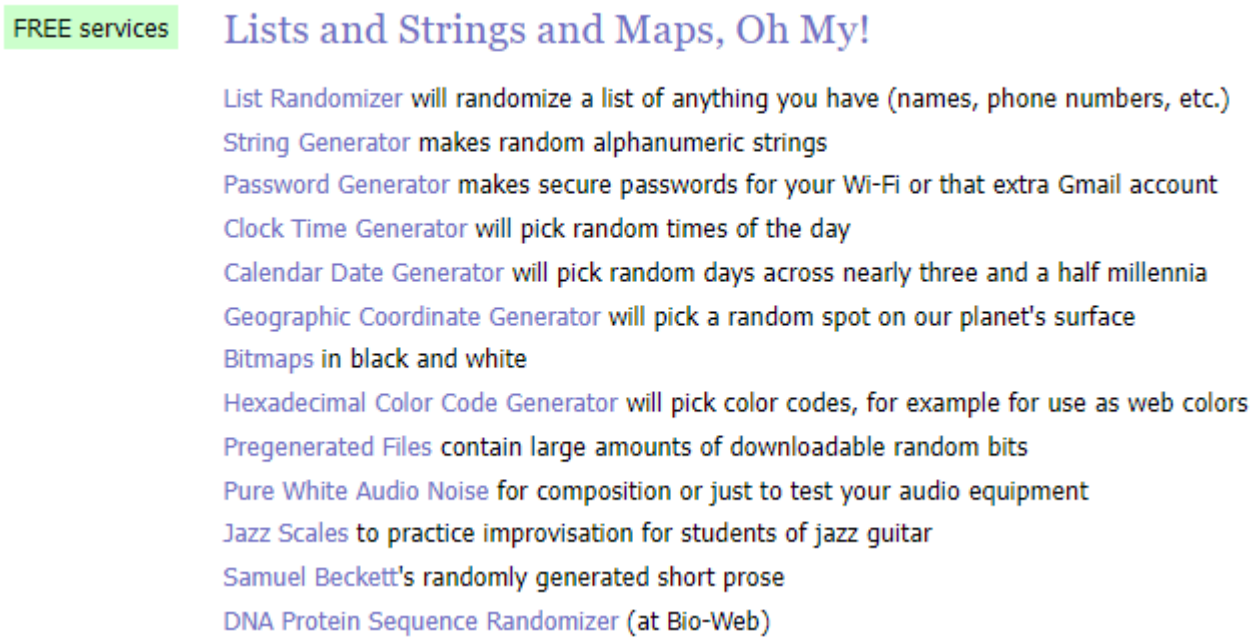
- Randomness are defined through the use of Random Number Generators (RNG)
- RNG are random variables generated following a uniform distribution U[0,1).

Pseudo-random numbers

- Random numbers generated by the GNA are not random, obey a known method that can be reproduced.
- Are pseudo-random numbers, not random.
- We can use them, although are dependent, thanks exists a set of test that assures that them seems random.

Random.org

- https://www.random.org/



RNG features

- Structural
  - Long period
  - Reticular structure coverage
  - Reproducibility
- Statistical aspects
  - Uniform distribution.
  - Density.
  - Statistical independence.
- Theoretical aspects
  - Complexity.
  - Stability of the process.
- Computational aspects
  - Is easy to program?
  - Memory requirements.
  - Efficiency

History of the RNG

- First families of the RNG are based on phisical or mechanical methods
  - Tables, lotery numbers, urn balls, gamma ray emissions, perforated disks, cosmic rays, thermal noise, radioactive phenomena

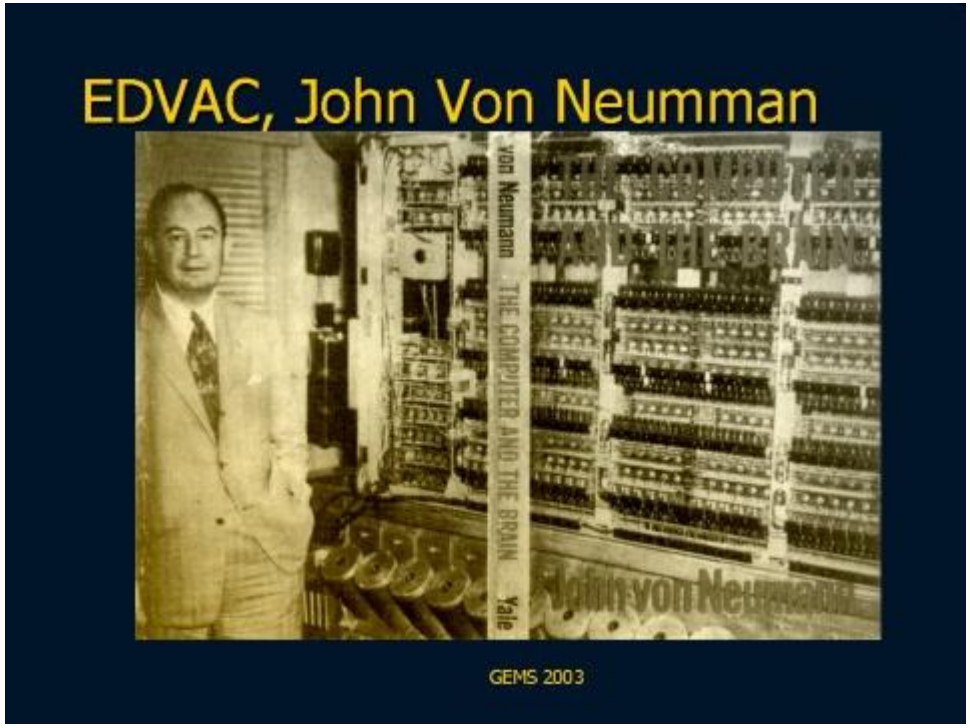
History

- *The first arithmetic generator was proposed by Von Neumann*
- *y Metropolis on 1940 : el "Middle square method"*

i	Zi	Ui	Zi2
0	2684	–	07203856
1	2038	0.2038	04153444
2	1534	0.1534	02353156
3	3531	0.3531	12446161
4	4631	0.4631	21446161
5	4461	0.4461	19900521
6	9005	0.9005	81090025
:	:	:	:

History

- *Von Neumann amb el EDVAC (Electronic Discrete Variable Automatic Computer) una de les primeres computadores electròniques. A diferencia del ENIAC, no era decimal, sinó binaria i va tenir el primer programa dissenyat per ser emmagatzemat. Aquest disseny es va convertir en estàndard d'arquitectura per la majoria dels ordinadors moderns.*



RNG classification

- Marsaglia y Zaman (1991), classification of the RNG:
  1. Congruential generators.
  2. Shift – Register generators.
  3. *Lagged* – Fibonacci generators.

Congruential generators

- Introduced by Lehmer on 1951
- A lot of the RNG are based on these.
- Defined with a recursive equation:
  - $Z_i = (a Z_{i-1} + c) \pmod m$
  - **m** is the module.
  - **a** is the multiplicative factor
  - **c** is the additive factor
  - **Zo** is the seed
- Obviously, $0 \leq Z \leq m-1$

Congruential generators

- To obtain the numbers that follows a uniform U[0,1) we must divide
  - Ui = Z i /m
- To avoid negativity, the enters m, c, a y Z must assure than:
  - 0 < m, a < m,
  - c < m, y Z 0 < m.

Congruential generators

- Full cycle congruential generator
- Since m=16 maximum different numbers expected are 16.
- Is convenient that the value of ***m be bigger***, big as is possible, as an example10^9.

i	*Ui*
0	-
1	0.375
2	0.063
3	0.500
4	0.688
5	0.625
6	0.313
7	0.750
8	0.938
9	0.875
10	0.563
11	0.000
12	0.188
13	0.125
14	0.813
15	0.250
16	0.438
17	0.375
18	0.063
19	0.500

$Z_0 = 7$

m = 16
a=5
c=3

Congruential generators

- **Cycles:** when a Zi takes as a value the same value than other Zj with j<i a cycle appears. The values will be repeated again and again.
- **Period:** the longitude of this cycle "i".
  - Since $0 \leq Z \leq m-1$, maximum value of the period is ***m-1***.
- **Full period:** when i=m

Congruential generators

- Parameters selection:
  - ***c*** and ***m*** must be prime between them.
  - If ***q*** is a prime number that divide ***m,*** then ***q*** must divide ***a-1***
  - If ***m*** is a multiple of 4, ***a-1*** is multiple of4 (***a*** cannot be multiple of 4).
- These conditions are assured if:
  - $m= 2^b$
  - $a=4k+1$
  - c odd (with a, c and k positive integers)
- With this we obtain a full period.

Congruential generators

- Congruential generators can be classified depending on the value of c variable:
  - si $c > 0$ GCL Mixt
  - si $c = 0$ GCL Multiplicative
  - si $c \neq 0$ y $a=1$ GCL Additive

LCG Mixt c>0

- To obtain long periods and high densities of U[0,1) is it desirable a high value of ***m***
- The division operation with ***m*** can be avoided explicitly.
  - A good value for ***m*** is 2^b , where b is the number of bits of the word of the computer (32 or 64 as usual).
  - The maximum integer we can obtain is $2^b - 1$.

LCG Mixt c>0

- This ***m*** allows to use the integer overflow in the computer avoiding the division by ***m***.
- *Example:*
- Supposing a computer of 4 bits (word) : $m = 2^4 = 16$.
  - If $Z_0 = 5$, $a = 5$ y $c = 3$
  - $(aZ_{i-1} + c) = (5*5)+3 = 28 = 11100$
- Since the long of the word if 4 bits we delete the first word in the left:
  - $1100 = 12 \therefore Z_1 = 12$

LCG Multiplicative c=0

- No full period

Schrage method

- Schrange method to avoid overflow.
  - $q = m / a$
  - $r = m \bmod a$
  - $k = n_i / q$
  - $n_{i+1} = a * (n_i - k * q) - r * k$

Other generators

- **More general concruentials**
- **Composite generators.**
- **Tausworthe generators.**
- **AWC/SWB**

More general congruential

- A LCG can be viewed as an special case of:
  - Zi = g(Zi-1, Zi-2, ...) (mod m)
  - Zi is between 0 and m-1. The Ui are obtained dividing Zi/m
  - **g** is a deterministic function depending on the previous value Zj.
- A generalization can be:
  - g(Zi-1, Zi-2, ...)=a'Z2
  - i-1+aZi-1,+ c (Gen. Quadratic)

RVG

Random variables generator

Inverse transform sampling

- Consider a random variable X with cumulative distribution $F_X(x)$. It is easy to show that the random variable:
  - $U = F_X(x)$
- Have a distribution U(0,1). Si $F_X(X)$ is strictly increasing, we can rewrite the equation in its equivalent
  - $X = F_X^{-1}(U)$
- This transformation is called the inverse transform. Presents a good method when the inverse of $F_X(x)$ is easy to calculate.

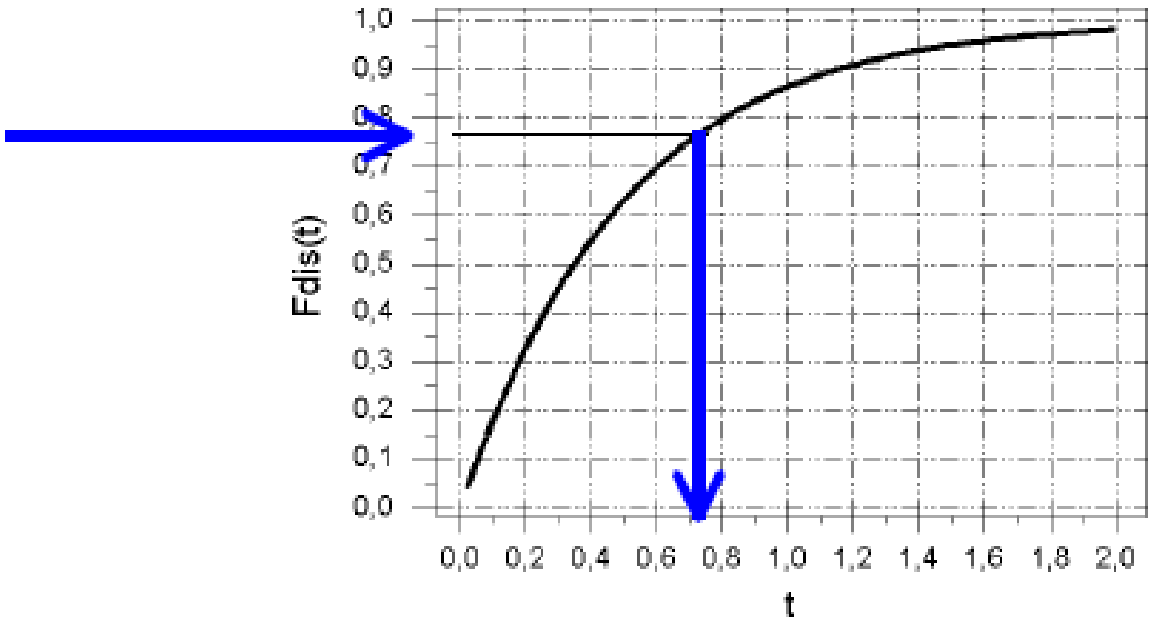
Inverse transform sampling (Example 1)

- Let X be a random variable that follows an exponential distribution with parameter λ. then:
  - $F_X(x) = 1 - exp(-\lambda x)$ if $x > 0$ and $F_X(x) = 0$ else.
- From here: $F_X^{-1}(u) = -\lambda^{-1} ln(1-u)$

Inverse transform sampling (Example 1)

- Inverse transform method for the exponential distribution.

$$r = 1 - e^{-\alpha \cdot x} \Rightarrow x = \frac{\ln(1-r)}{-\alpha} = \frac{\ln(r)}{-\alpha}$$



Examples de RVG

Uniform U(a,b)

- Inverse transform
  - $U \in [0,1)$
  - $X = a + (b - a)u$

Exponencial Exp(λ)

- Inverse transform.
- u∈[0,1)

$$x = \ln(1-u) \Big/ -\lambda$$

Normal N(μ,σ)

- x follows a normal N(μ,σ)
  - $x = \mu + \sigma \cdot (u_1 + \ldots + u_{12} - 6)$
- On u_k , $k = 1,\ldots,12$ are independent variables U(0,1).

Normal N(μ,σ): Box-Muller

- Box-Muller polar transformation:
- If u_1 i u_2 are independent random variables U(0,1) then
  - $x_1 = (-2*\ln(u_1))^{1/2}*\cos(2\pi*u_2)$
  - $x_2 = (-2*\ln(u_2))^{1/2}*\cos(2\pi*u_1)$
- *Then* x_1 *and* x_2 are random variables N(0,1) independents.

RNG tests

Assuring that seems random

RNG tests

- Determinist mechanism to obtain randon munbers (pseudo-random numbers)
- No exist a method to select the parameters (as an example parameters a, c and m of a LCG).
- Using a RNG we can guarantee that:
  - We obtain a long period.
  - The arithmetical efficiency of the method.

RNG tests

- Test are classified, according Law and Kelton (1999):
- ***Empirical***: Static tests.
- ***Theoretical***: No tests in the statistical sense. Using numerical parameters to test its quality.

Empirical tests

- These tests looks in a statistical way the obtained numbers to test how closer are for a U[0,1)
- Look to find the desired properties of a true random stream of numbers.
  - **Uniformity** of the values of the distribution.
  - **No correlation** in the sequence.

Empirical tests

- Uniformity can be guaranteed using an specific generator that obtains a “full period”.
- Tests process:
  - n.a.’s $\{u_i\}$ $0 \leq u_i \leq 1$ $i=0,1,2,\ldots$
  - integers $\{y_i\}$ $0 \leq y_i \leq d$ $i=0,1,2,\ldots$

Chi square test

- We divide the [0,1] in ***k equal intervals***, with (k>100, y n/k ≥5).
- If fj (j=1,2,...,k) is the amount of random numbers that lie in each interval j. If the numbers are correctly distributed, the frequency fits with the expected n/k.
- We calculate the Chi statistic.

$$\chi^2 = k/n \sum_{j=1}^{k} (f_j - n/k)^2$$

Chi^2 test example

- Values obtained with:
- $Z_i = 630{,}360{,}016\ Z_{i-1} \pmod{231-1}$
- $k=2^{12} = 4096$, $n=2^{15} = 32{,}768$
- Obtaining$\chi 2 = 4141$
- But

$$\chi^2_{4096,0.90} = 4211.4; \alpha = 0.10$$

- We can consider that the numbers follows a IID U(0,1)

Series test

- Is a Chi square generalization for big dimensions.
- We divide the stream in vector of size d.
  - ***d***: U1=(u1, u2, ..., ud) , U2=(ud+1, ud+2, ..., u2d), ...
  - We calculate the Chi square test with this expression:

$$\chi^2(d) = \frac{k^d}{n} \sum_{j_1=1}^{k} \sum_{j_2=1}^{k} \cdots \sum_{j_d=1}^{k} \left(f_{j_1 j_2 \cdots j_d} - \frac{n}{k^d} \right)^2$$

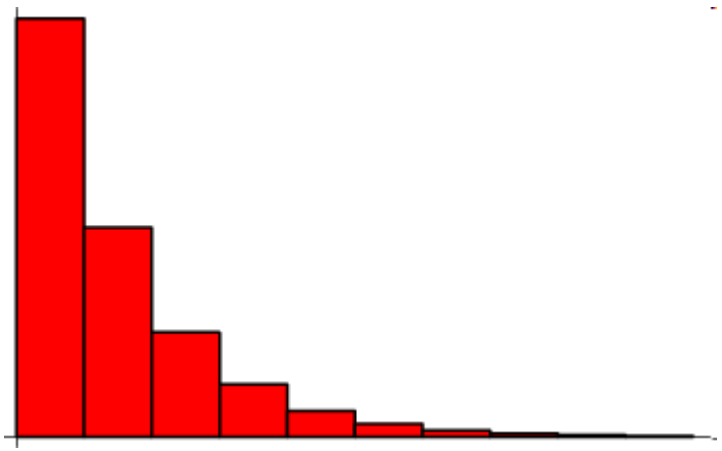
Series test example

- d=2, H_0: Pairs (ui,u2), (u3,u4), ..., (ud-1, ud) are IID U(0,1).
- We generate 32,768 pairs de Ui´s. K=64 ∴ g.l. = $64^2 - 1 = 4095$
- α=0.10 χ^2 =4211.4 y $\chi^2(2)$=4016.5
- ∴ We accept the uniformity of the generated numbers.

Gap test

- A gap is a sub-sequence of r consecutive values with defined distances between a and b:
  - $0 \leq a < b \leq 1$
- In examining a sequence of length n calculate the number of gaps of length 0,1,2,…,(t-1) y $r \geq t$
  - $P = prob(a \leq uk \leq b)$
  - $Pt = P(1-P)t$ is the probability of a long gap. t (if the sequence is uniform and independent). Follows a **geometric** distribution with parameter P {Ui(a, b)} = b - a
- We apply X2.

Geometric distribution



- Like the exponential is memory less
- Can be used to represent the number of failures before the first exit.

$$P(Y = n) = (1 - p)^n p$$

Gap test example

- We generate random numbers between (0,1)
- We note the distance between the different apparitions of numbers in the interval [a,b] of the sequence.
- As an example, if
  - [a,b] = [0.4,0.7]
  - The sequence is: 0.1, 0.5, 0.9, 0.6, ...,
  - The longitude for the first gap is 2 (between numbers 0.5 and 0.6)
  - We store the “n gaps” and we group all that have a biggest value in the category.

Run test

- To test independency, no uniformity.
- Involves examining the sequence of Ui's to detect segments of ascending numbers (run up) or descending numbers (run down).
- It is based on analyze a sequence of Ui's to detect ascendant segments of numbers (run up), o descendent segments of numbers (run down).
- Chi-square test is applied to the observed and expected frequencies.

Run test example

- u1,u2,..., u10= 0.86, 0.11, 0.23, 0.03, 0.13, 0.06, 0.55, 0.64, 0.87, 0.10

run up i	size	elements
1	1	0.86
2	2	0.11,0.23
3	2	0.03, 0.13
4	4	0.06,0.55,0.64,0.87
5	1	0.10

i	Executions of size i (r_i)
1	2
2	2
3	0
4	1
5	0
6	0

Run test example

- $r_1=2$, $r_2=2$, $r_3=0$, $r_4=1$, $r_5=0$, $r_6=0$
- For n>4000, R is closer to a Chi square distribution with 6 degrees of freedom
  - H_0: Ui's are IID random variables.
- For n<4000 next expression can be applied:

$$R = \frac{1}{n}\sum_{i=1}^{6}\sum_{j=1}^{6} a_{ij}(r_i - nb_i)(r_j - nb_j)$$

- Where a_{ij} is the (i,j) th element of the matrix that presents Knuth in his book "Handbook of Simulation" the same for b_i´s.

Permutation test

- For a positive integer that *t!* be small comparing this value with the longitude of the stream (Ui).
- We divide the Ui in blocs of size *t*, (*U*1,…, *Ut*), (*Ut*+1,…, *U*2*t*), ….
- On H_0 the *t!* possible arrangements in a bloc of long t must be uniform distributed.
- Again, a Chi square test is used to test its independence..

Theoretical tests

- This type of testing realizes a comprehensive analysis, examining the complete cycle. Therefore do not require the series of numbers, are required only the constants of the generator.
- The more well known tests are:
  - Spectral test.
  - Lattice test.
- They are based on Marsaglia discovering (in 1968), the random number generator falls mainly on planes.

Theoretical tests

- If u1, u2 ,... Is a sequence of random numbers generated by a LCG, the superposition of segments of size ***d*** of the series (u1,u2,...,ud), (u2,u3,...,ud+1), implies the generation of a relatively small number of planes in the ***d***–dimension [0,1)***d***.
- Spectral test returns the distance between the different hiper-planes.

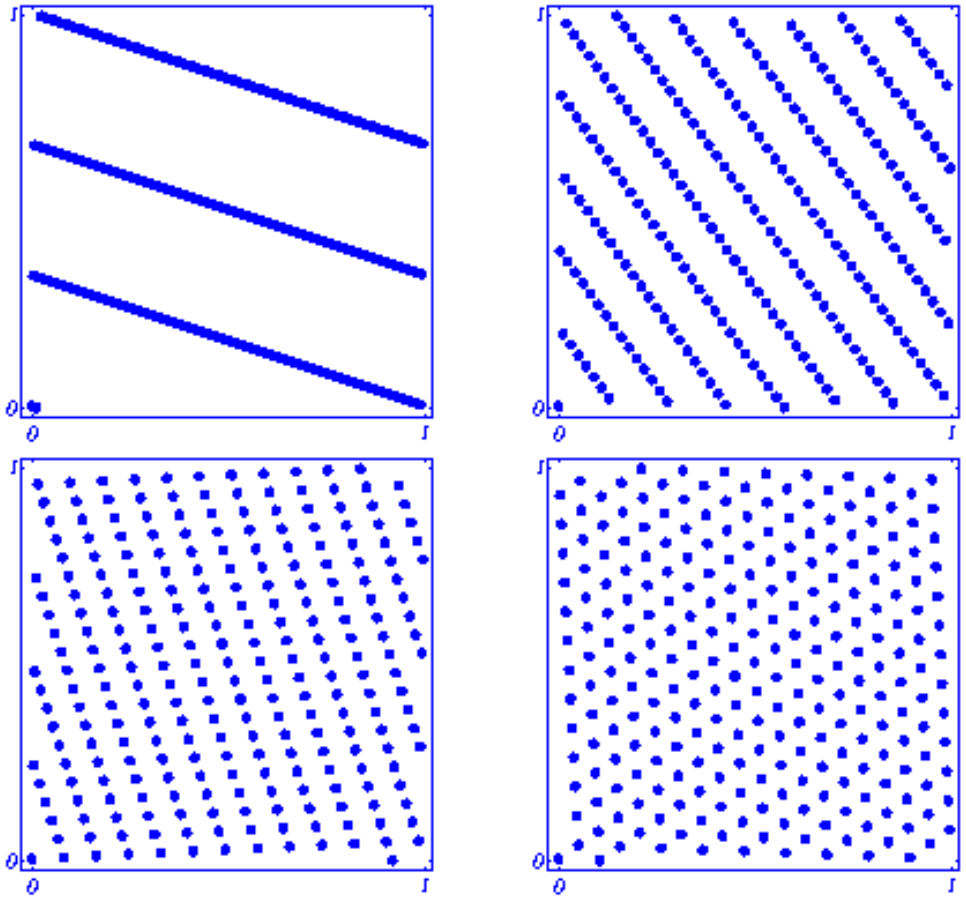
Examples (dimension 2)

- LCG(256,a,1,0) with a = 85, 101, 61, 237.
- Espectral test for 0.3162, 0.1162, 0.0790, 0.0632.
- Espectral test normalized of 0.1839, 0.5003, 0.7357, 0.9196.

- http://random.mat.sbg.ac.at/tests/theory/spectral/

Examples (dimension 2)

- LCG(256,a,1,0) with a = 85, 101, 61, 237.



Examples (dimension 2)

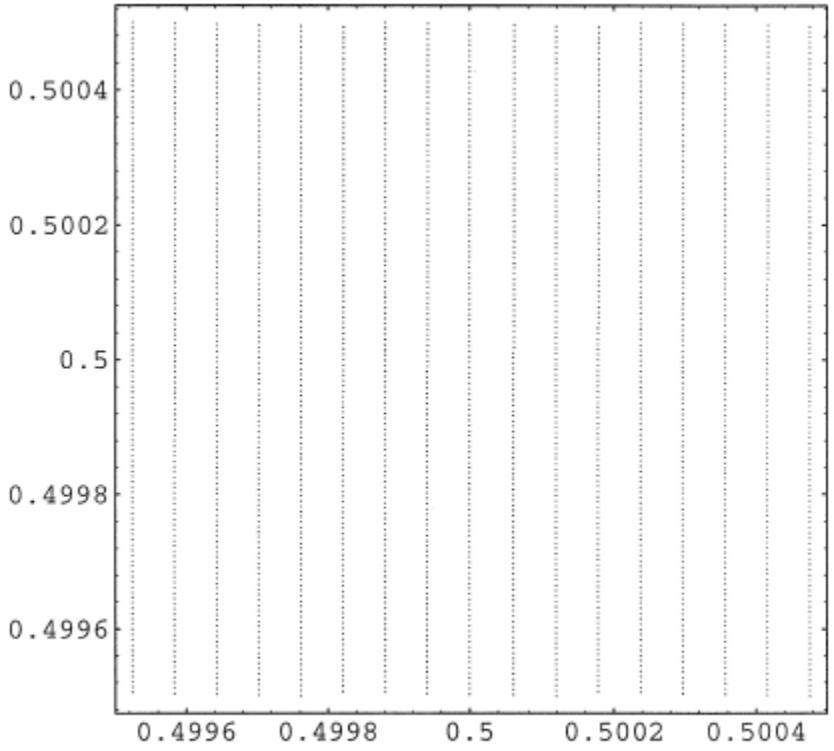


Fig. 3. Minimal Standard: LCG($2^{31}-1$,16807,0,1).

Examples (dimension 2)

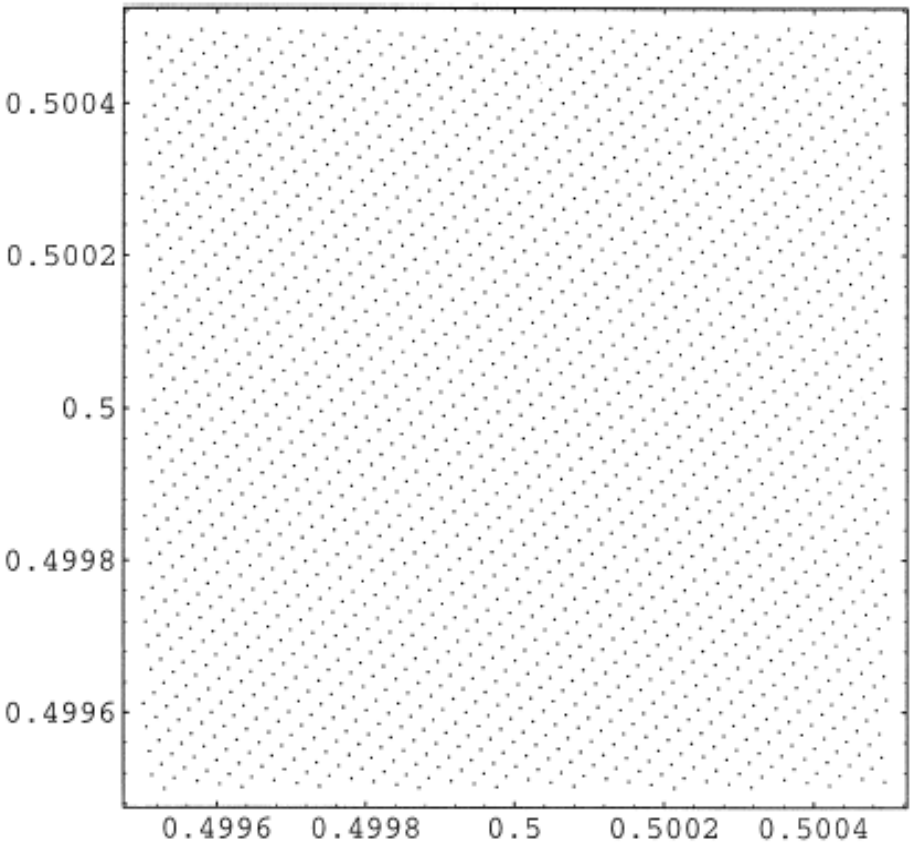


Fig. 4. SIMSCRIPT: LCG($2^{31}-1$,630360016,0,1).

LGC(2^{31},65539,0,1)

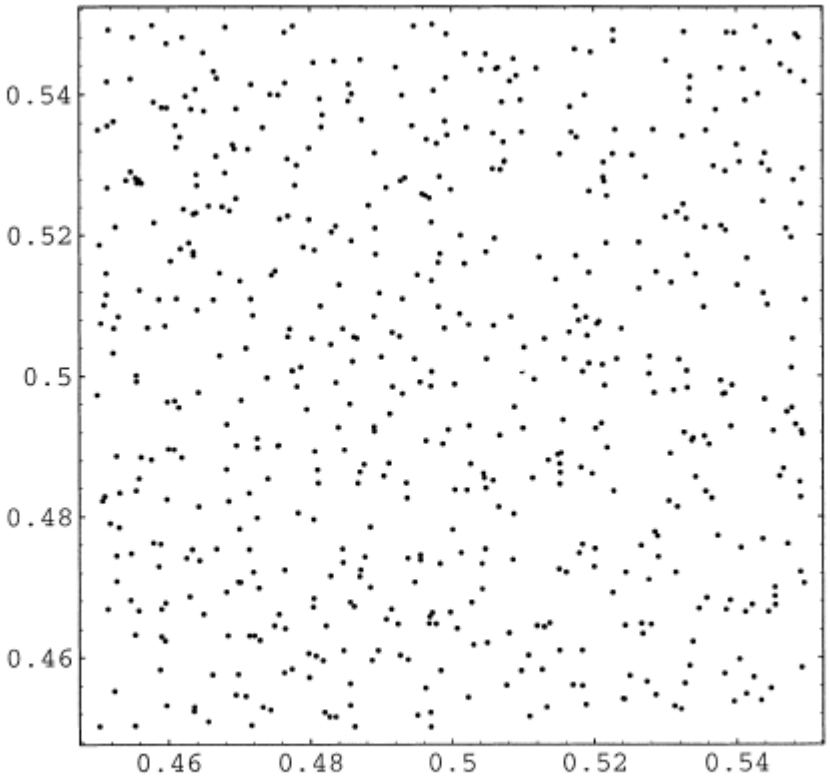


Fig. 1. LCG(2^{31}, 65539, 0, 1) Dimension 2: Zoom into the unit interval.

LGC(2^{31},65539,0,1)

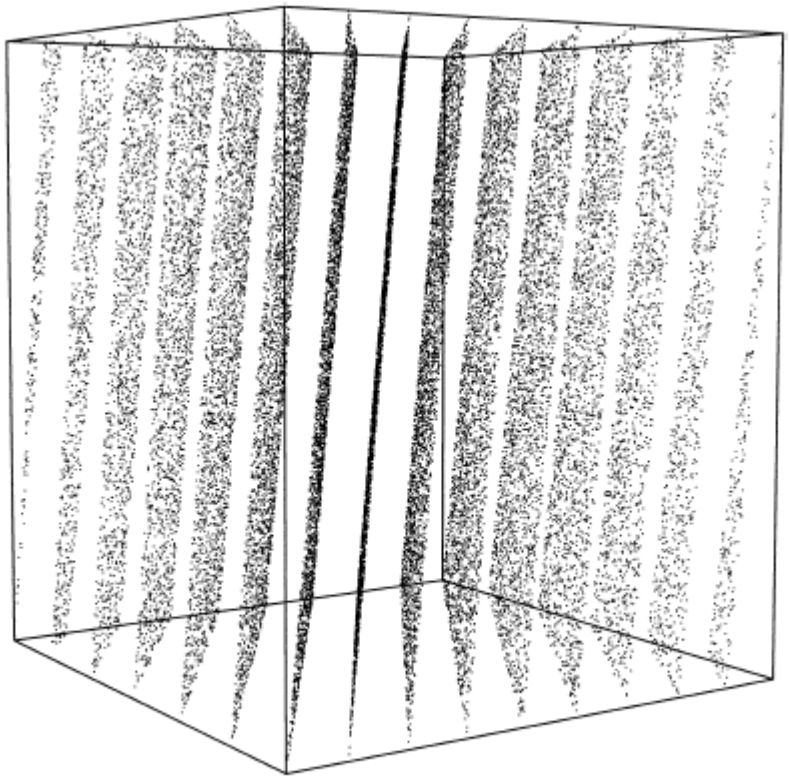


Fig. 2. LCG(2^{31}, 65539, 0, 1) Dimension 3: The 15 planes.

ICG (2^{31}-1,1288490188,1,0)

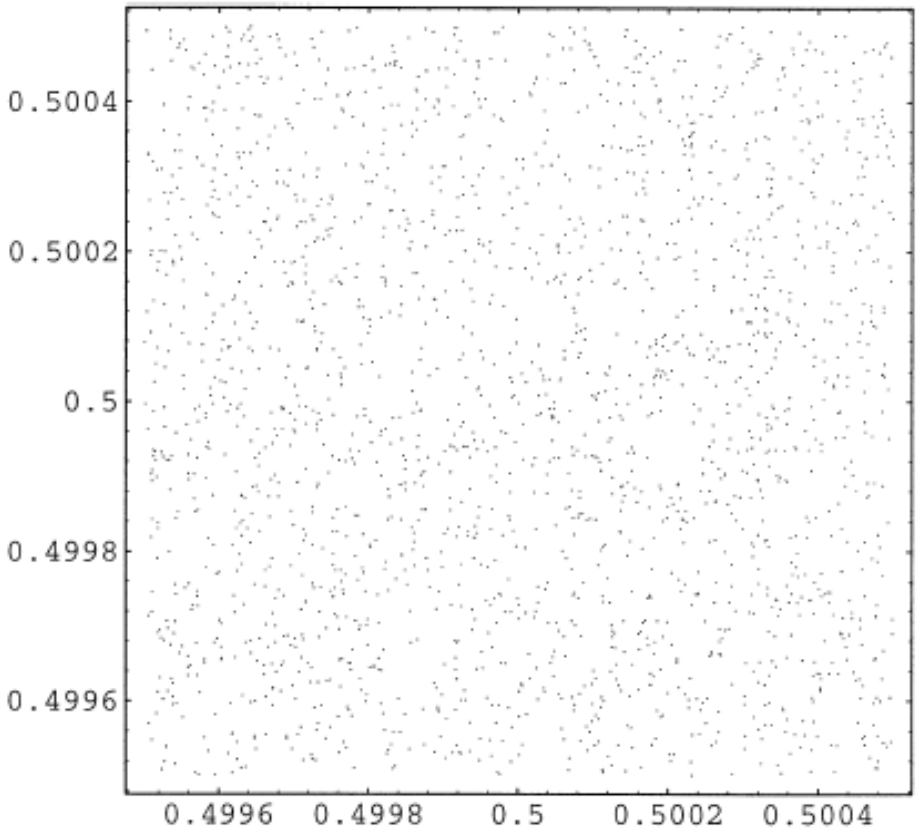
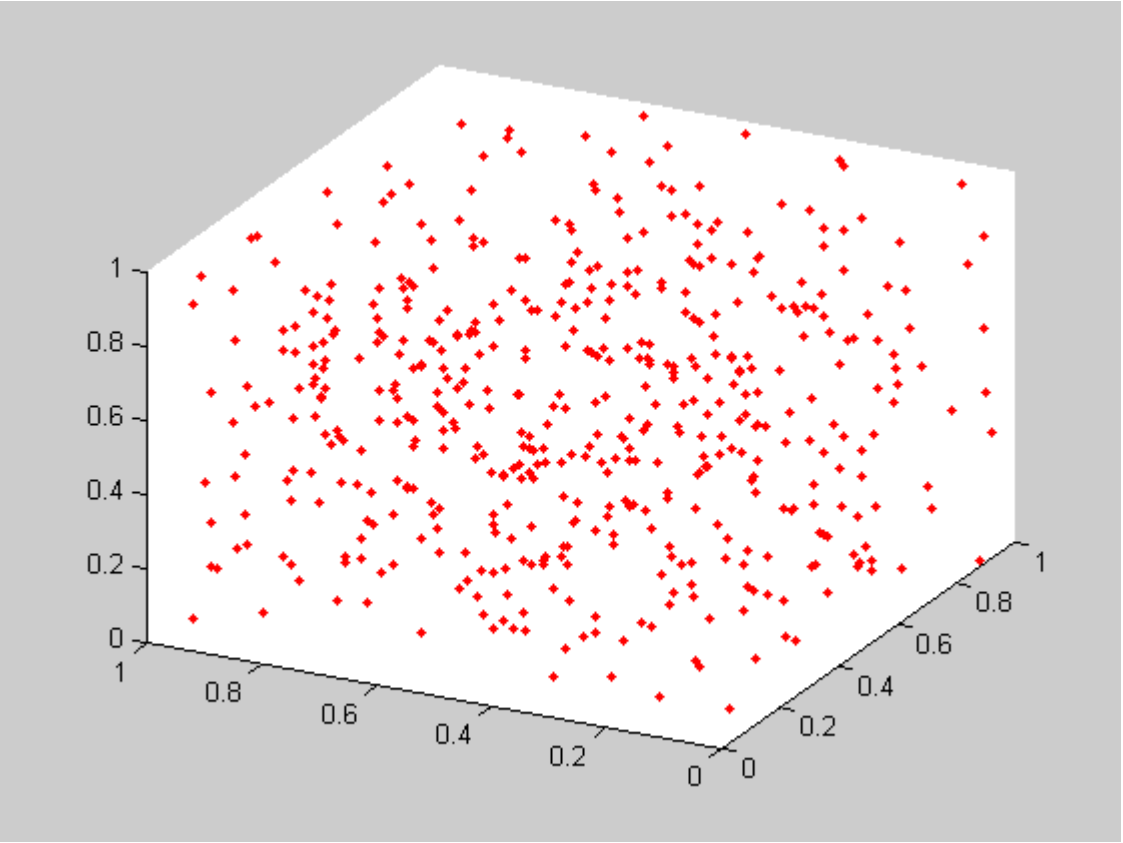


Fig. 5. All points of ICG($2^{31}-1$,1288490188,1,0).

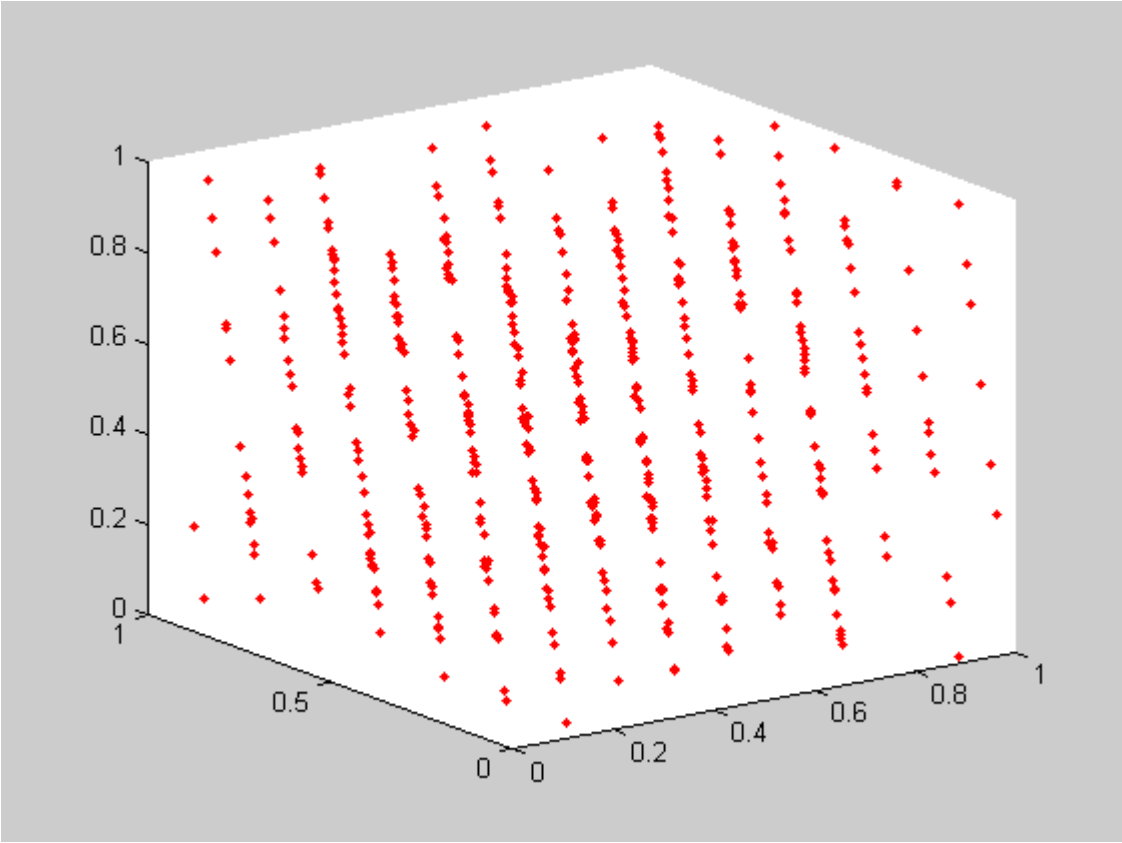
RANDU MatLab

http://www.stats.uwo.ca/computing/MatLab/randu.html

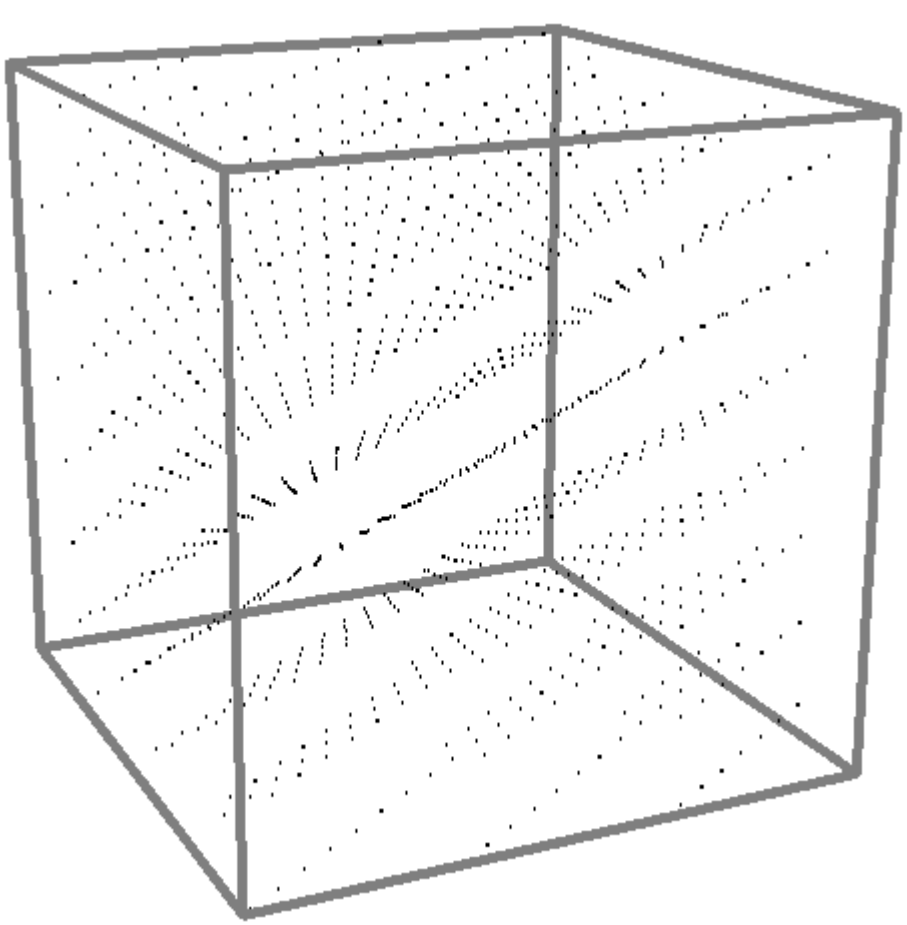


RANDU MatLab

http://www.stats.uwo.ca/computing/MatLab/randu.html

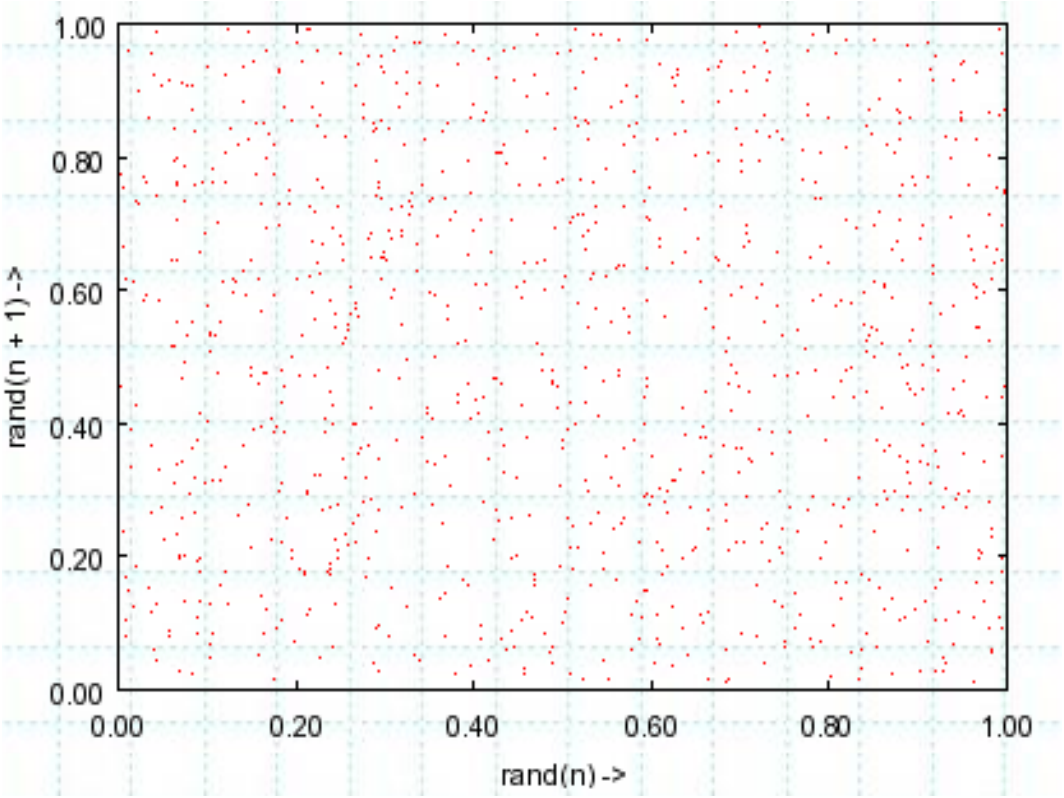


Examples (dimension 3)



Examples (dimension 2)

```
unset key
set xrange [0: 1]
set yrange [0: 1]
set zrange [0: 1]
set title "Plot del generador"
set xlabel "rand(n) ->"
set ylabel "rand(n + 1) ->"
set zlabel "rand(n + 2) ->"
set format x "%3.2f"
set format y "%3.2f"
set format z "%3.2f"
set tics
set sample 1000
set style function dots
set parametric
plot rand(0), rand(0)
```

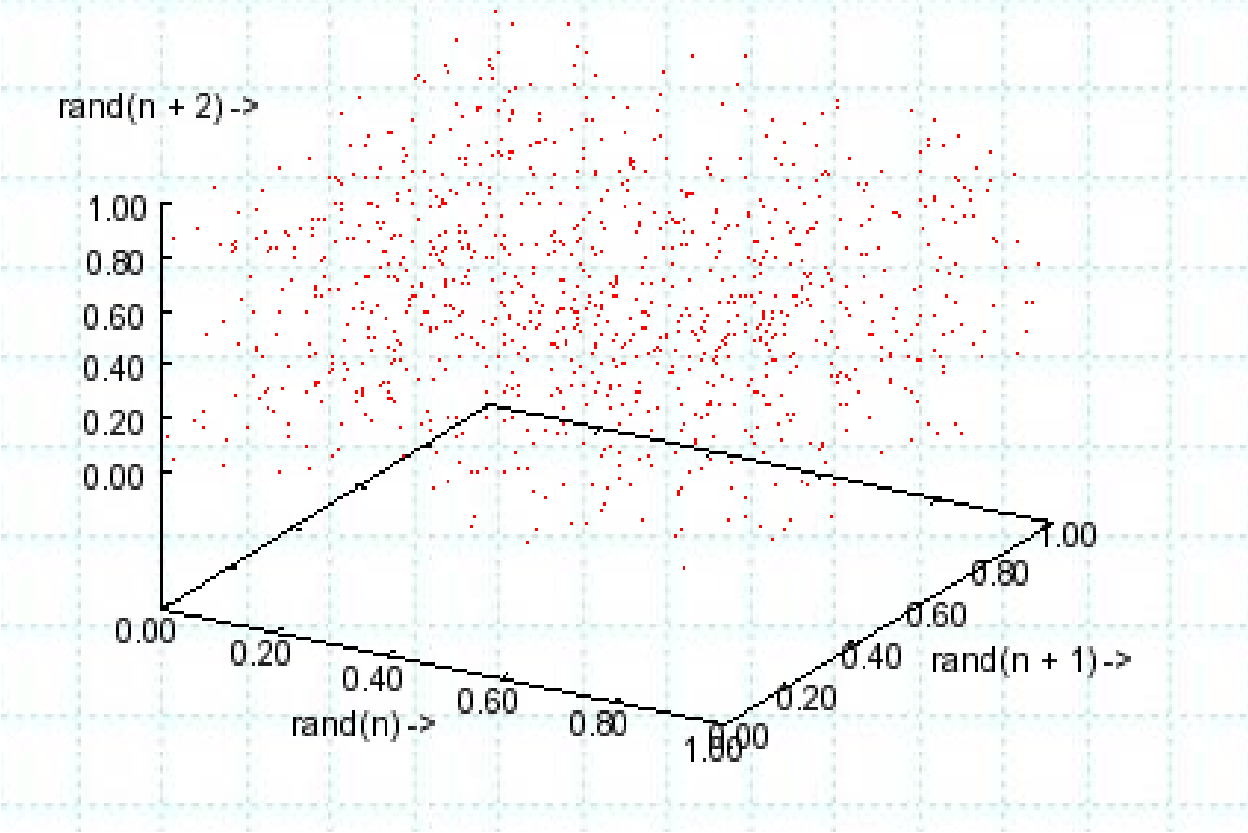


Examples (dimension 3)

print "3D plot ahead, one moment please ..."

set sample 50

splot rand(0), rand(0), rand(0)



Benford's Law

Benford's Law

- Benford's Law states that numbers from various sources, which we think should have random elements, are distributed anti-intuitively. The leading digits of the numbers are not distributed evenly, instead 30% begin with '1' falling off to 5% which begin with '9'. The first digit distribution for leading digits 1 through 9 is (30.1, 17.6, 12.5, 9.7, 7.9, 6.7, 5.8, 5.1, 4.6).

Examples

- One of the examples Benford himself noted was “sizes of river basins”. Is it mysterious that for every basin size that starts with ‘9’, there are 6 or 7 that start with ‘1’?
- More at: http://www.benfords-law.com/

$$P(d) = \log_{10}(d+1) - \log_{10}(d) = \log_{10}\left(\frac{d+1}{d}\right) = \log_{10}\left(1+\frac{1}{d}\right).$$

d	$P(d)$	Probabilidade de ser o primeiro dígito
1	30.1%	
2	17.6%	
3	12.5%	
4	9.7%	
5	7.9%	
6	6.7%	
7	5.8%	
8	5.1%	
9	4.6%	

Detection fraud in comptability

- Hal Varian – Chief economist, Google
- Mark Nigrini – Accuntability frauds.

Detection of fraud in COVID-19 cases reporting

- https://kevinbasset.medium.com/i-used-benfords-law-to-analyze-covid-19-in-113-countries-1a1194668069

Examples

- Divide in bins:
  - Go to random.org
  - Generate 100 random numbers.
  - Test those numbers (for each bin). What do you observe?
- Plot those numbers in diferent dimensions.
  - What do you obseve?

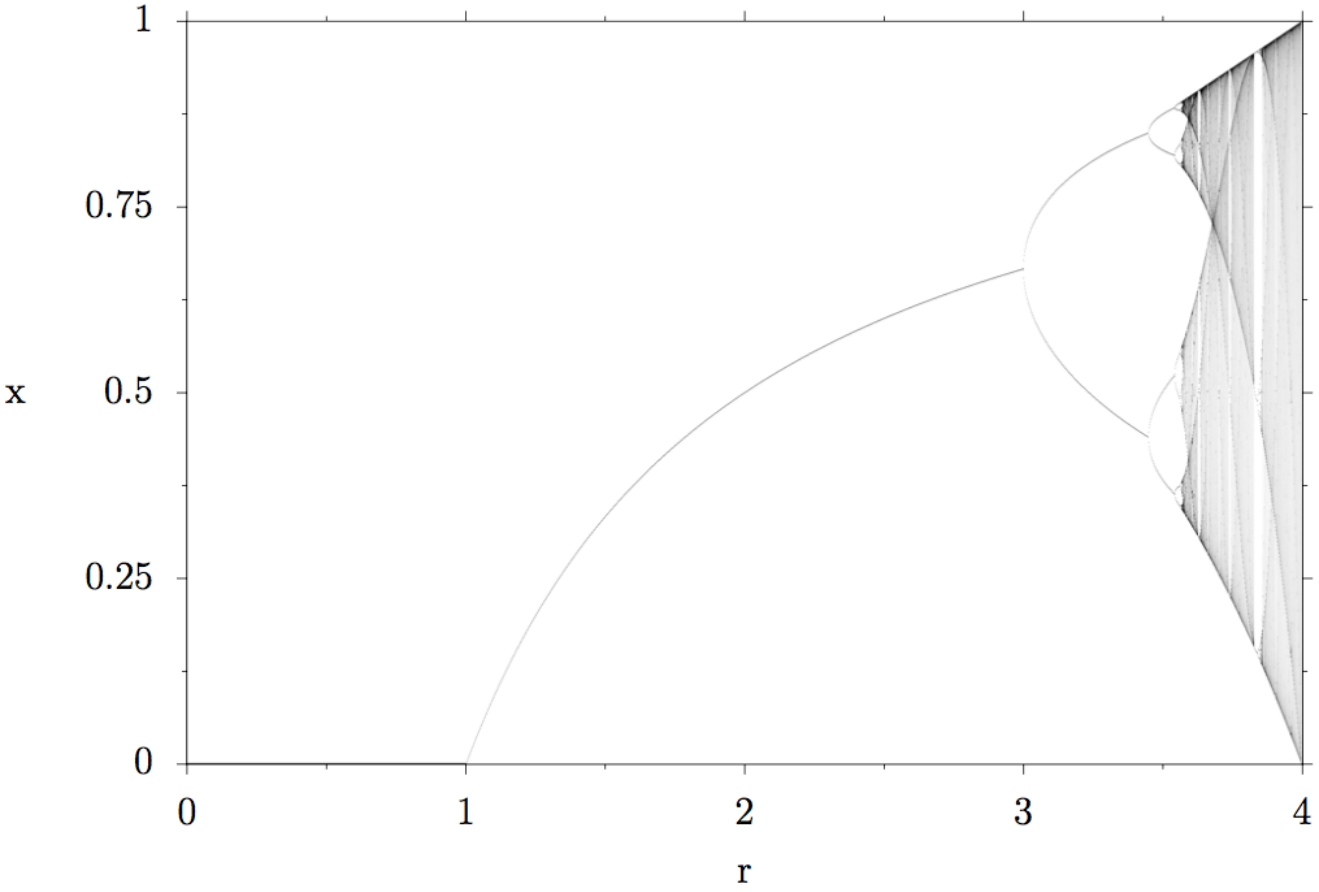
Simulism

- Read the article on:
- https://en.wikipedia.org/wiki/Simulated_reality
- Discuss about it.

Chaos

The stunning beauty of Chaos theory

Bifurcation diagram of the logistic map



$$x_{i+1} = r \cdot x_i (1 - x_i)$$

The logistic equation

- The logistic equation describes a demographic model with two counteracting processes that govern the size of the population: reproduction vs starvation due to a limited food supply.
- If there are no animals in the population x has a value of 0, and x=1 means that the population has reached its maximum size (due to limited food supply). r is the rate of reproduction.
- The index i means population at time i, index i+1 means population at the next time step. This means that if in this model we know the rate of reproduction r and the current size of the population x_i, we can calculate the population at the next time step x_i+1.