SIMULATING RANDOMNESS

MAR BENAVIDES I NEBOT · IES GELIDA, Gelida (BCN), Spain.

GOAL

Learning about random numbers and different methods to generate them. Developing a program to obtain pseudo-random numbers, prove their faithfulness and use them to perform experiments.

RANDOM NUMBERS (RN)

Sequence of numbers that satisfy: uncorrelation (no relationship among numbers), uniformity (fairness) and uniqueness (nobody can have information about it).

True RN ← METHODS OF GENERATION → Pseudo RN

True RN

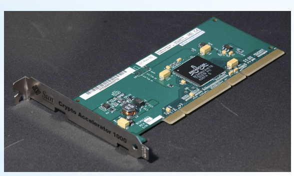
FEATURES

- ✧ Uniformity
- ✧ Uncorrelation
- ✧ Uniqueness
- ✧ Inefficient
- ✧ Nondeterministic
- ✧ Aperiodic

- ✧ Nuclear decay of radiation source
- ✧ Quantum vacuum fluctuations
- ✧ Thermal noise

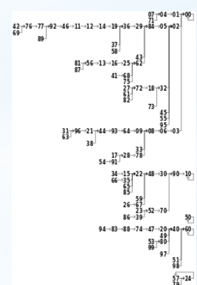
APPLICATIONS

- ✧ Lottery
- ✧ Raffles
- ✧ Gambling
- ✧ Security (cryptography)



Pseudo RN

- ✧ Computational algorithms:
  - Linear congruential generators (LCG)
  - Middle-square method



FEATURES

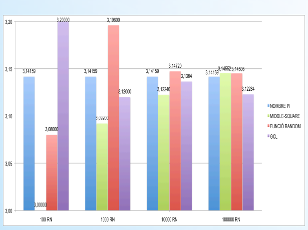
- ✧ Uniformity
- ✧ Correlation
- ✧ Nonuniqueness
- ✧ Efficient
- ✧ Deterministic
- ✧ Periodic

APPLICATIONS

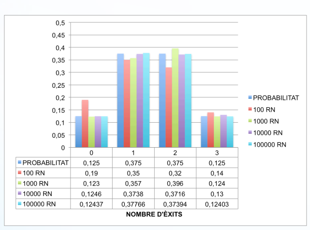
- ✧ Simulations
- ✧ Statistics
- ✧ Random sampling

EXPERIMENTS AND TEST

Approximation of Pi by Monte Carlo method



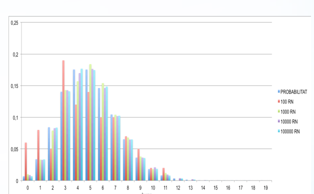
Simulation of a binomial distribution



	0	1	2	3
PROBABILITAT	0,125	0,375	0,375	0,125
100 RN	0,19	0,35	0,32	0,14
1000 RN	0,123	0,357	0,396	0,124
10000 RN	0,1246	0,3738	0,3716	0,13
100000 RN	0,12437	0,37766	0,37394	0,12403

NOMBRE D'ÈXITS

Simulation of a Poisson distribution

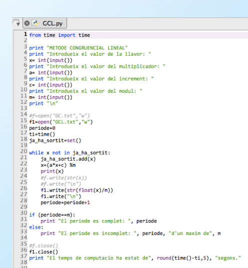


Chi-squared test

(analyse the randomness and frequency of the generated random numbers)

VALOR INFERIOR DE LA REGIÓ CRÍTICA	**0,484**
100 RN	4,9
1000 RN	1,02
10000 RN	3,067
100000 RN	3,9556
VALOR SUPERIOR DE LA REGIÓ CRÍTICA	**11,14**

All pseudo random numbers used in this work were generated by LCG and middle-square method




```
from time import time

print "METODE CONGRUENCIAL LINEAL"
print "Introdueix el valor de la llavor: "
x= int(input())
print "Introdueix el valor del multiplicador: "
a= int(input())
print "Introdueix el valor del increment: "
c= int(input())
print "Introdueix el valor del modul: "
m= int(input())
print "\n"

#f=open("GC.txt","w")
f1=open("GCL.txt","w")
periode=0
ti=time()
ja_ha_sortit=set()

while x not in ja_ha_sortit:
    ja_ha_sortit.add(x)
    x=(a*x+c) %m
    print(x)
    #f.write(str(x))
    #f.write("\n")
    f1.write(str(float(x)/m))
    f1.write("\n")
    periode=periode+1

if (periode==m):
    print "El periode es complet: ", periode
else:
    print "El periode es incomplet: ", periode, "d'un maxim de", m

#f.close()
f1.close()
print "El temps de computacio ha estat de", round(time()-ti,5), "segons."
```

CONCLUSIONS

Random numbers have a wide variety of applications. Depending on the use of the random numbers it is better to generate them by TRNG or PRNG.