ALEATORITZACIÓ

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

L'atzar

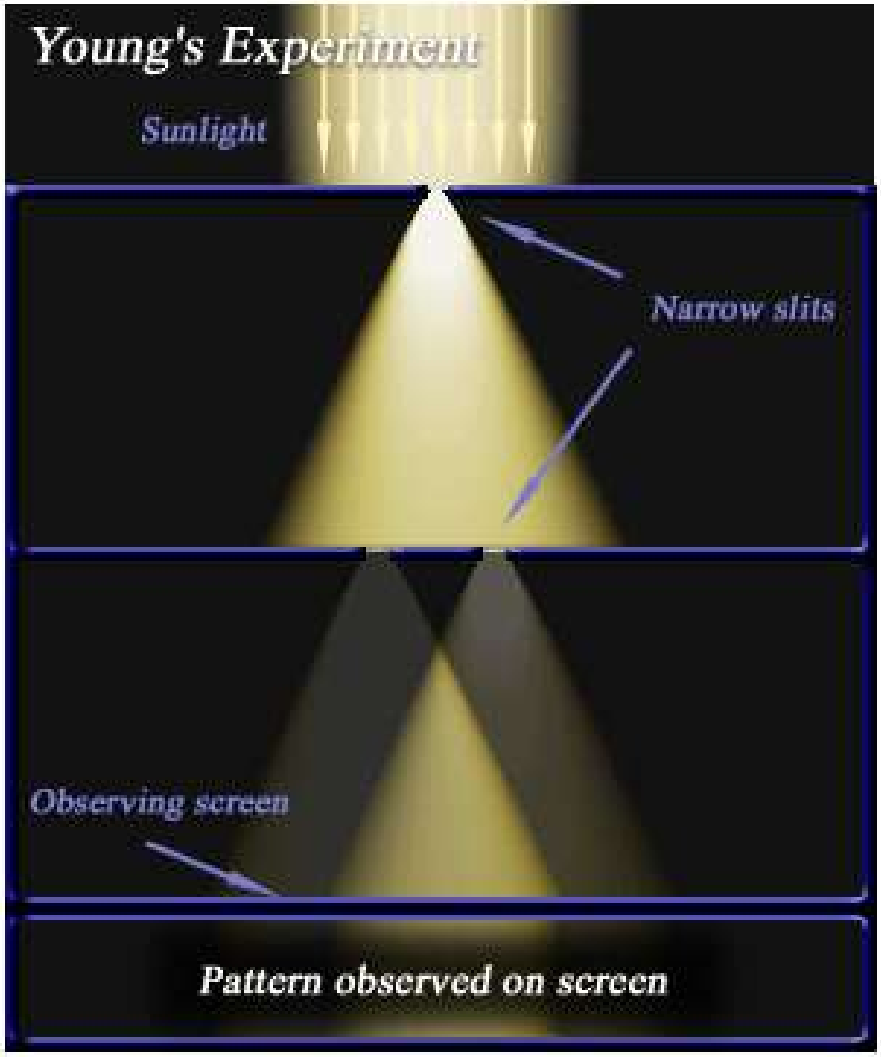
Entendre el que no podem entendre.

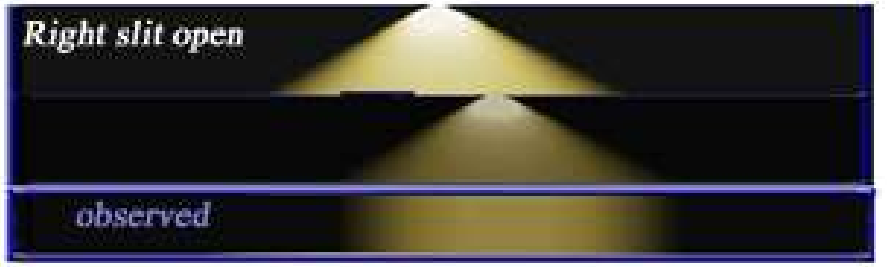
Sistema indeterminista

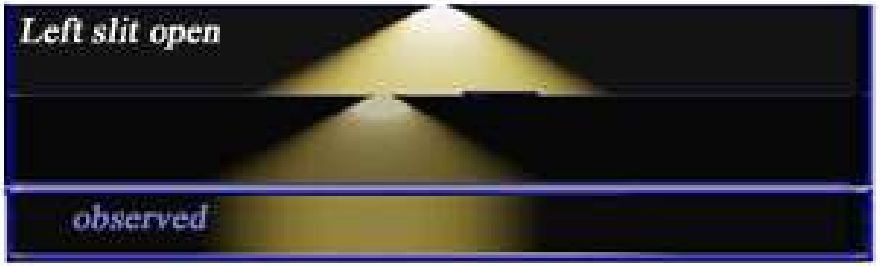
- No es pot predir a priori el comportament del sistema.
  - Sistema que depèn de factors que no es controlen agrupats en lo que s'anomena atzar.

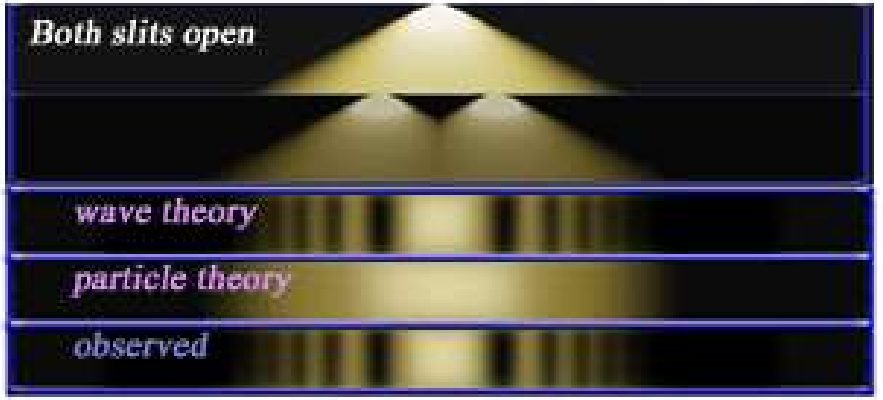
Mon determinista o indeterminista?
Deu no juga als daus amb l'univers.

L'experiment d'en Thomas Young









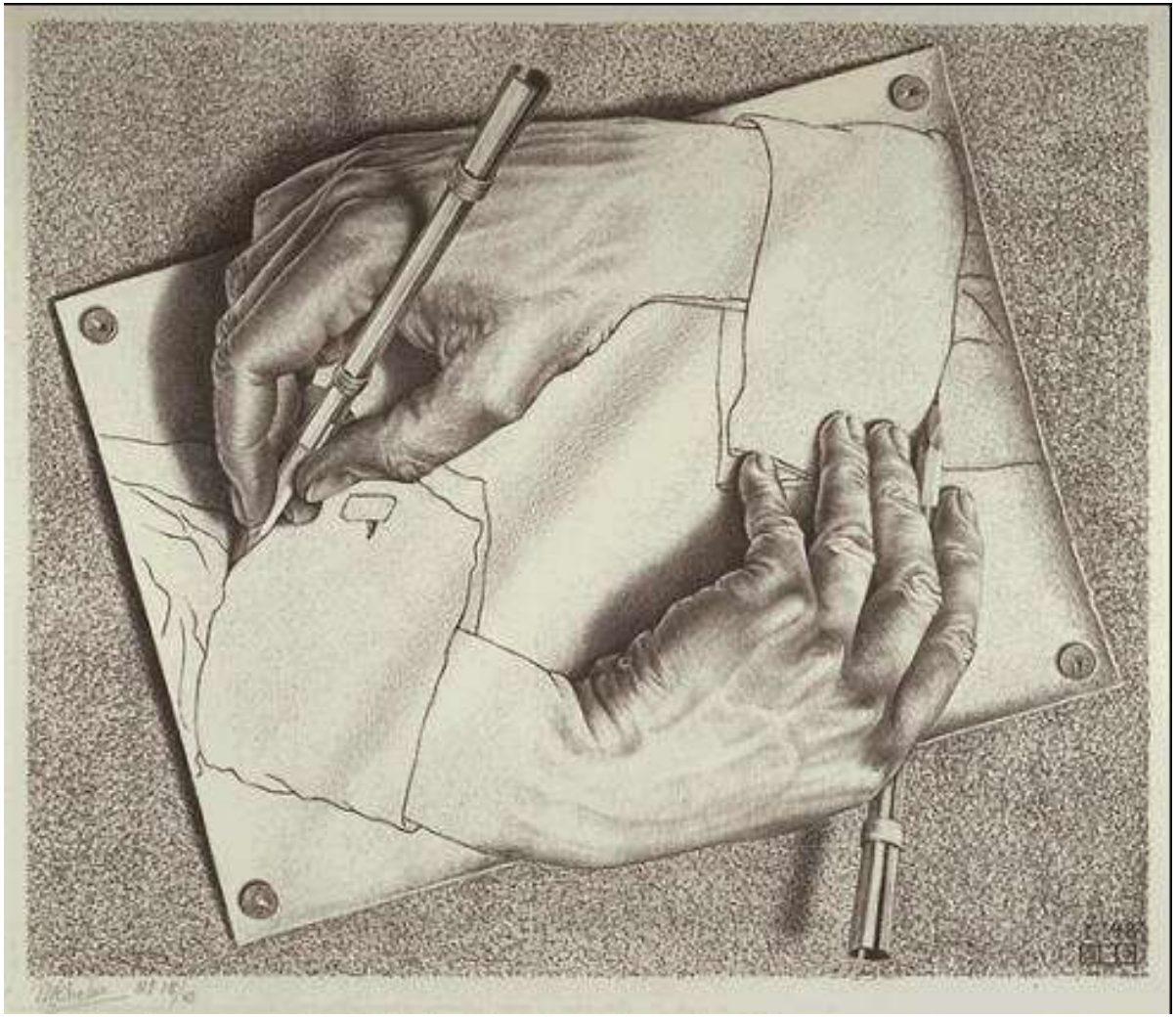
El gat de Schrödinger

- El gat es viu o mort?

- La curiositat “pot” matar al gat.

Interpretació de Copenhaguen

- Escher, Drawing Hands 1948



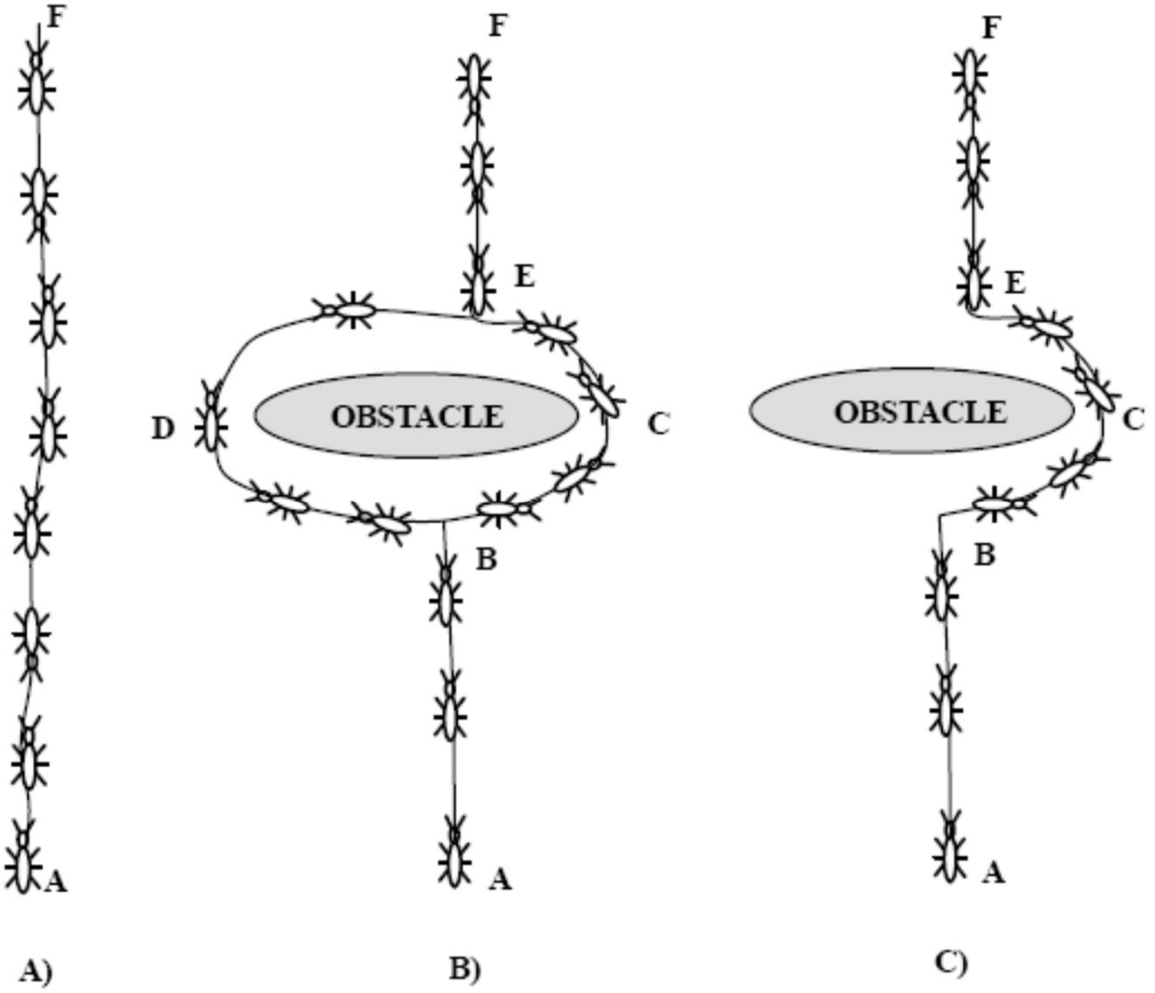
En qualsevol cas...

- Existeixen sistemes indeterministes. Com a mínim per nosaltres.

Exemples de sistemes indeterministes

- **Fila de formigues** que travessa un cartró amb dues entrades iguals.
  - Si la quantitat de formigues es suficient, acabaran passant únicament per un dels dos forats.
  - No es pot determinar a priori quin dels dos forats serà el triat.

Fila de formigues



Mes sobre formigues

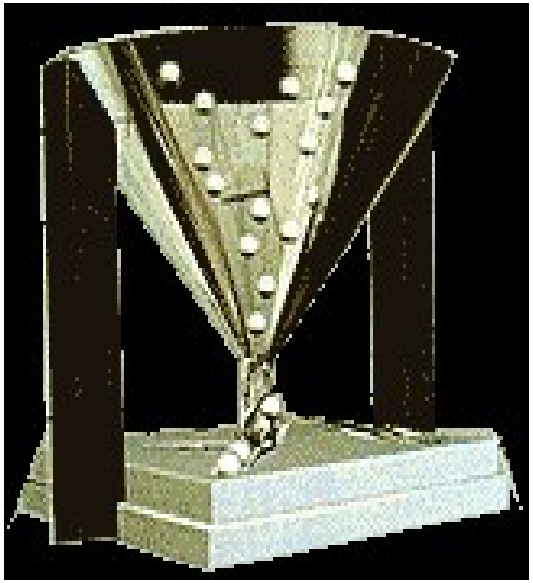
- In contrast to the top-down organization that characterizes many human endeavors, many social species achieve their communal goals using a purely bottom-up approach with no central command-and-control structure. A swarm of termites, for example, exhibits a collective intelligence that far exceeds the intelligence of any individual insect, which by itself has limited capabilities for processing and communicating information.

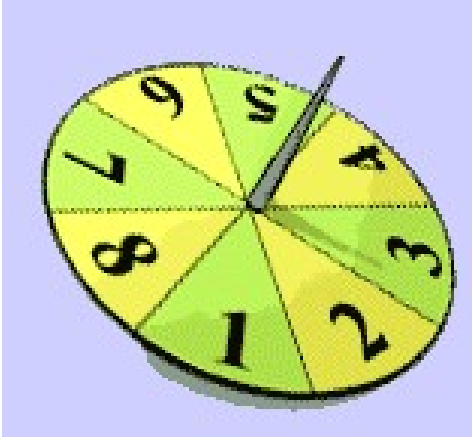
Termites

GNA

Generació de nombres aleatoris

Nombres aleatoris





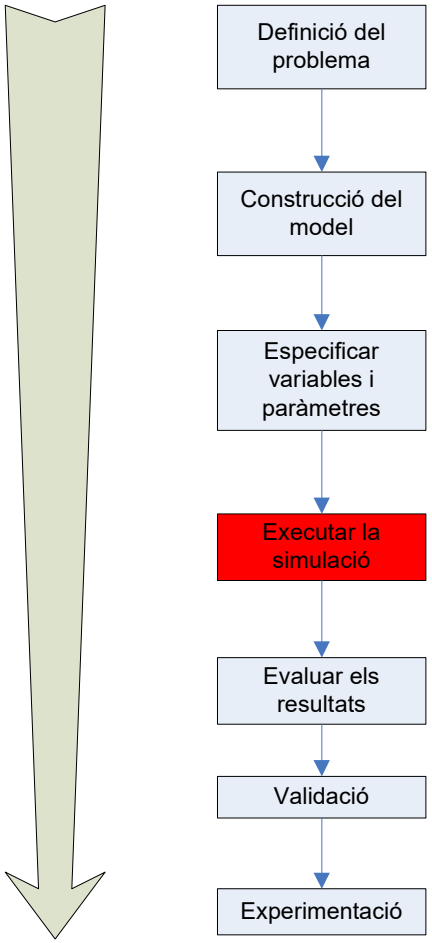
I Ching (sistema predictiu)

- The text of the I Ching is a set of predictions represented by a set of 64 abstract line arrangements called hexagrams (卦 guà). Although just the numbers 1 to 64 could have been used, the ancient Chinese instead used a figure composed of six stacked horizontal lines (爻 yáo). Each line is either Yang (an unbroken, or solid line), or Yin (broken, an open line with a gap in the centre). With six such lines stacked from bottom to top there are 26 or 64 possible combinations, and thus 64 hexagrams represented.

Bagua

GNA en la simulació

- Definir els objectes i les variables del sistema.
- Especificar els paràmetres, regles de decisió, distribucions de probabilitat.
- Determinació de les condicions inicials.
- Generació dels GNA/GVA.
- Proves estadístiques.
- DOE.



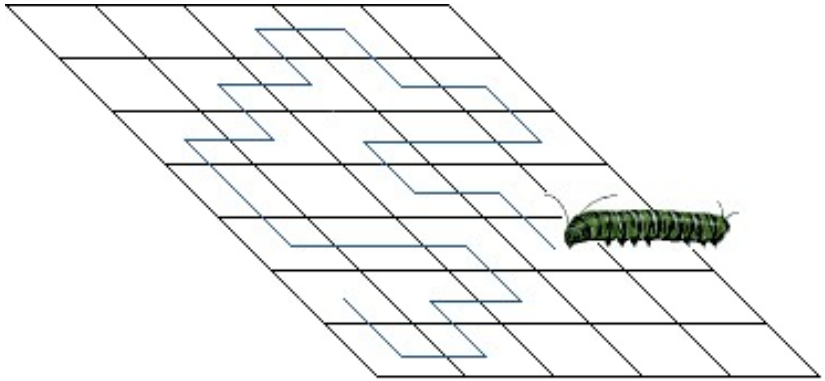
Els GNA en la ciència

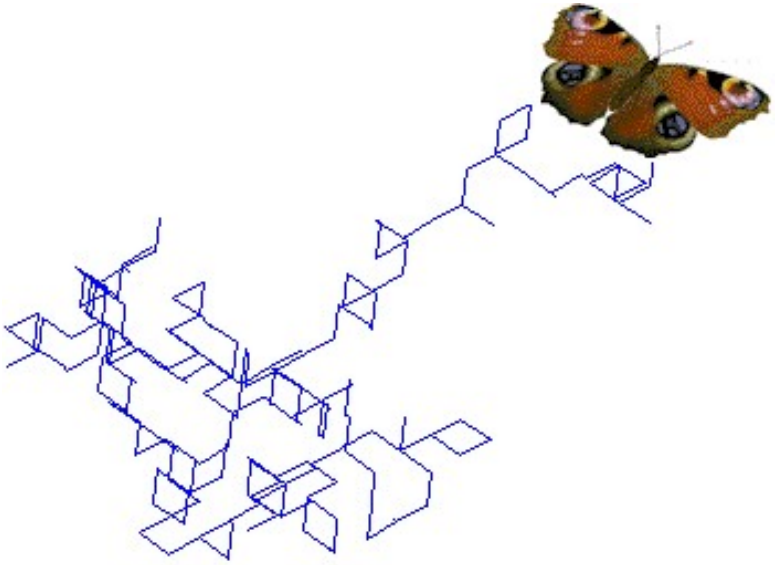
- Els GNA s'usen en moltes àrees a banda de la simulació:
  - Mètodes de Montecarlo.
  - Mètodes de simulació.
  - Estadística computacional.
  - Implementació d'algoritmes probabilístics.
  - Computació científica amb variables estocàstiques.
  - Probes VLSI Very-large-scale integration, el procés de crear circuits integrats a partir de combinar milers de circuits basats en transistors en un únic chip.
  - Criptografia.
  - Jocs
  - I-Ching

Els GNA i els Jocs

- WoW, LineageII, Age of Conan.
- SimCity, Sims.
- ...

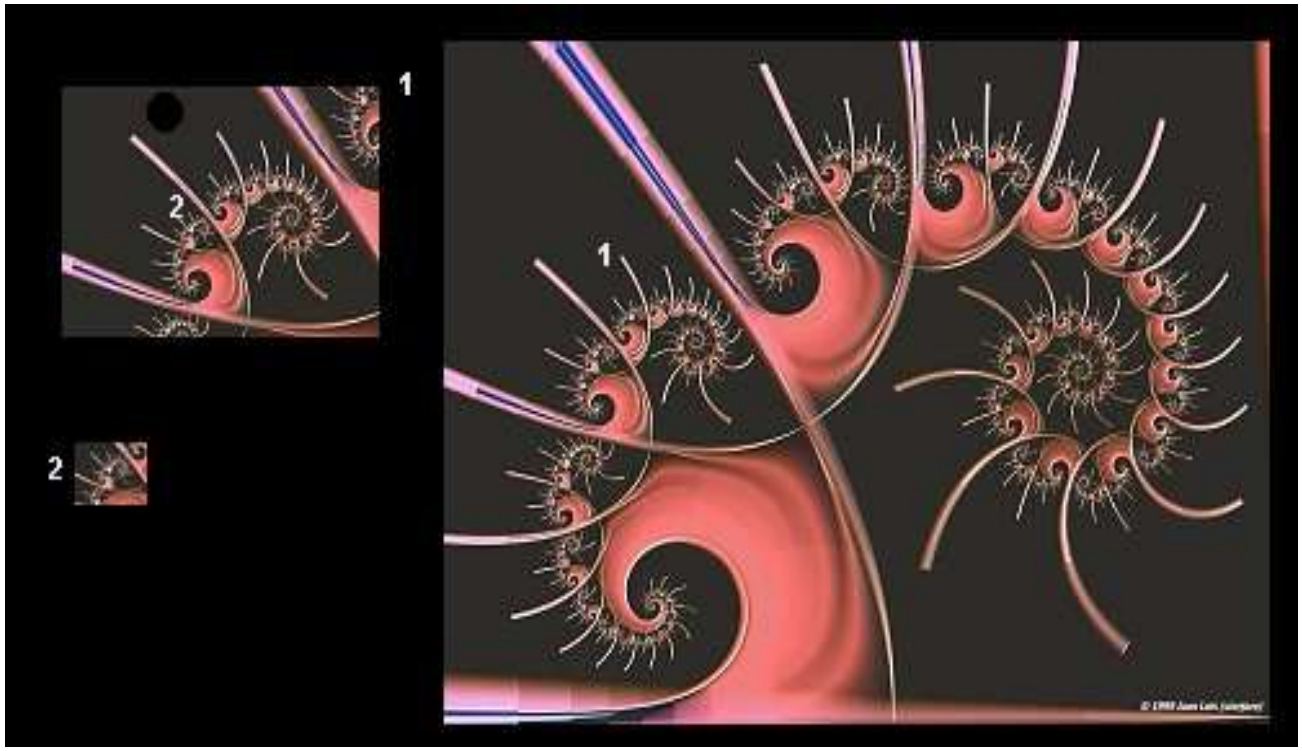
Representació de moviment





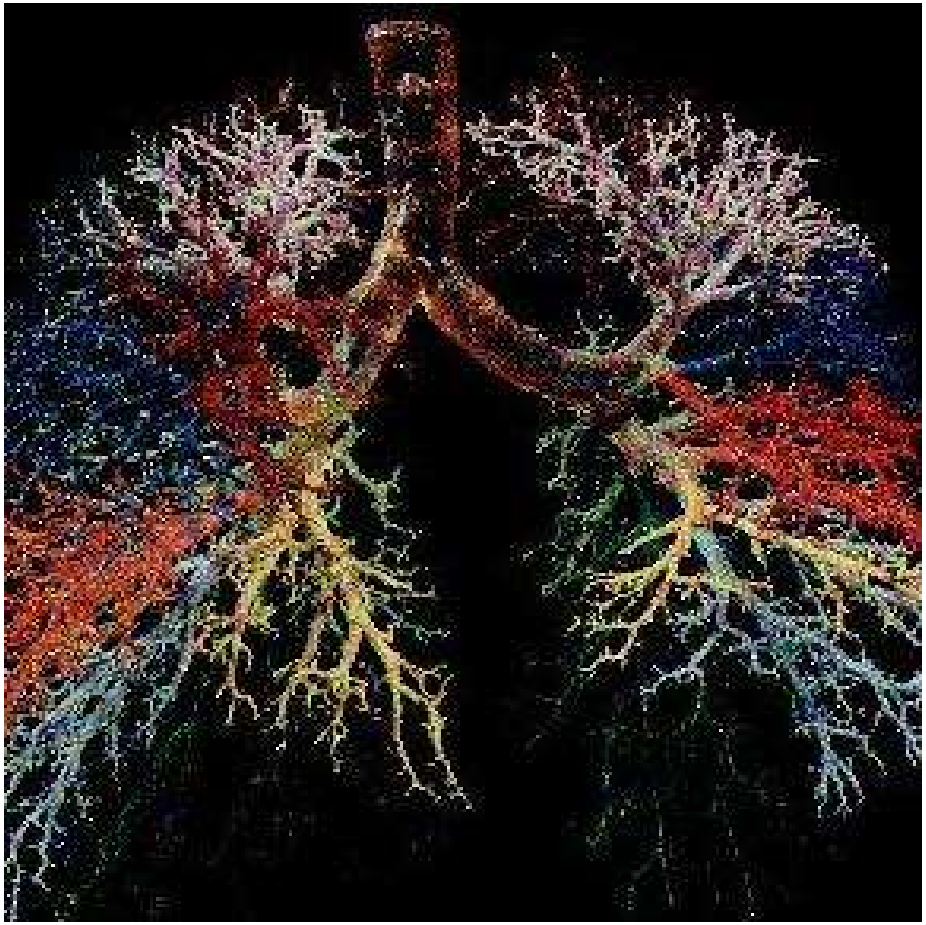
Representacions complexes

- Julia



La natura

- Tenen els pulmons dimensió de Hausdorff 2½ ?



Random number generator

Ús en un simulador

- Els nombres aleatoris constitueixen un component fonamental d'un simulador.
- L'aleatorietat vindrà introduïda pels generadors de nombres aleatoris (GNA)
- Els GNA son variables aleatòries generades segons una distribució uniforme U[0,1).

Nombres pseudo-aleatoris

- Els nombres generats pels GNA no son aleatoris, donat que procedeixen d'un mètode reproduïble i conegut
- Son nombres pseudo-aleatoris, NO ALEATORIS.
- Es poden usar doncs encara que son dependent entre ells un conjunt de tests permeten assegurar que semblen aleatoris.

Característiques dels GNA

- Estructurals
  - Llarg període
  - Cobertura de l'estructura reticular
  - Reproductibilitat
- Aspectes estadístics
  - Uniformitat de la distribució.
  - Densitat.
  - Independència estadística.
- Aspectes teòrics
  - Complexitat.
  - Estabilitat del procés.
- Aspectes computacionals
  - Facilitat de programació.
  - Requeriments de memòria.
  - Eficiència.

Estructurals

- **Llarg període**: Interessa GNA amb un llarg període.
  - **Període**: nombre màxim de valors diferents que el GNA es capaç de generar a partir d'uns paràmetres determinats. (*streams*)
  - Els GNA amb un període més gran son els *Mersenne Twister* de Matsumoto y Kurita [MATS94], produeixen sèries de l'ordre de 2^{19937}-1, un nombre de l'ordre de 6000 dígits decimals (algoritme MT19937B en http://www.math.keio.ac.jp/~matumoto).

Estructurals

- **Cobertura de l'estructura reticular**: possible cobertura de l'estructura reticular incompleta o anòmala.
  - Es basa en el treball de Marsaglia [MARS68] en el que s'identificava la Propietat de les k-tuples de nombres aleatoris generats mitjançant determinats mètodes a distribuir-se en hiperplans i com recobreixen aquests una estructura reticular.
  - Una mala distribució d'aquests de forma que la distància entre hiperplans sigui excessiva, detectada mitjançant el test espectral [HELL98], pot arribar a representar un seriós inconvenient per problemes de grans dimensions en el que s'usin números aleatoris empaquetats en k-tuples amb k>3 (verue K. Entacher, a http://random.mat.sbg.ac.at/~charly/server/server.html).

Estructurals

- **Reproductibilitat**: ser capaços de reproduir de forma fàcil i sistemàtica qualsevol sèrie de nombres pseudo-aleatoris. Usant GNA.
  - Al nombre base de la sèrie s'anomena llavor.
  - Aquesta característica és fonamental en la fase d'experimentació, anàlisi de resultats i tècniques de reducció de la variança.

Aspectes estadístics

- **Uniformitat de la distribució:** Els nombres aleatoris han de distribuir-se de forma uniforme a lo llarg de tot el rang de valors possibles que pugui prendre la sèrie, de forma que no apareguin trams no recoberts ni agrupacions no desitjades.

Aspectes estadístics

- **Densitat:** Aquesta propietat té molta relació amb la dimensió del període i del cicle de la sèrie de nombres generada.
  - A més gran densitat, més gran precisió i finor en la definició dels fenòmens.

Aspectes estadístics

- **Independència estadística:** Bàsicament una seqüència de nombres aleatoris es aquell conjunt de nombres sobre els que no es pot establir cap relació rellevant amb els restants nombres de la sèrie.
  - Molts dels mètodes de generació més habituals son recursius, per lo que no es pot afirmar-se el principi anterior ja que cada nombre es calculat mitjançant un algoritme determinista aplicat a un altre nombre que l'ha precedit en la sèrie.
  - Es per aquest motiu que els nombres s'anomenen pseudo-aleatoris.
  - Cal assegurar que compleixen els requisits especificats mitjançant la superació de determinats tests.

Aspectes teòrics

- **Complexitat:** els procediments de generació han de ser necessàriament simples, poc costosos en termes computacionals i, en conseqüència, la complexitat algorítmica dels mateixos haurà d'estar convenientment acotada.

Aspectes teòrics

- **Estabilitat del procés:** Els algoritmes de generació (GNA) han de ser estables i no degenerar a lo llarg de la seva utilització en un experiment de simulació.

Aspectes computacionals

- **Facilitat de programació**: possibilitat de realitzar probes i depuració d'errors.
  - Implementació raonable, fiabilitat, mantenibilitat y capacitat de reutilització.

Aspectes computacionals

- **Requeriments de memòria:** Quantitat de memòria usada pel GNA.
  - Encara que aquest no és un requeriment tant crític com ho va estar en el passat, criteris de racionalitat en determinats problemes poden aconsellar l'ús d'algoritmes amb baixa ocupació de memòria.

Aspectes computacionals

- **Eficiència:** La velocitat d'execució del GNA.
  - Fa referència al temps de posada a punt i el temps d'execució.
  - En alguns projectes, com l'aplicació de mètodes de Montecarlo en l'àmbit de la física o de la biologia, pot arribar a ser un factor fonamental.

Història dels GNA

- Les primeres famílies de generadors es van basar en mètodes físics o mecànics
  - Taules, nombres de loteries, boles d'urna, emissions de raig gamma, discs perforats, raigs còsmics, soroll tèrmic, fenòmens radioactius

Història dels GNA

- Aquests mètodes no son útils en simulació:
  - No permeten reproduir fàcilment les sèries. Introdueixen una gran complexitat en la depuració d'errors i en la gestió d'experiments.
- El professor *George Marsaglia* ha elaborat i comercialitza un CD amb 60 fitxers de 10 Mb de nombres aleatoris binaris usables per experimentació:
  - Combinació de diferents GNA.
  - Dispositius físics
  - Cançons de RAP
  - http://stat.fsu.edu/~geo/

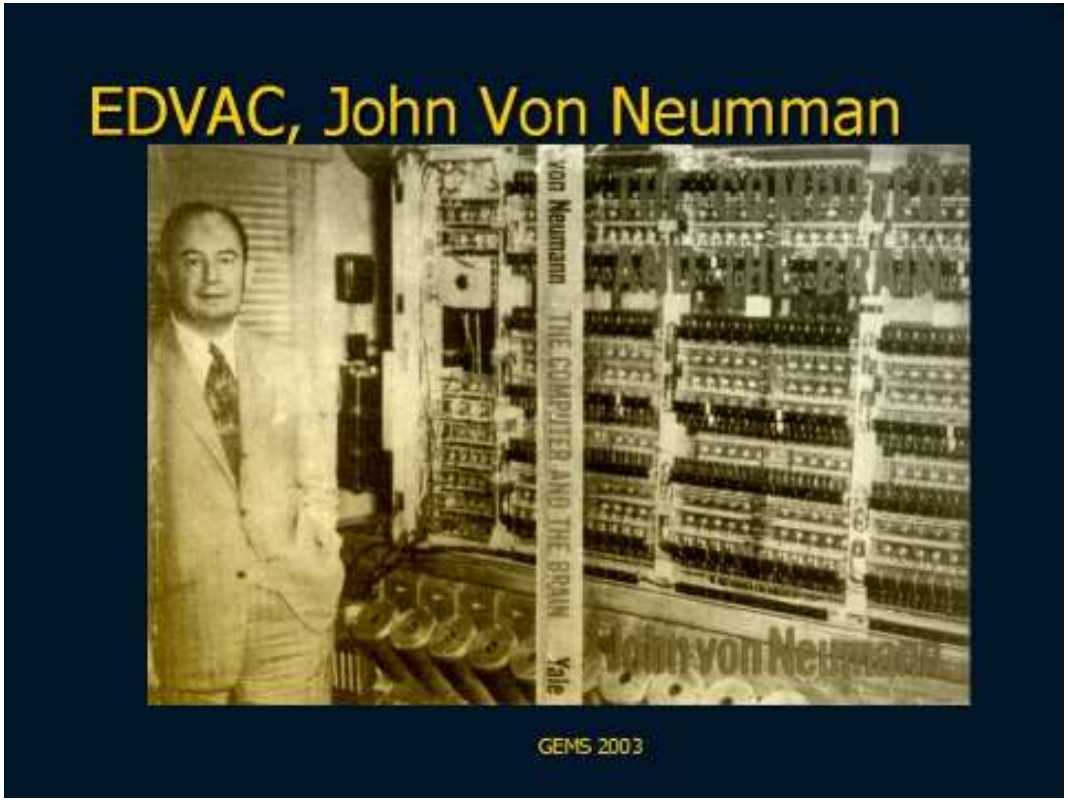
Història

- *El primer generador aritmètic va ser proposat per Von Neumann*
- *y Metropolis en 1940 : el "Mètode del quadrat mig"*

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
:	:	:	:

Història

- 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.



Tipus de GNA

- Marsaglia y Zaman (1991) classifiquen els GNA en tres classes principals:
  1. Generadors Congruencials (de congruències).
  2. Generadors Shift – Register.
  3. Generadors *lagged* – Fibonacci.

Generadors congruencials

- Los GCL van ser introduïts per Lehmer al 1951.
- Molts dels generadors que existeixen en l'actualitat son d'aquest tipus.
- Aquests generadors estan definits per la següent fórmula recursiva:
  - $Z_i = (a\ Z_{i-1} + c)\ (\mathrm{mod}\ m)$
  - **m** es el mòdul.
  - **a** es el multiplicador
  - **c** es el factor additiu.
  - **Zo** es la llavor
- Es compleix, evidentment, que $0 \leq Z_i \leq m-1$

Generadors congruencials

- Per obtenir els nombres que es desitja, conforme una U[0,1) cal fer la divisió
  - Ui = Z i /m
- Per evitar la negativitat els enters m, c, a y Z 0 han de complir:
  - 0 < m, a < m,
  - c < m, y Z 0 < m.

Generadors congruencials

- Exemple de GNA de cicle complert.
- Donat que m=16 el màxim nombre de números que pot generar es 16.
- Es convenient que el valor de ***m*** sigui molt gran, tant com puguem, per exemple de l'ordre de10^9.

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

Generadors congruencials

- **Cicles**: quan Zi pren un valor igual a un altre Zj amb j<i apareix un cicle, els valors es repeteixen eternament.
- **Període**: la longitud d'aquest cicle, "i".
  - Donat que $0 \leq Z\,i \leq m-1$, el valor mes gran del període es ***m***.
- **Període complert** (full period): quan i=m

Generadors congruencials

- Selecció dels paràmetres:
  - ***c*** y ***m*** han de ser primers entre si.
  - Si ***q*** es un nombre primer que divideix a ***m***, llavors ***q*** també divideix a ***a-1***
  - Si ***m*** es múltiple de 4, ***a-1*** es múltiple de 4 (***a*** no pot ser múltiple de 4).
- Aquestes condicions es compleixen si:
  - $m = 2^b$
  - $a = 4k+1$
  - c senar (con a, c y k enters positius)
- D'aquesta forma obtenim període complert.

Generadors congruencials

- Els generadors congruencials es poden classificar en funció dels valors que prenguin la seva variable c:
  - si $c > 0$ GCL Mixt
  - si $c = 0$ GCL Multiplicatiu
  - si $c \neq 0$ y $a=1$ GCL Additiu

LCG Mixt c>0

- Per obtenir períodes llargs densitats altes dels U[0,1) es desitjable un valor molt gran de ***m***
- Dividir entre ***m*** es una operació que es pot eliminar de forma explícita.
  - Un bon valor de ***m*** es 2^b , on b es el número de bits de la paraula del ordinador (por exemple 32 o 64 bits).
  - L'enter mes gran que es pot obtenir es $2^b - 1$.

LCG Mixt c>0

- Aquesta ***m*** permet aprofitar el "integer overflow" en l'ordinador i evitar la divisió explícita por ***m***.
- *Exemple:*
- Suposant que es té un ordinador amb longitud de paraula de 4 bits: $m = 2^4 = 16$.
  - Si $Z_0 = 5$, $a = 5$ y $c = 3$
  - $(aZ_{i-1} + c) = (5*5)+3 = 28 = 11100$
- Com la longitud de paraula es únicament de 4 bits s'elimina el primer dígit de l'esquerre:
  - $1100 = 12 \therefore Z_1 = 12$

LCG Multiplicatius c=0

- Impedeix obtenir períodes complerts.
- Es poden obtenir períodes de mida ***m***-1, si els valors de ***m*** i ***a*** son seleccionats adequadament.
- Al igual que per c>0 es convenient usar $m=2^b$ per eliminar explícitament la divisió.

LCG Overflow

- El principal problema dels GCM radica en el fet que al usar nombres enters podem tenir problemes de *overflow*.
- Per solucionar aquest problema, Schrage [SCHR79] va desenvolupar el mètode de la *divisió simulada*.
- Aquest mètode es basa en el fet de que l'operació calculada perla formula recursiva del LCG es pot expressar com:
  - $ax(\mathrm{mod}m) = g(x) + mh(x)$

El mètode de Schrage

- El mètode es pot sumaritzar en el següent procediment:
  - $q = m / a$
  - $r = m \bmod a$
  - $k = n_i / q$
  - $n_{i+1} = a * (n_i - k * q) - r * k$

Altres generadors

- **Congruències més generals.**
- **Generadors compostos (*Composite Generators*)**
- **Generadors de Tausworthe**
- **AWC/SWB**

Congruències més generals.

- Un GCL pot ser pensat com un cas especial de generador definit per:
  - $Z_i = g(Z_{i-1}, Z_{i-2}, \ldots) \pmod{\boldsymbol{m}}$
  - Z_i cau entre 0 i *m-1*. *Els* U_i s'obtenen dividint Z_i/m
  - ***g*** es una funció determinista del valor previ Z_i.
- Així, una generalització de los GCL seria:
  - g(Zi-1, Zi-2, ...)=a'Z2
  - i-1+aZi-1,+ c (Gen. Quadràtic)
- Un cas especial d'aquest mètode es quan a'= a =1, c=0 y ***m*** es potencia de 2. En aquest generador, donat que Zi només depèn de Zi-1 y de que $0 \leq Zi \leq m-1$, el període es, com a molt, tan gran como ***m***.

Tests de GNA

Validesa dels GNA

Tests o proves de GNA

- En l'actualitat, tots els generadors usats en aplicacions professionals es basen en un **mecanisme determinista** (no son pròpiament aleatoris per natura, per aquest motiu els denominen pseudo-aleatoris).
- No existeix garantia d'aleatorietat en funció d'un mètode determinat **d'elecció de paràmetres** (per exemple els paràmetres a, c i m de un GCL).
- Usant GNA únicament garantim:
  - Un llarg període.
  - L'eficiència aritmètica del procediment.

Tests o proves de GNA

- Las probes sobre la qualitat dels generadores de nombres aleatoris es classifiquen en: (segons Law y Kelton, 1999):
- ***Empíriques***: Son proves de tipus estàtic.
- ***Teòriques***: No son proves en el sentit estadístics. Es basen en els paràmetres numèrics del generador per provar las seva qualitat.

Proves empíriques

- Aquest tipus de proves examina de forma estadística els nombres obtinguts mitjançant un generador per esbrinar quan s'assemblen a uns nombres IID de una U[0,1)
- Es busca si es compleixen les propietats que exigiríem a una tira realment aleatòria.
  - **Uniformitat** de la distribució dels valors de la seqüència.
  - **No correlació** en la seqüència.

Proves empíriques

- La uniformitat pot venir garantida per l'ús d'un generador apropiat amb el que es pugui obtenir un "període complert".
- Els tests processen:
  - n.a.'s $\{u_i\}$ $0 \leq u_i \leq 1$ $i=0,1,2,\ldots$
  - enters $\{y_i\}$ $0 \leq y_i \leq d$ $i=0,1,2,\ldots$

Test de Chi quadrat

- Es divideix el rang [0,1] en ***k intervals iguals***, amb (k>100, y n/k ≥5).
- Sigui fj (j=1,2,...,k) la quantitat de nombres aleatoris que cauen en cada un dels intervals j. Si estan ben distribuïts, la freqüència trobada coincidirà amb l'esperada n/k.
- Es calcula el valor de Chi:

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

Exemple test de Chi^2

- Valors obtinguts amb:
- $Z_i = 630{,}360{,}016\ Z_{i-1} \pmod{231-1}$
- $k=2^{12} = 4096$, $n=2^{15} = 32{,}768$
- S'obté $\chi^2 = 4141$
- Mentre que

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

- Per tant es pot considerar que els nombres obtinguts amb aquest generador son IID U(0,1)

Test de sèrie

- Es una generalització del Chi quadrat per dimensions mes grans
- Es separa tota la sèrie de nombres generats en vectors de mida
  - ***d***: U1=(u1, u2, ..., ud) , U2=(ud+1, ud+2, ..., u2d), ...
  - I es calcula el valor de Chi quadrat amb aquesta fórmula:

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

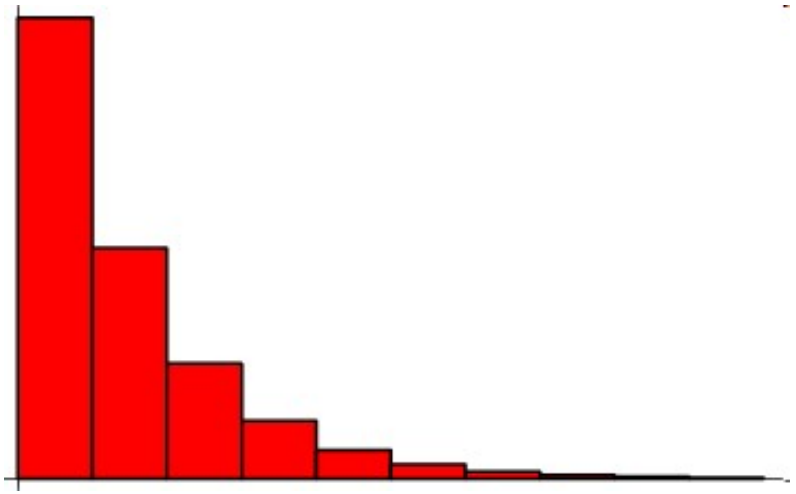
Exemple de test de sèrie

- d=2, H_0: Els parells (ui,u2), (u3,u4), ..., (ud-1, ud) son IID U(0,1).
- Es van generar 32,768 pares de Ui´s. K=64 ∴ g.l. $= 64^2 - 1 = 4095$
- α=0.10 χ^2 =4211.4 y $\chi^2(2)$=4016.5
- ∴ S'accepta la uniformitat dels nombres generats.

Test de Gap

- Un gap es una sub-seqüència de **r** valors consecutius amb una distància entre ells compresos entre a i b:
  - $0 \leq a < b \leq 1$
- Al examinar una seqüència de longitud n calculem el nombre de gaps de longitud 0,1,2,…,(t-1) y r≥t
  - P=prob(a≤ u_k ≤b)
  - Pt=P(1-P)t es la probabilitat d'un gap de long. t (si la seqüència és uniforme i independent). Segueix una **geomètrica** de paràmetre $P\{Ui(a, b)\} = b - a$
- Apliquem X^2.

Distribució geomètrica



- No té memòria com l'exponencial
- Pot representar el nombre de "fallades" abans del primer èxit.

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

Exemple test de Gap

- Generem nombres entre (0,1)
- Anotem la distància entre succesives aparicions de nombres en l'interval [a,b] de la seqüència.
- Per exemple si
  - [a,b] = [0.4,0.7]
  - La seqüència es: 0.1, 0.5, 0.9, 0.6, ...,
  - La longitud del primer GAP es 2 (entre els nombres 0.5 i 0.6)
  - Es guarden “n” GAPS i s’agrupen tots els que tinguin un valor més gran que t en la mateixa categoria.

Test de Run

- És una prova d'independència i no d'uniformitat.
- Consisteix en examinar la seqüència de Ui's per detectar segments de nombres ascendents (run up), o descendents.
- S'aplica un test de Chi quadrat a les freqüències observades i esperades.

Exemple de test de run

- 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	mida	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	Execucions de mida i (r_i)
1	2
2	2
3	0
4	1
5	0
6	0

Exemple de test de run

- $r_1=2$, $r_2=2$, $r_3=0$, $r_4=1$, $r_5=0$, $r_6=0$
- Per n>4000, R s'aproxima a una distribució Chi quadrat amb 6 graus de llibertat.
  - H_0: els Ui's son IID variables aleatòries.
- Per n<4000 s'aplica la fórmula:

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

- On a_{ij} es el (i,j) ésim element de la matriu que proporciona Knuth en el seu llibre "Handbook of Simulation" al igual que els b_i´s.

Test de permutació

- Per a un enter positiu *t* tal que *t!* Sigui petit en comparació a la longitud de la seqüència (Ui).
- Dividir Ui en blocs de mida *t*, (*U*1,…, *Ut*), (*Ut*+1,…, *U*2*t*), ….
- Sota la hipòtesi H_0 les *t!* possibles ordenacions dintre d'un bloc de longitud t han d'estar uniformement distribuïdes.
- Un test de Chi quadrat es fa per provar la seva independència.

Proves teòriques

- Aquest tipus de proves realitzen un anàlisi global, es a dir, examinen el cicle complert. Les proves teòriques més conegudes son:
  - Spectral test.
  - Lattice test.
- Es basen en el que va descobrir Marsaglia el 1968, els nombres aleatoris produïts per un generador cauen principalment en plans.

Proves teòriques

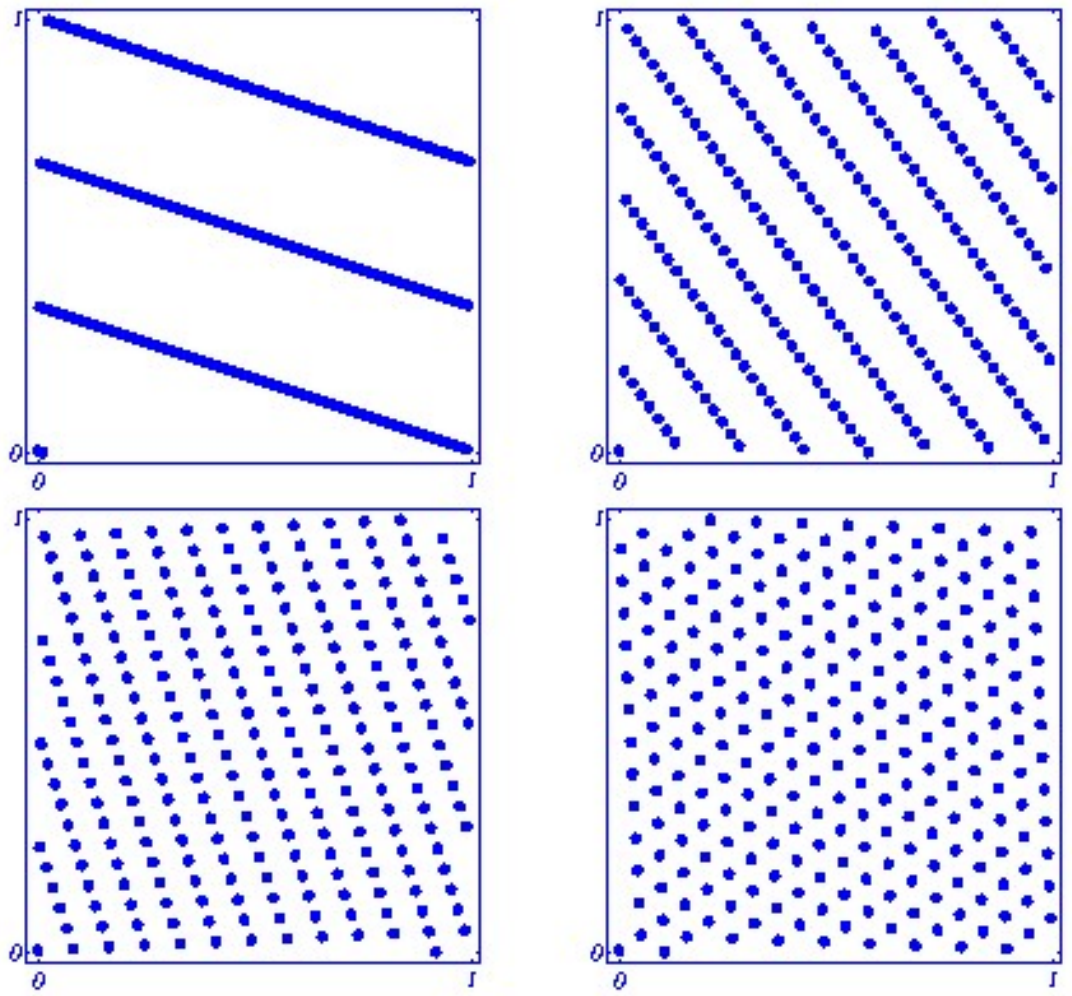
- Si u1, u2 ,... es una seqüència de nombres generats per un LCG, la superposició de segments de mida ***d*** de la sèrie (u1,u2,...,ud), (u2,u3,...,ud+1), totes elles cauran en un nombre relativament petit de plans a través de tota la unitat hiper-cúbica ***d***–dimensional [0,1)***d***.
- El test espectral retorna la distància entre els diferents hiper-plans.

Exemples (dimensió 2)

- LCG(256,a,1,0) amb a = 85, 101, 61, 237.
- Test espectral de 0.3162, 0.1162, 0.0790, 0.0632.
- Test espectral normalitzat de 0.1839, 0.5003, 0.7357, 0.9196.

Exemples (dimensió 2)

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



Exemples (dimensió 2)

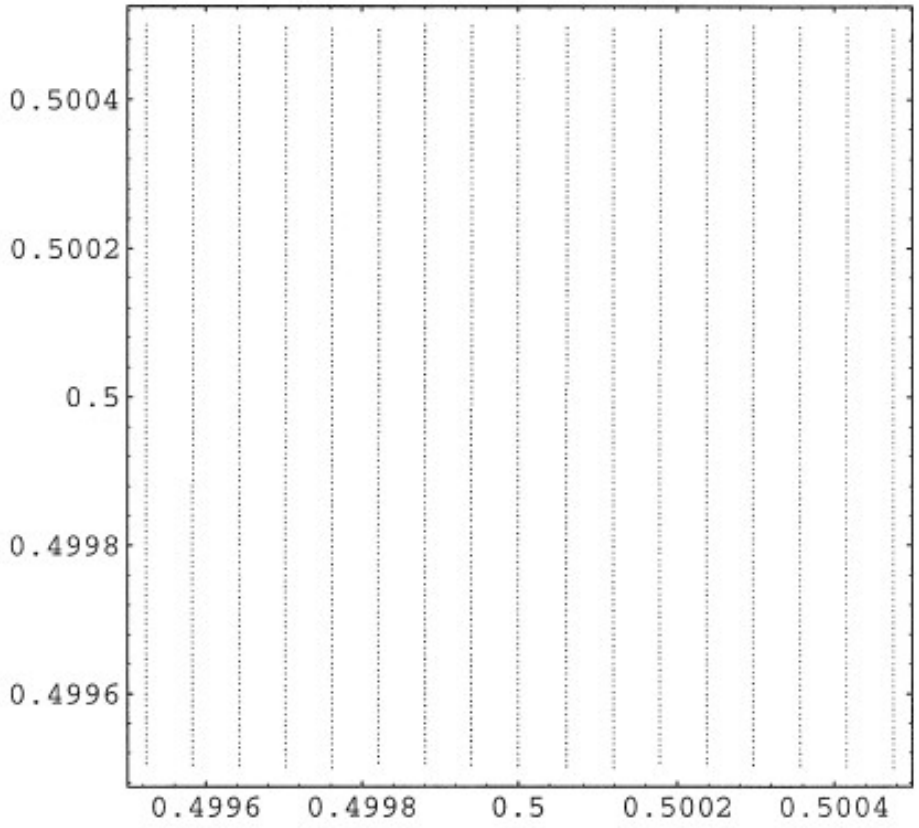


Fig. 3. Minimal Standard: LCG($2^{31}-1$,16807,0,1).

Exemples (dimensió 2)

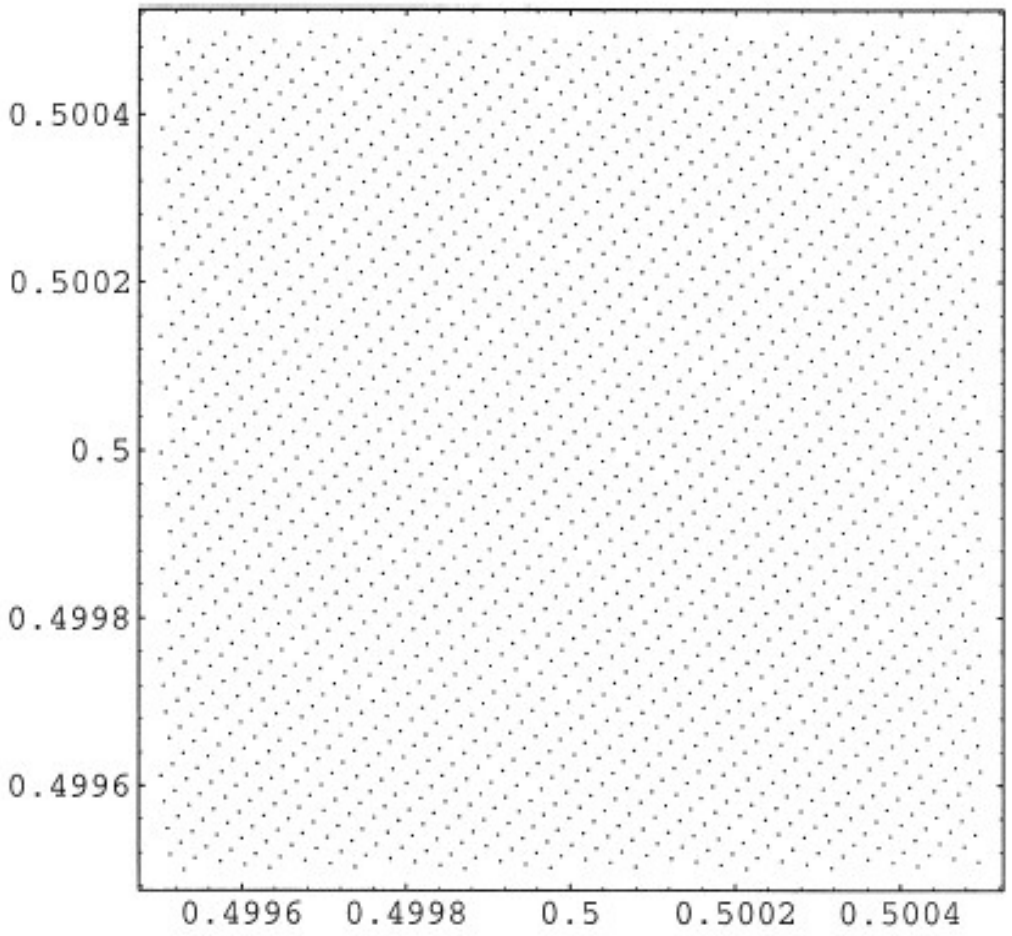


Fig. 4. SIMSCRIPT: LCG($2^{31}-1$,630360016,0,1).

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

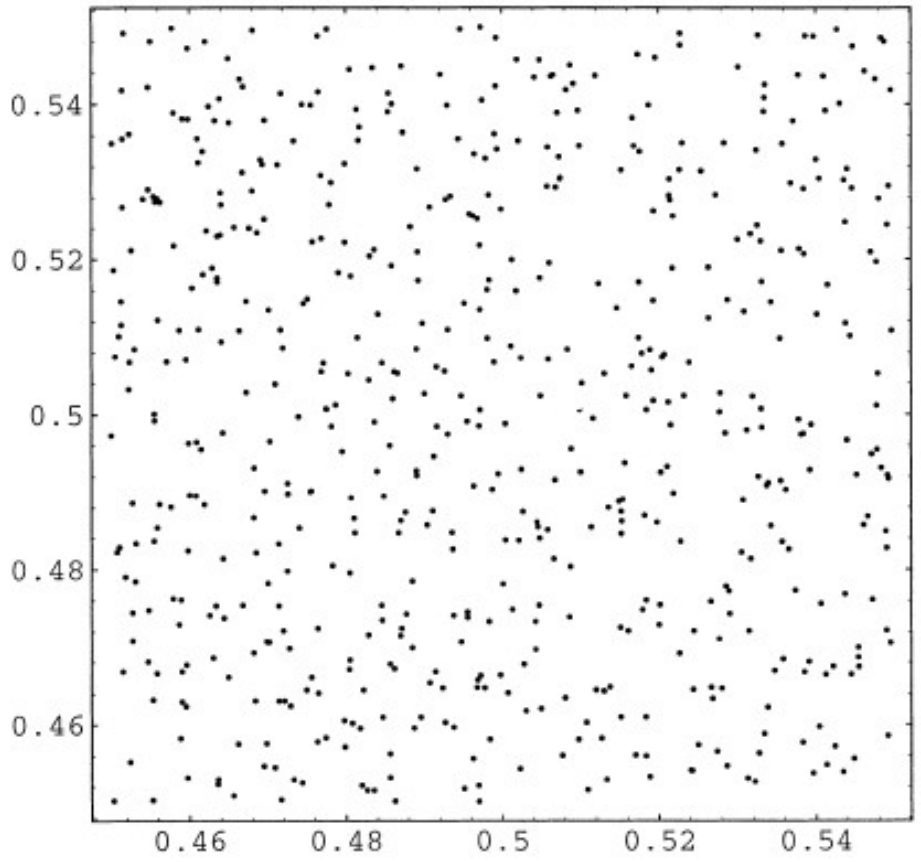


Fig. 1. LCG(2^{31}, 65539, 0, 1) Dimension 2: Zoom into the unit interval.

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

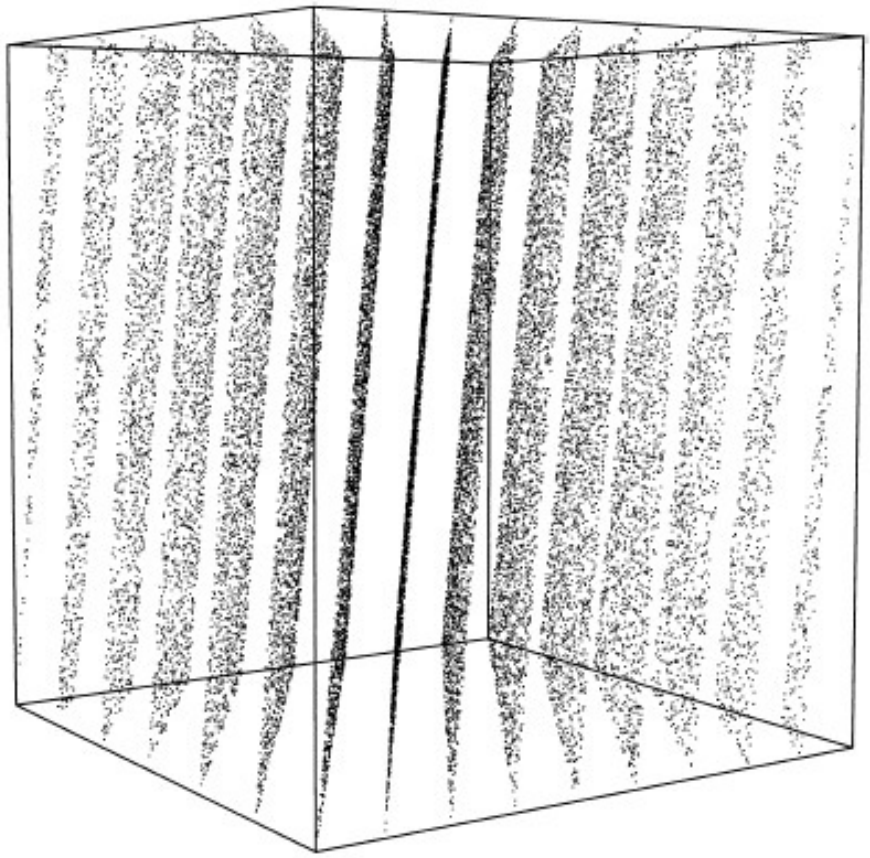


Fig. 2. LCG(2^{31}, 65539, 0, 1) Dimension 3: The 15 planes.

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

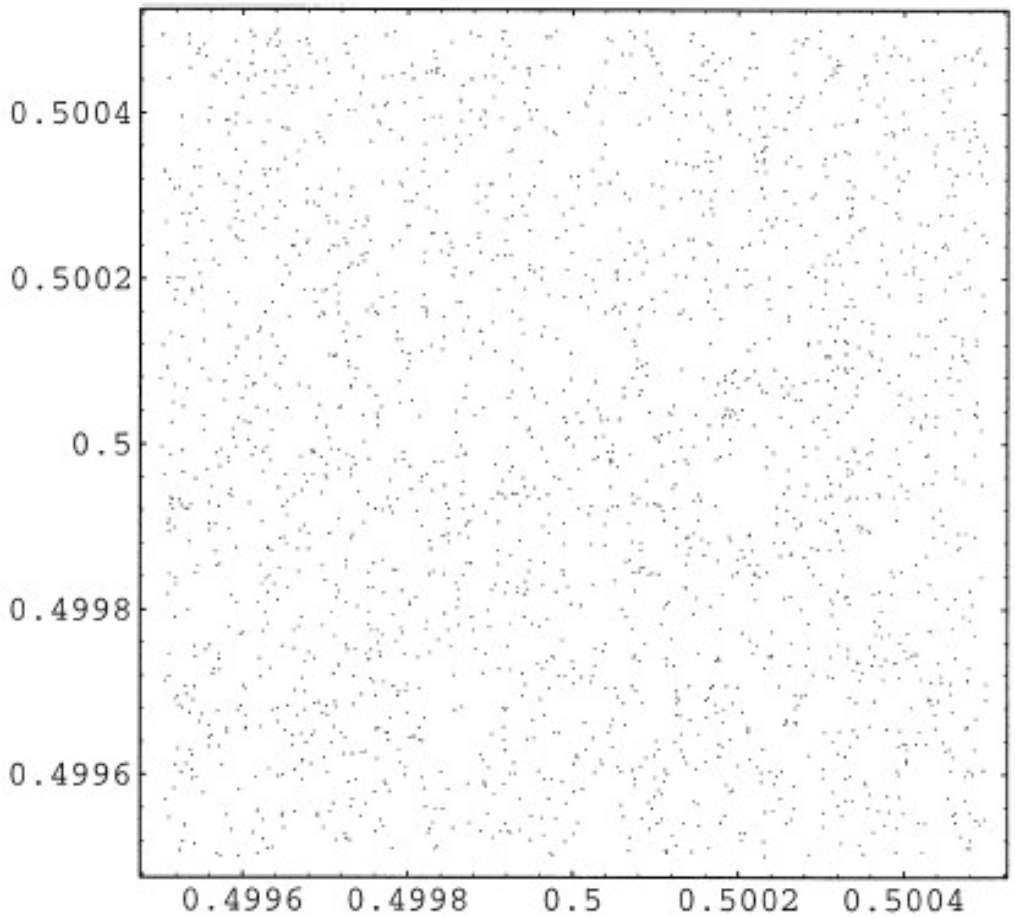
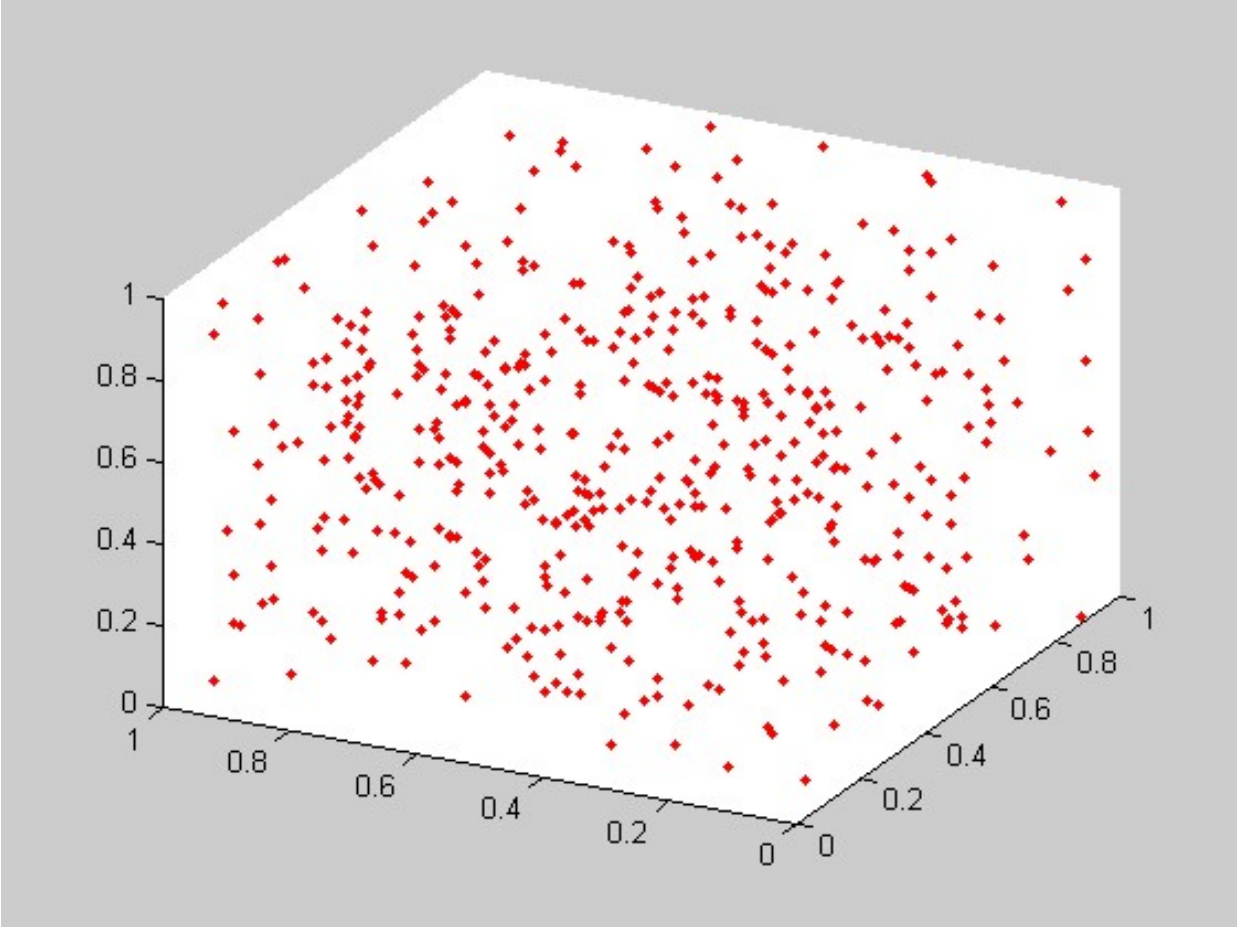


Fig. 5. All points of ICG($2^{31}-1$,1288490188,1,0).

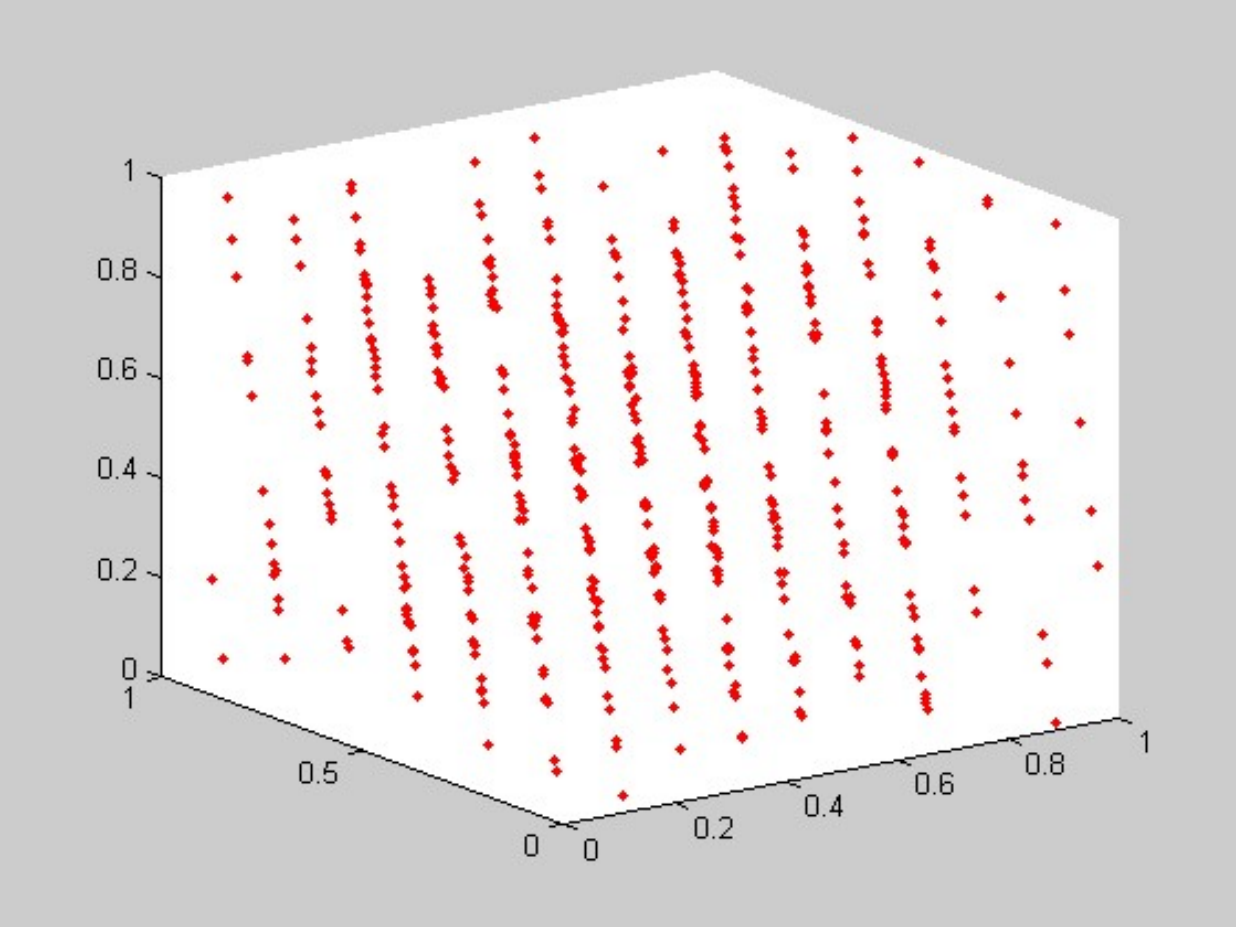
RANDU MatLab

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

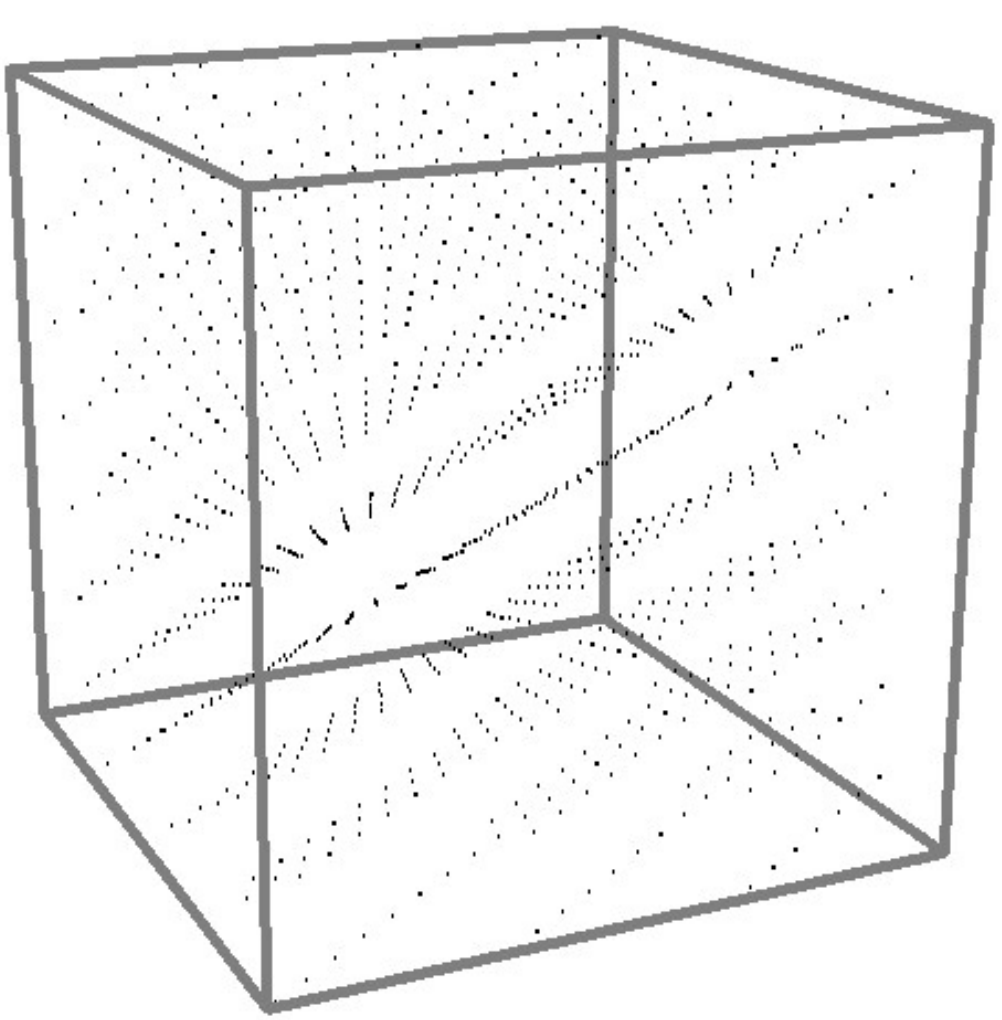


RANDU MatLab

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

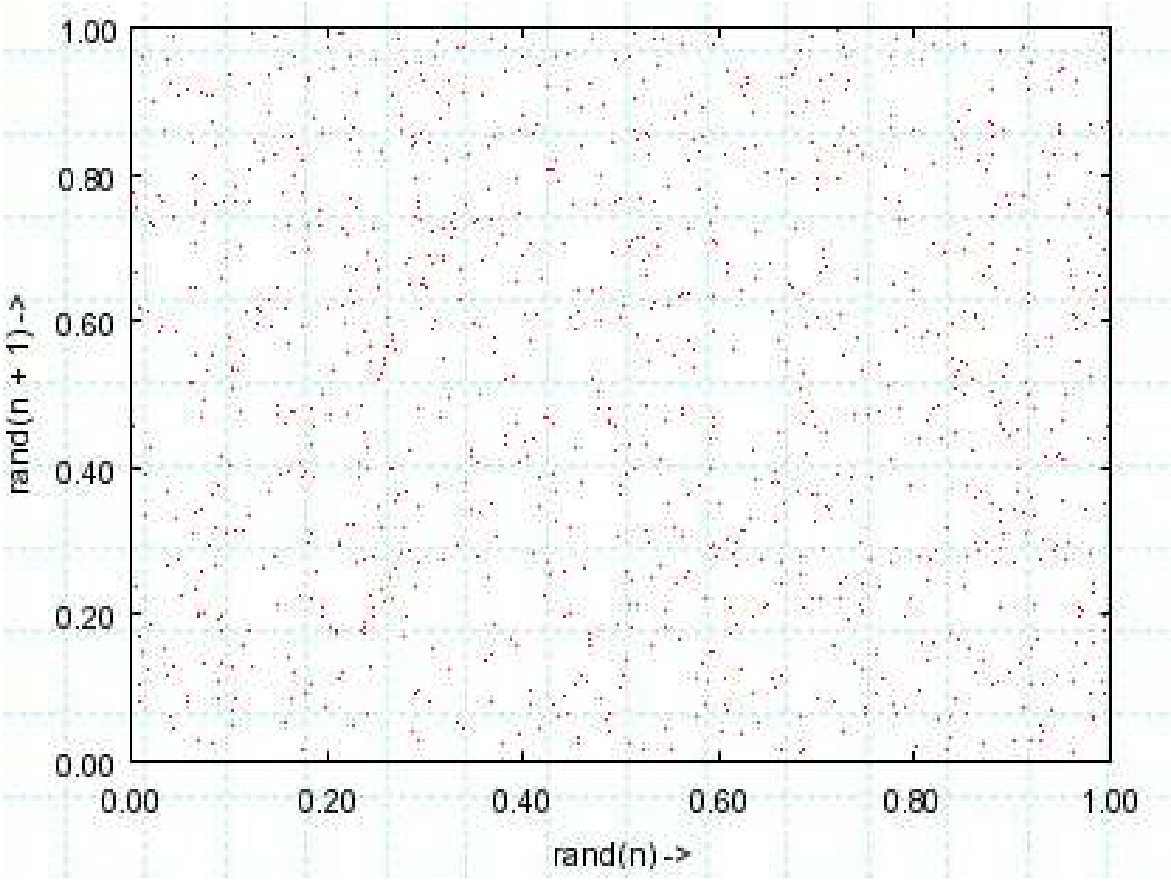


Exemples (dimensió 3)



Representació a GNUPlot

```
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)
```

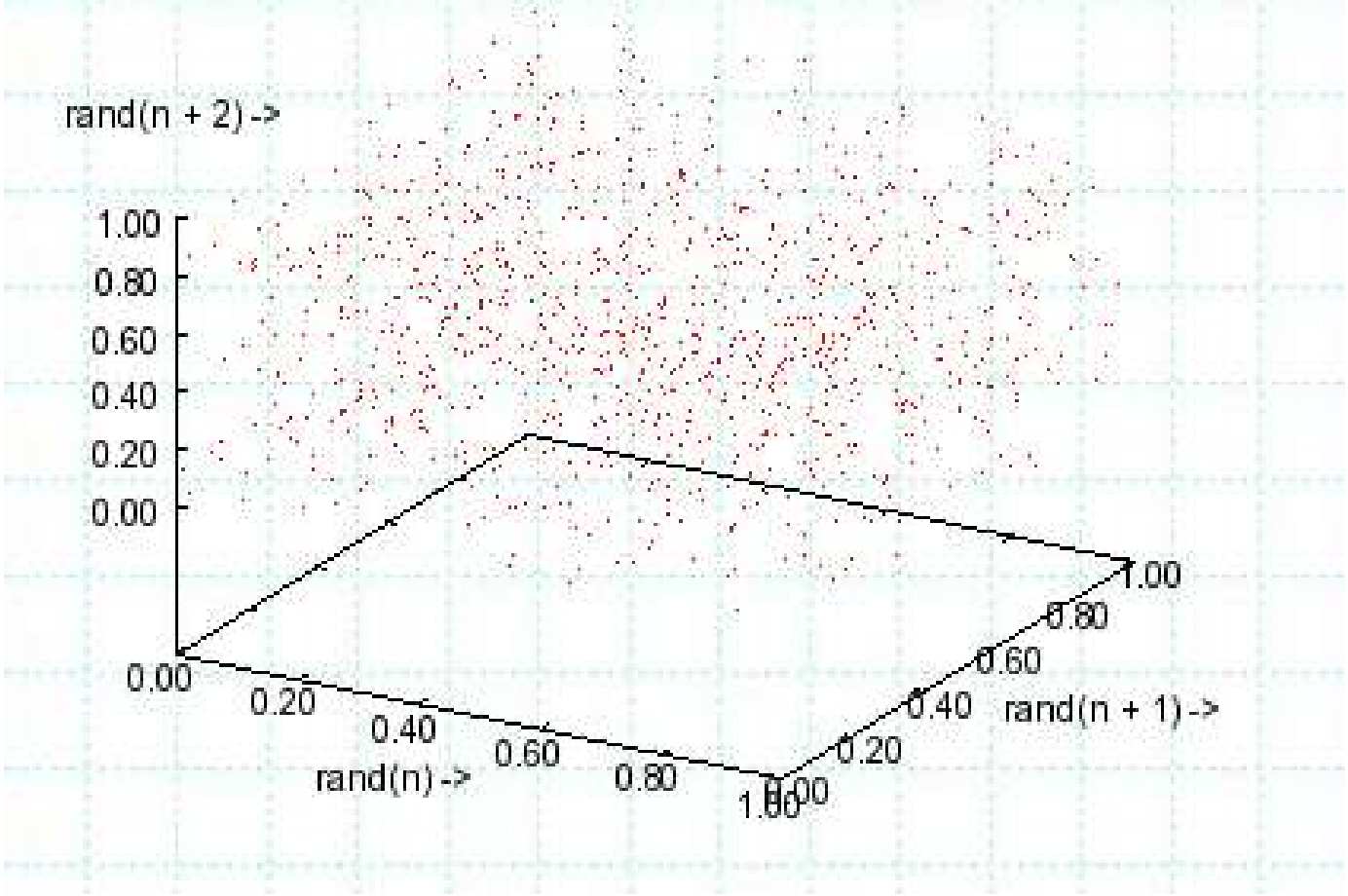


Representació a GNUPlot 3D

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

set sample 50

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



Exercici: Test de Chi quadrat

- A desenvolupar a classe.

GVA

Generació de variables aleatòries

GVA Mètode de la transformada inversa

- Considerem una variable aleatòria X, amb una distribució acumulada $F_X(x)$. Es senzill mostrar que la variable aleatòria:
  - $U = F_X(x)$
- Té una distribució U(0,1). Si $F_X(X)$ es estrictament creixent, es pot reescriure l'equació en la seva forma equivalent
  - $X = F_X^{-1}(U)$
- Aquesta transformació s'anomena la transformada inversa. Presenta un bon mètode quan la inversa de $F_X(x)$ es fàcil de calcular.

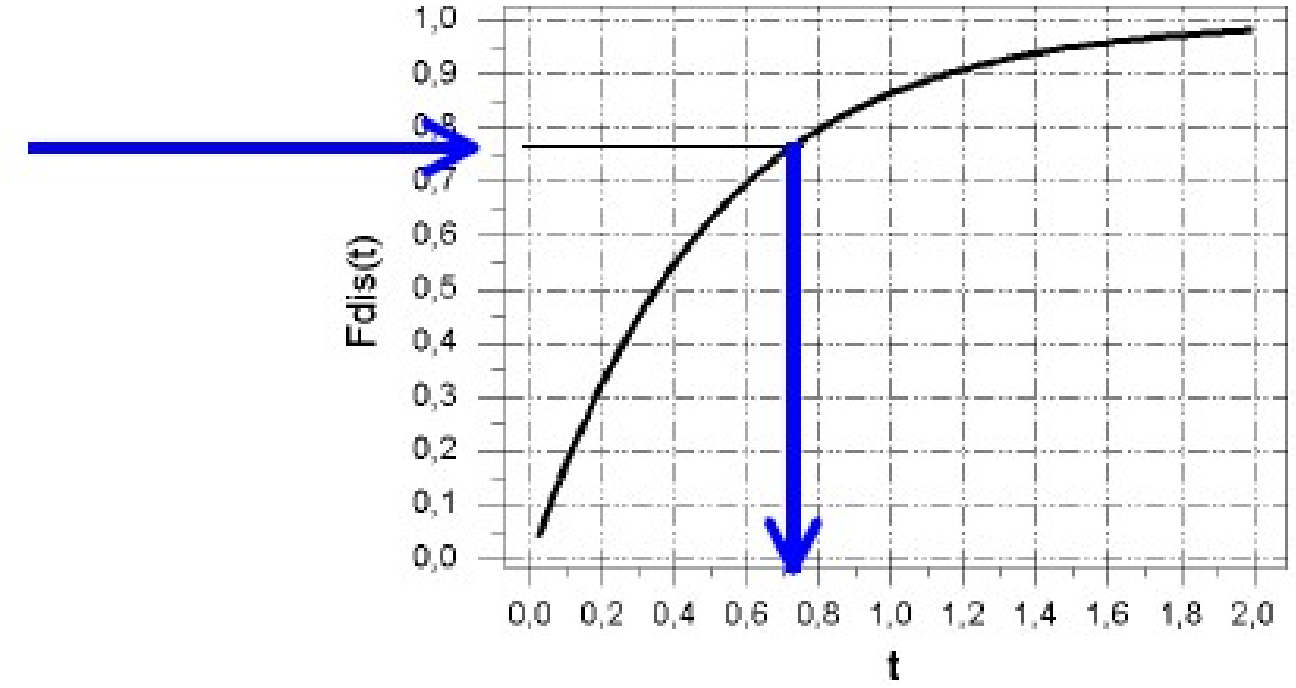
GNA Mètode de la transformada inversa (Exemple 1)

- **Exemple:** Sigui X una variable aleatòria que segueix una distribució exponencial amb paràmetre λ. Llavors:
- $F_X(x) = 1 - exp(-\lambda x)$ si $x > 0$ and $F_X(x) = 0$ altrament.
- A partir d'aquí: $F_X^{-1}(u) = -\lambda^{-1} ln(1-u)$
- El mètode de la transformada inversa es pot aplicar també a distribucions discretes,però no es pot calcular la inversa de la distribució acumulada. No obstant es pot agafar la inversa generalitzada definida per:
  - $FX^{-1}(u) = \min\{x \mid u \leq FX(x)\}$

Transformada inversa: Exponencial

- Mètode de la transformada inversa per a la distribució Exponencial.

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



GNA Mètode de la transformada inversa (Exemple 2)

- Sigui X una variable aleatòria Bernoulli amb paràmetre *p*. La seva funció de distribució acumulada es donada per
  - $FX(x) = 0$ si $x < 0$
  - $FX(x) = 1 - p$ si $0 \leq x < 1$
  - $FX(x) = 1$ si $1 \leq x$
- Per tant
  - Si $1 - p < u$ llavors $FX\text{-}1(u) = 1$
  - Sinó $FX\text{-}1(u) = 0$.
- Substituint *u* per $1 - u$ es pot observar que la variable aleatòria de Bernoulli amb paràmetre *p* pot ser generada de una variable aleatòria U(0,1) amb el següent algoritme:
  - Si $U < p$, llavors retorna 1.
  - Sinó retorna 0.

GNA Mètode d'acceptació - rebuig

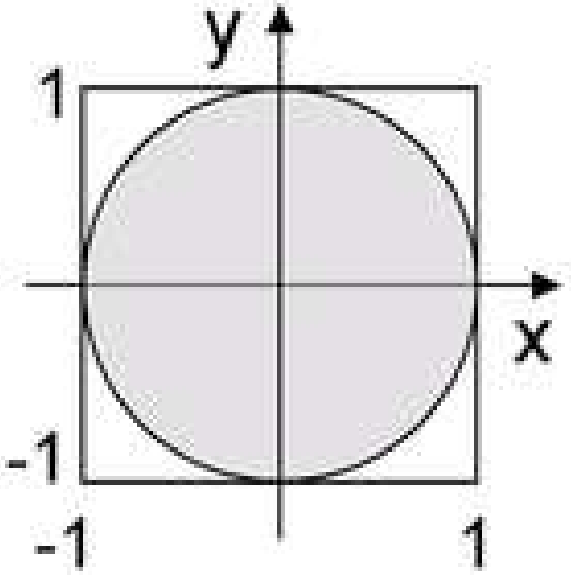
- S'explica el mètode d'acceptació – rebuig per a distribucions contínues.
- Suposem que es vol simular la variable aleatòria X amb densitat f(x).
- És difícil amb el mètode de la transformada inversa.
- Suposem que coneixem un mètode per simular una variable aleatòria amb densitat g(y) tal que:
- $f(y) \leq c \cdot g(y)$
- Per tot y i una constant positiva c, el mètode d'acceptació – rebuig es descriu amb el següent procediment.
  1. Generar Y a partir de la distribució de densitat $g(y)$.
  2. Generar U de U(0,1).
  3. Si $U \cdot c \cdot g(y) \leq f(y)$, retornar X=Y sinó retornar al pas 1

GNA Mètode d'acceptació - rebuig

- El següent teorema mostra que la variable aleatòria X té efectivament densitat f(x).
- **Teorema:** La variable X generada a partir del mètode d'acceptació rebuig té densitat f(x).
- Demostració:
- Cal usar el fet que a partir del procediment descrit
  - $P(X \leq x) = P\{Y \leq x \mid U \leq f(y)/(c \cdot g(y))\}$
- Amb U i Y sent independents.

GNA Mètode d'acceptació - rebuig

- **Exemple:** Es volen generar nombre que pertanyin a una circumferència:
  - *Es generen punts (x,y) que pertanyen a (-1,1).*
    - *En aquest cas c=1, g(y)=* 2u-1, on u∈[0,1).
  - *S'avalua si compleixen la condició:* $x^2+y^2 \leq 1$
  - En cas que la compleixin s'accepten, si no es rebutgen.



GNA Mètode d'acceptació - rebuig

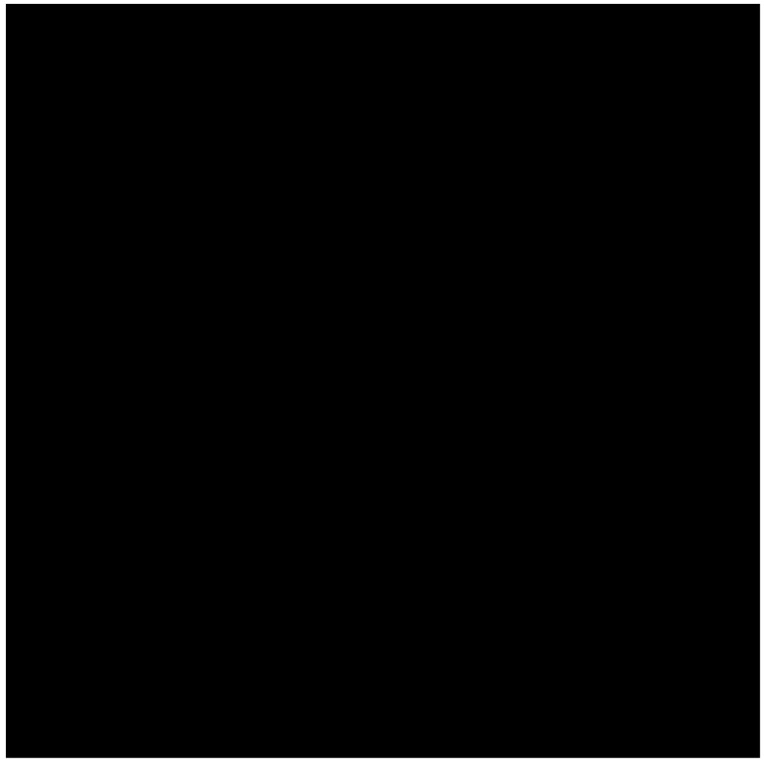
- **Exemple:** Sigui Z una variable aleatòria normal. Suposem que es vol simular el valor absolut $X=|Z|$ que descriu la desviació del seu valor mitjà. La seva densitat de probabilitat es donada per:
  - $f(x) = (2/\pi)^{1/2} exp(-x2/2)$
- Es pot agafar Y una variable aleatòria exponencial amb paràmetre 1, per tant $g(y) = exp(-y)$. Es fàcil mostrar que
  - $f(y)/g(y) = (2e/\pi)^{1/2} exp(-(y-1)2/2)$
- Per tant $c = (2e/\pi)^{1/2}$.

GNA Mètode d'acceptació - rebuig

- Òbviament hom vol seleccionar $c \cdot g(y)$ tant a prop com sigui possible de $f(y)$ per tal de mantenir la proporció de punts rebutjats petita, però per una altra banda, $g(y)$ es selecciona de forma que tingui una densitat simple per evitar llargs càlculs per Y.

GNA Mètode d'acceptació - rebuig

- L'algoritme de **Ziggurat** permet generar valors que segueixi una distribució de probabilitat normal.
- http://en.wikipedia.org/wiki/Ziggurat_algorithm



The Alias Method

for generating values from finite discrete distributions

Mètode Àlies

Aquest mètode es vàlid per variables amb probabilitat concentrada en un nombre finit de punts.

$$X\ t.q.\ P(X = x_i) = p_i\ on\ i = 1,2,..,k$$

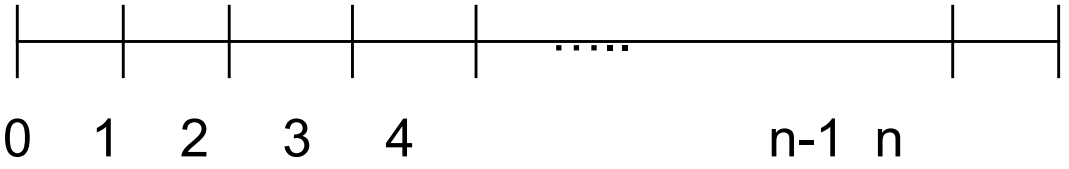
Cal una fase prèvia de pre-procés.

Discrete Uniform

- Suppose we want to simulate

  $X \sim \text{unif}\{1,2,\ldots,n\}$

- We could do this by generating $U \sim \text{unif}(0,1)$,

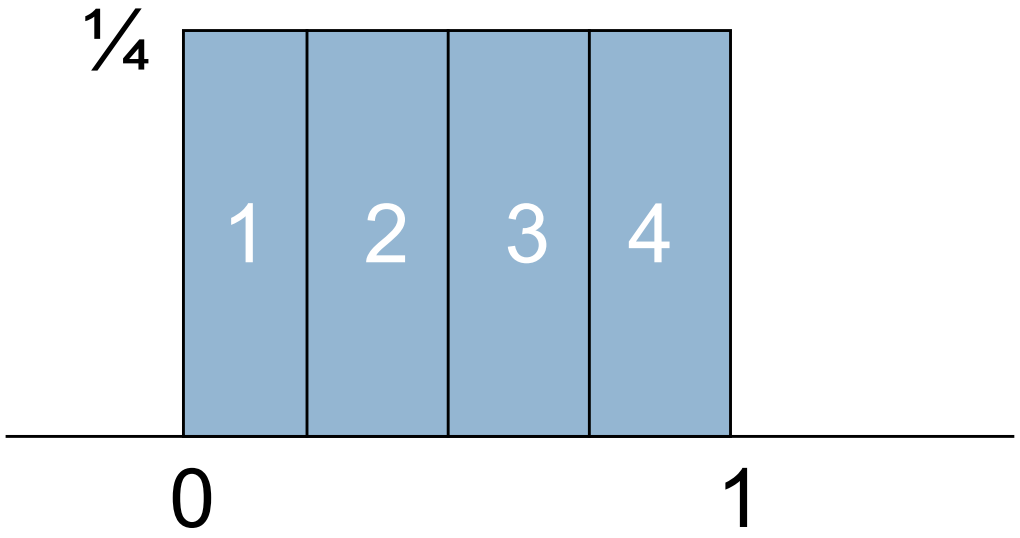


and letting $X = \lceil nU \rceil$.

Discrete Uniform (alternative)

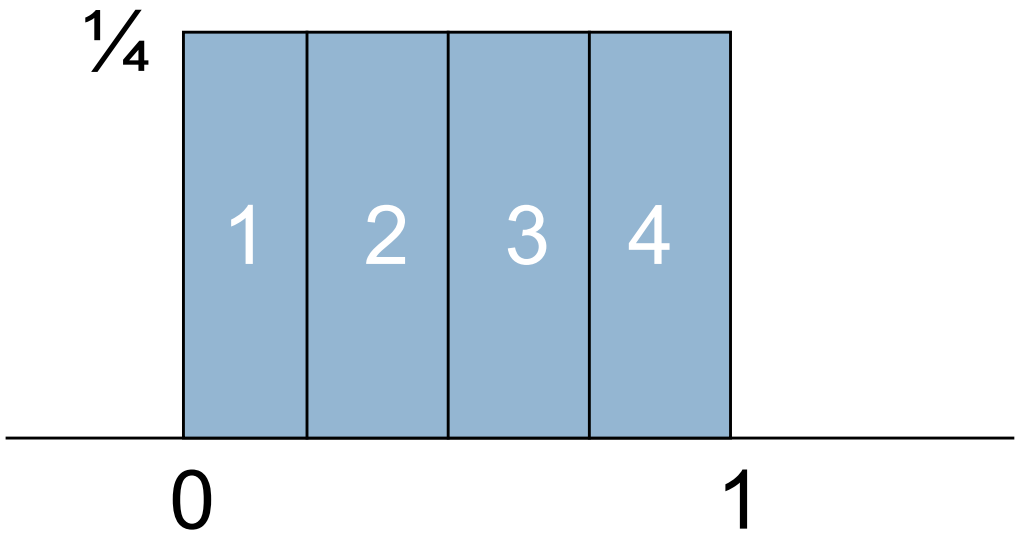
- Alternative, not intended for use, but to motivate the alias algorithm for more general discrete distributions:

- By example: X~unif{1,2,3,4}



Discrete Uniform (alternative)

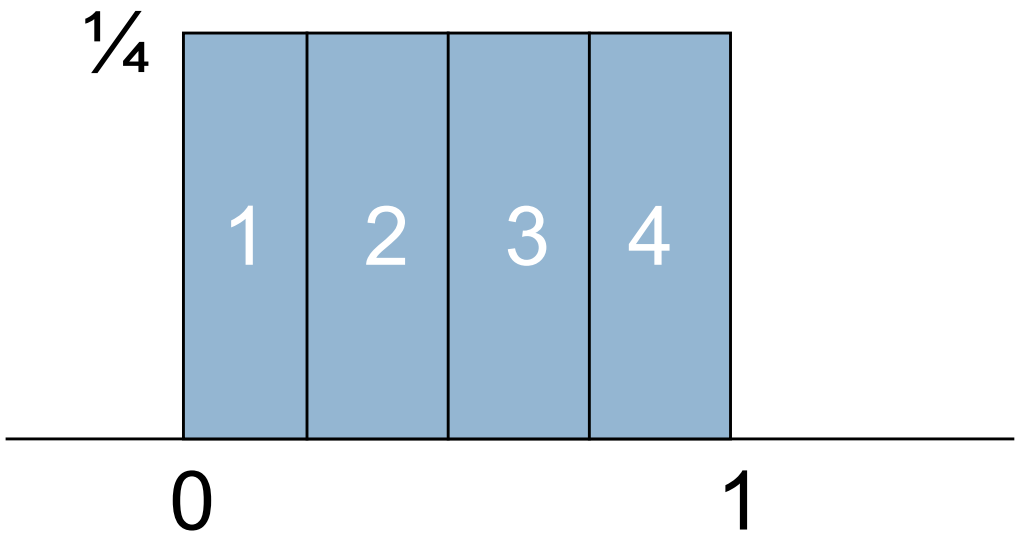
- By example: X~unif{1,2,3,4}



- Draw points, (x,y), uniformly in this rectangle, return the rectangle number.

Discrete Uniform (alternative)

- By example: X~unif{1,2,3,4}



- For this simple example, the y-coordinates don't matter.
- Actually, neither do the exact positions of the x-coordinates…

The Alias Method for Discrete Distributions

- Distribution:

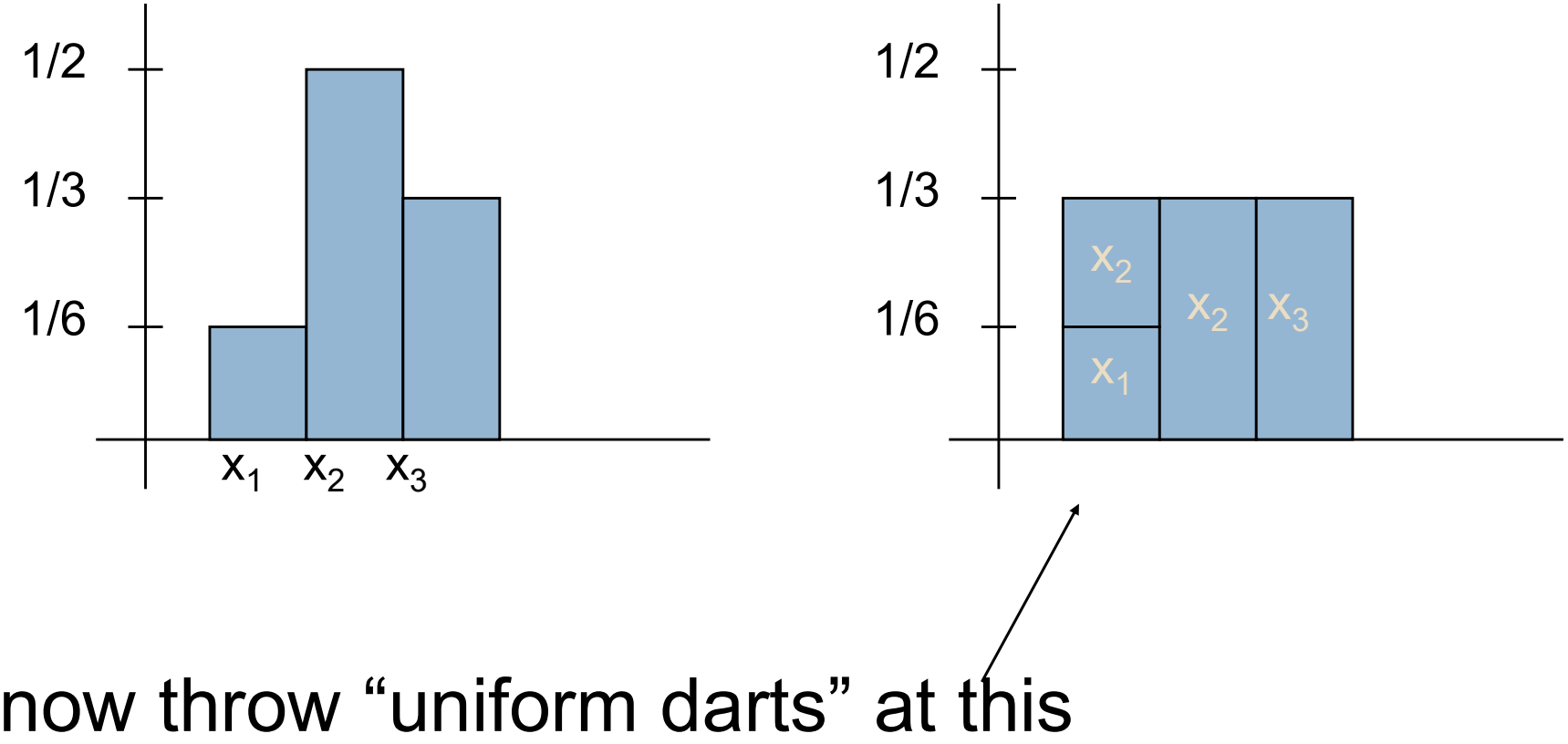
x	x_1	x_2	…	x_n
$P(X=x)$	p_1	p_2	…	p_n

- For non-uniform p's, the idea is to "shift" probability mass around so as to make a rectangular region with the vertical regions partitioned up into subregions

and then to use the previous idea….

The Alias Method for Discrete Distributions

Example:



now throw “uniform darts” at this

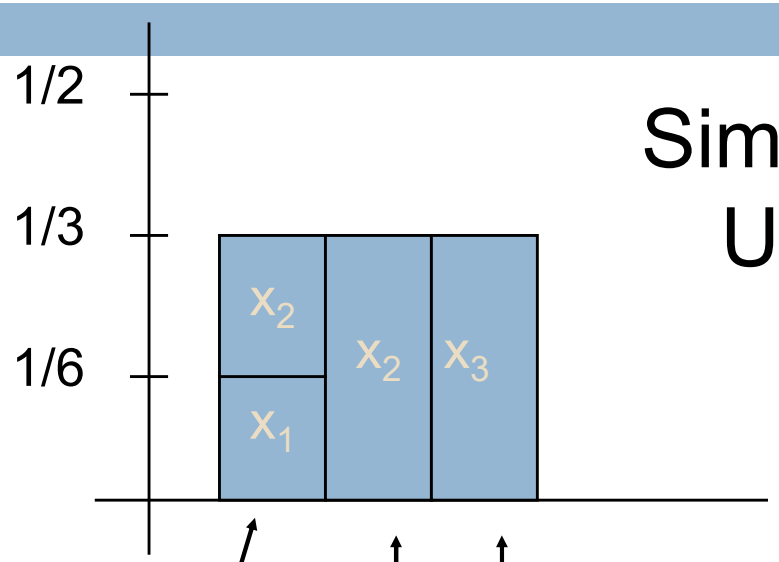
The Alias Method for Discrete Distributions

- The vertical region never needs to be chopped up into more than two pieces.
  - ✓ one part is associated with the original x_i
  - ✓ the other part is associated with an alternative value x_j for some $j \neq i$.

The alternate value is called the alias value.

The Alias Method for Discrete Distributions

Example:



Simulate
U~unif{1,2,3}

If it's in here, flip a coin to decide x_1 versus x_2.

If it's in here, return x_2.

If it's in here, return x_3.

The Alias Method for Discrete Distributions

- In order to do this in an organized and efficient way, we need to set up an "alias table"

i	x_i	p_i	a_i	r_i
1	x_1	1/6	x_2	0.5
2	x_2	1/2	x_2	1
3	x_3	1/3	x_3	1

$$r_i = \frac{\text{height of lower part of rectangle i}}{\text{height of rectangle i}}$$

The Alias Method for Discrete Distributions

i	x_i	p_i	a_i	r_i
1	x_1	1/6	x_2	0.5
2	x_2	1/2	x_2	1
3	x_3	1/3	x_3	1

- This is a pain to do and is time consuming…

but after this one-time computational cost, simulation of a value takes one uniform, one comparison, and at most two memory references!

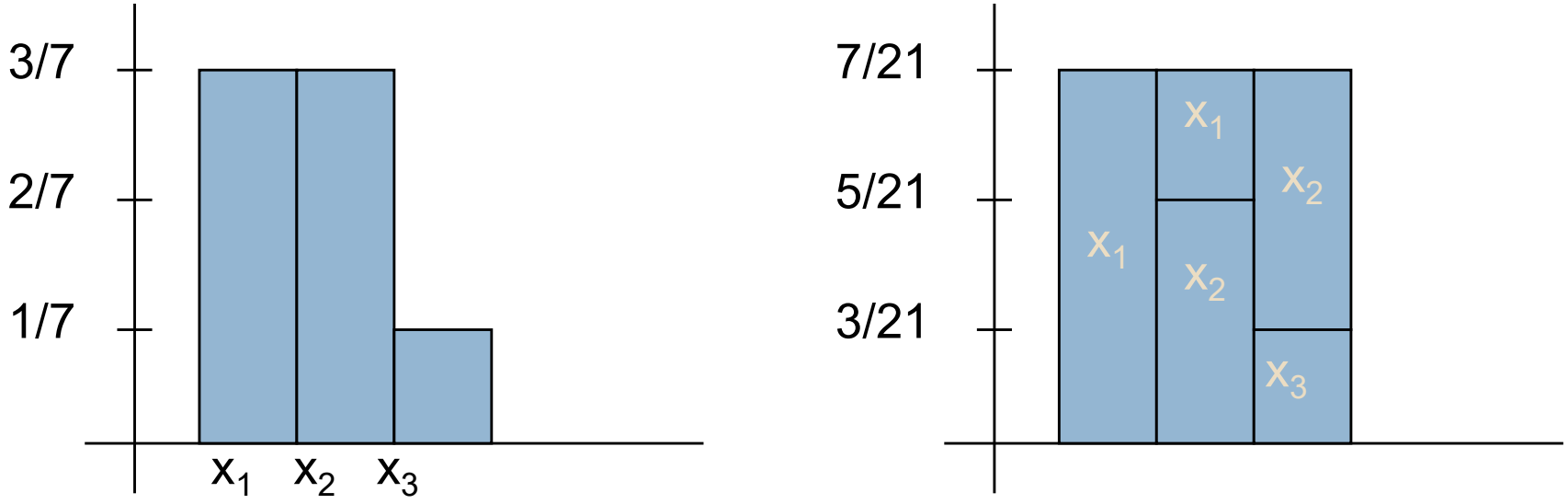
The Alias Method for Discrete Distributions

- Once you have the alias table, the algorithm is:

1. Generate $U_1, U_2 \overset{\text{iid}}{\sim} \text{unif}(0,1)$
2. Set $I = \lceil nU_1 \rceil$.
3. If $U_2 \leq r_I$, then return x_I.
   Otherwise, return a_I.

The Alias Method for Discrete Distributions

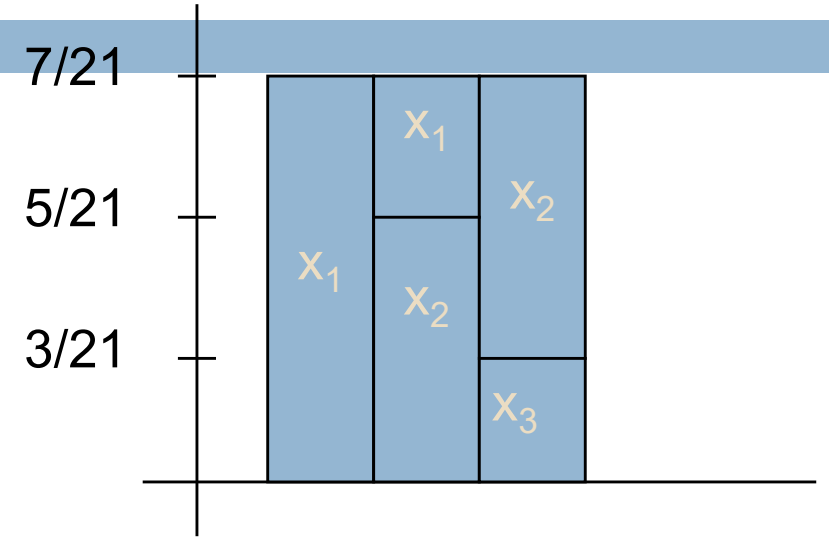
Another Example:



now throw “uniform darts” at this

The Alias Method for Discrete Distributions

- The “alias table” for



i	x_i	p_i	a_i	r_i
1	x_1	3/7	x_1	1
2	x_2	3/7	x_1	5/7
3	x_3	1/7	x_2	3/7

The Alias Method for Discrete Distributions

- One way to set up an alias table:
  - ✓ Set up vectors x, p, r and a each of length n.
    - Fill the vector x with the values $x_1, x_2, \ldots, x_n$
    - Fill the vector p with the values $p_1, p_2, \ldots, p_n$
    - Initialize the vector r with the values

      r[i]=np[i]

    - Initialize the vector a with the values

      a[i]=x[i]

The Alias Method for Discrete Distributions

- Define sets H and L ("high" and "low") to hold indices so that
  -
  -
- If H is empty, stop everything. This is a uniform distribution!
- If H is not empty…

The Alias Method for Discrete Distributions

- select an index j from L and an index k from H
  - Set a[j] = k
  - Set r[k] = r[k]+r[j]-1
- If
  - r[k]<1, add k to L
  - r[k]≥1, remove k from H
- remove j from L

If H is empty, stop, otherwise, return to

Mètode Àlies

1. Generar u=U(0,1), siguin y=1+[ku}, z=frac(ku)
2. Si z<=Q(y), llavors k=y. Si z>Q(y), llavors k=A(y). Es pren $x=x_k$.
3. Cal determinar els valors de Q(i) i els alies A(i), de forma que es tingui:

$$p_i = \frac{Q(i)}{k} + \sum_{j \diagup \Lambda(j)-i} \frac{1-Q(j)}{k}$$

Mètode Àlies

- Fase de pre procés

1. Per cada i=1,..,k es defineix
   - Q(i)=1,$a_i=p_i$, l_i=cert.
2. Es repeteixen les següents operacions com a molt k-1 vegades.
   1. Seleccionar i tal que $a_i<1/k$, l_i= cert. Si això no es possible finalitzem.
   2. Seleccionar k tal que $a_i>1/k$, l_i=cert
   3. Fer l_i= fals, A(i)=j, Q(i)=ka_i, $a_i=a_i-(1-ka_i)/k$

Mètode Àlies

- Exemple

$$X \equiv B(3,\frac{1}{3}), P(X=x) = \binom{3}{x}\left(\frac{1}{3}\right)^{x}\left(\frac{2}{3}\right)^{3-x} \; amb \; x = 0,1,2,3.$$

- El vector de probabilitats es
  - (8/27, 12/27, 6/27,1/27)

Mètode Àlies

I	Q(i)	I_i	a_i	A(i)
1	1	Cert	8/27	-
2	1	Cert	12/27	-
3	1	Cert	6/27	-
4	1	Cert	1/27	-

Mètode Àlies

I	Q(i)	I_i	a_i	A(i)
1	1	Cert	0,083	-
2	1	Cert	12/27	-
3	1	Cert	6/27	-
4	4/27	Fals	1/27	1

Mètode Àlies

I	Q(i)	I_i	a_i	A(i)
1	1	Cert	0,083	-
2	1	Cert	0,416	-
3	24/27	Fals	6/27	2
4	4/27	Fals	1/27	1

Mètode Àlies

I	Q(i)	I_i	a_i	A(i)
1	1/3	Fals	0,083	2
2	1	Cert	1/4	-
3	24/27	Fals	6/27	2
4	4/27	Fals	1/27	1

Exemples de GVA

Alguns mètodes per generar les variables aleatòries més comunes.

Uniforme U(a,b)

- Mètode de la transformada inversa
  - $U \in [0,1)$
  - $X = a + (b - a)u$

Exponencial Exp(λ)

- Mètode de la transformada inversa.
- u∈[0,1)

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

Normal N(μ,σ)

- x segueix una normal N(μ,σ)
  - $x = \mu + \sigma \cdot (u_1 + \ldots + u_{12} - 6)$
- On u_k , $k = 1,\ldots,12$ son variables independents U(0,1).

Normal N(μ,σ): Box-Muller

- Un mètode simple es el de Box-Muller basat en una transformació polar:
- Si u_1 i u_2 son variables aleatòries U(0,1) independents, llavors
  - $x_1 = (-2*\ln(u_1))^{½}*\cos(2\pi*u_2)$
  - $x_2 = (-2*\ln(u_2))^{½}*\cos(2\pi*u_1)$
- *Llavors* x_1 *i* x_2 Son variables aleatòries N(0,1) independents.

Normal N(μ,σ): Acceptació rebuig

- Una versió eficient d'aquesta transformació ve donada per el següent algoritme d'acceptació-rebuig:
- Generador: Mètode polar de la transformació Box-Muller (acceptació - rebuig)
- Sigui $V1 = 2U1 - 1$, $V2 = 2U2 - 1$, $W = (V1)2 + (V2)2$.
  - Si $(W < 1)$
  - $Y = (-2\ln(W)/W)^{1/2}$
  - $X1 = \mu + \sigma V1Y$, $X2 = \mu + \sigma V2Y$
  - sinó
  - Tornar al pas 1

Bernoulli B(1,p) i Binomial B(N,p)

- Com s'ha vist abans una Bernoulli de paràmetre *p* es pot generar de una U(0,1 seguint el següent algoritme:
  - Si U < p , retornar 1.
  - sinó retornar 0.
- Per generar una Binomial B(N,p) s'usa el fet que es la suma de N variables Bernoulli B(1,p) independents
  - X = 0
  - Per (i = 1 to N) fer
    - {X = X +B(1,p)}
  - Retornar X

Geométrica GEOM(1,p)

- Una distribució geomètrica proporciona el temps fins que s'aconsegueix el primer èxit en una seqüència d'experiments de Bernoulli.

- Generar GEOM(1,p):
  - $a = 1/\ln(1-p)$
  - Retornar $X = 1 + \text{floor}[a \times \ln(U)]$

Binomial negativa GEOM (N,p)

- Una variable aleatòria binomial negativa pot ser calculada com la suma de variables geomètriques independents.

- Generar GEOM(N,p):
  - X = 0
  - For(i = 1 to N) {Set X = X +GEOM(1,p)}
  - Retornar X

Poisson (λ)

- El mètode usat es basa en el següent fet: Si els esdeveniments ocorren de forma aleatòria e el temps amb una taxa *r* i *X* es el nombre d'esdeveniments que tenen lloc en un període *t*, llavors *X* és una distribució de Poisson amb paràmetre $\lambda = r \cdot t$.
- Per tant:
- $X = \min\{n: U1 \cdot U2 \cdot \ldots \cdot Un < exp(-\lambda)\} - 1$
- o equivalentment
- $X = \max\{n: U1 \cdot U2 \cdot \ldots \cdot Un \geq exp(-\lambda)\}$
- Generador:
  - Set a = exp(-λ), r = 1, X = -1
  - Mentre > a) {Set r = r·U, X = X + 1)}
  - Retornar X

Filosofia

Pensem una mica sobre tot plegat

Simulism

- http://www.simulation-argument.com/
  - http://simulation-argument.com/simulation.html