Performance Evaluation in Networks

Simulation & Random generation

Study of a system performances

Investigation tools :

- Mathematical/numerical analysis of models
- Simulation of the system (math model, scale model)
- Experiments/measures on real system

Simulation

Simulation

Imitation of a real system, based on a model of the reality which picks some key features of the structure and the dynamics of the system

Numerical/computer/*in silico* simulation

Experiments where the real system is replaced by a computer program implementing a mathematical model of the system.

Ingredients of simulation

- **Models :** deterministic or not, probabilistic or not, states and time discrete or continuous, various specification languages.
- **Software :** many softwares commercial or not, various programming languages.
- **Hardware :** from general monoprocessor to high performance computing with super-computers or clusters.

Advantages of simulation

In the framework of performance evaluation :

- Good alternative to study systems complex to observe or analyse.
- Often cheaper than experiments on real systems.
- Possibility to accelerate time : simulation time vs real time.
- Experimentation under control, possibility to play easily with model parameters

Risks of simulation

In the framework of performance evaluation :

- Bugs in the simulator.
- Bugs in the model
- Errors/difficulties to interpret the results of simulation

Relevance of results sometimes guaranteed by theorems : e.g. a.s. convergence of empirical means towards parameters of the asymptotic behavior

Random Number Generators

Question : how to generate numbers (boolean, integers, rationals, reals) following fixed laws and using a fixed source of randomness ?

Postulate (random generator)

We have a function `Random` *with values in* $[0,1]$ *such that*

1. *One call to* `Random` *returns a r.v. of uniform law over* $[0,1]$.
2. *Successive calls to* `Random` *return independent r.v.*

Remark : `Random` can take any value in $[0,1]$ but $\forall a \in [0,1]$, $\mathbb{P}[\texttt{Random} = a] = 0$. May seem difficult to achieve in practice, but ways to approach this.

Simulate a probability law via Random

Definition

An algo simulates a proba law when one of the variables outputs follow this law, assuming that the successive calls to the random generator are a sequence of i.i.d. r.v. of uniform law over $[0,1]$.

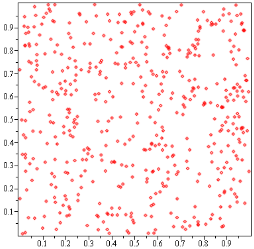
Proposition (Simulation of the uniform law over $[0,1]^d$)

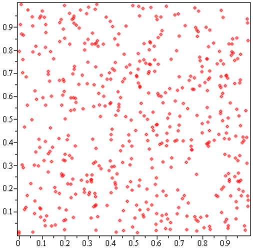
For all $d \in \mathbb{N}^*$, let $(\mathtt{Rand_1}, \ldots, \mathtt{Rand_d})$ d be the successive calls to `Random`*, then for any box $D =]a_1, b_1] \times \cdots \times]a_d, b_d]$, $0 \le a_i < b_i \le 1$, $i = 1, \ldots, d$, we have :*

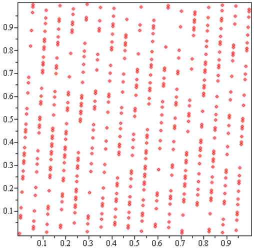
$$\mathbb{P}[(\mathtt{Rand_1}, \ldots, \mathtt{Rand_d}) \in D] = (b_1 - a_1) \cdots (b_d - a_d)$$

Randomness at first sight

Question for 500 pts : which one is i.i.d. uniform in $[0,1]^2$?

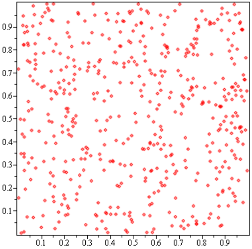




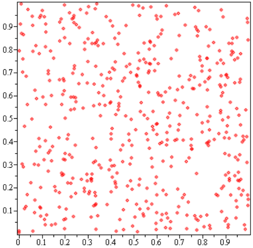


Randomness at first sight

Question for 500 pts : which one is i.i.d. uniform in $[0,1]^2$?

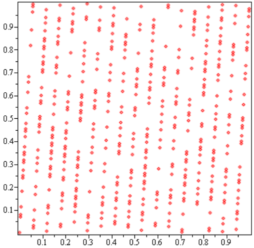


atmospheric noise



(u_{2n}, u_{2n+1}) with $u_n = x_n/2^{31}$,

$x_{n+1} = 65539 x_n \mod 2^{31}$, $x_0 = 131$



(u_{2n}, u_{2n+1}) with $u_n = x_n/250$,

$x_{n+1} = 17 x_n + 1 \mod 250$, $x_0 = 1$

First examples

Vocabulary : simuler une loi / échantillonner selon une loi / random sampling

Example 1 : uniform law over $[a, b]$, $a, b \in \mathbb{R}$.

Example 2 : non biaised dice with 6 faces.

First examples

Vocabulary : simuler une loi / échantillonner selon une loi / random sampling

Example 1 : uniform law over $[a,b]$, $a,b \in \mathbb{R}$.

$$X \leftarrow (b-a) \times \texttt{Random} + a$$

Example 2 : non biaised dice with 6 faces.

First examples

Vocabulary : simuler une loi / échantillonner selon une loi / random sampling

Example 1 : uniform law over $[a,b]$, $a,b \in \mathbb{R}$.

$$X \leftarrow (b-a) \times \texttt{Random} + a$$

Example 2 : non biaised dice with 6 faces.

$$X \leftarrow \lceil 6 \times \texttt{Random} \rceil$$

Sampling by inversion (I)

sampling by inversion is for non-decreasing function

Definition (pseudo-inverse of F)

$\forall u \in \mathbb{R}, F^{(-1)}(u) \stackrel{\text{def}}{=} \inf\{x \in \mathbb{R} | F(x) \geq u\}$ $(= F^{-1}(u)$ if F cont strict$\nearrow)$.

Proposition

Let X real r.v. with cumulative distribution F and U of uniform law over $[0,1]$, then $\widetilde{X} = F^{(-1)}(U)$ follows the law of X.

Algorithm (sampling by inversion)

$$X \leftarrow F^{(-1)}(\mathtt{Random})$$

Use : useful when one has an explicit expression and/or a simple computation of $F^{(-1)}$.

For sampling by inversion, the easiest way to work on the proof is if we assume that F is contiuous. But in general we should

Sampling by inversion (II)

Exemple 1 : simulate the law exp $F(x) = (1 - e^{-\lambda x})\mathbb{1}_{\mathbb{R}_+}(x)$

Exemple 2 : simulate a discrete law with values in $\{x_1 < x_2 < x_3 < \dots\}$ a finite or countable sets.

Sampling by inversion (II)

Exemple 1 : simulate the law exp $F(x) = (1 - e^{-\lambda x})\mathbb{1}_{\mathbb{R}_+}(x)$
Formula : $\forall u \in]0,1],\ F^{-1}(u) = -\ln(1-u)/\lambda$
Algo : $X \leftarrow -\ln(\texttt{Random})/\lambda$ (not necessary to compute $-\ln(1-\texttt{Random})/\lambda$ since $\texttt{Random}$ and $1-\texttt{Random}$ have the same law)

Exemple 2 : simulate a discrete law with values in $\{x_1 < x_2 < x_3 < \dots\}$ a finite or countable sets.

Sampling by inversion (II)

Exemple 1 : simulate the law exp $F(x) = (1 - e^{-\lambda x})\mathbb{1}_{\mathbb{R}_+}(x)$
Formula : $\forall u \in]0,1],\ F^{-1}(u) = -\ln(1-u)/\lambda$
Algo : $X \leftarrow -\ln(\texttt{Random})/\lambda$ (not necessary to compute $-\ln(1-\texttt{Random})/\lambda$ since $\texttt{Random}$ and $1-\texttt{Random}$ have the same law)

Exemple 2 : simulate a discrete law with values in $\{x_1 < x_2 < x_3 < \dots\}$ a finite or countable sets.

Formulas : $\forall x \in \mathbb{R},\ F(x) = \begin{cases} 0 & \text{si } x < x_i \\ p_1 + \dots + p_i & \text{si } x_i \leq x < x_{i+1} \end{cases}$

$\forall u \in]0,1],\ F^{-1}(u) = \{x_i | F_{i-1} < u \leq F_i\}$ où $F_i = p_1 + \dots + p_i$

Algo :

$i \leftarrow 1,\ choix \leftarrow \texttt{Random},$ repeat $(choix > F_i)$ until $i \leftarrow i+1,$ $X \leftarrow x_i$

Sampling by inversion (III)

Example 2bis : Poisson law of parameter λ

Algo :

$P \leftarrow e^{-\lambda}$, $F \leftarrow P$, $X \leftarrow 0$, $choix \leftarrow \texttt{Random}$,
while $(choix > F)$ do $\{ X \leftarrow X+1,\ P \leftarrow \lambda P/X,\ F \leftarrow F+P \}$

Exemple 2ter : Discrete law over n values

Algo :

precompute F_i, $1 \leq i \leq n$ in a sorted array,
find index i such that $F_{i-1} < \texttt{Random} \leq F_i$ by dichotomy

space	precomputation time	sampling time	calls to Random
$O(1)$	0	$O(n)$	1
$O(n)$	$O(n)$	$O(\log(n))$	1
$O(?)$	$O(?)$	$O(1)$	1

- In the first method, we compute Fi one by one, so we don't need to store anything- In the second, we compte Fi

Sampling by conditional rejection (I)

Ingredients :

- Algo $\mathscr{A}$ probabilistic (with calls to `Random`) where output variable X = v.a. $\Omega \to \mathbb{R}$.
- Event $F \subseteq \Omega$ of probability $\mathbb{P}(F) > 0$.
- Algo $\widetilde{\mathscr{A}}$: repeat $\mathscr{A}$ until F occurs/realized.

Proposition

1. *When $\widetilde{\mathscr{A}}$ stop, $\tilde{X}$ the output variable of $\mathscr{A}$ follows the law of X conditioned by F, i.e. with cumulative distribution $\mathbb{P}(X \leq x|F) = \mathbb{P}(\{X \leq x\} \cap F)/\mathbb{P}(F)$.*
2. *Number of loop executions in $\widetilde{\mathscr{A}}$ follows a geometrical law of parameter $\mathbb{P}(F)$.*

Sampling by conditional rejection (II)

Example1 : let $\alpha \in]0,1[$, repeat $X \leftarrow \texttt{Random}$ until $X \leq \alpha$.

Example2 : let D and D' two measurable areas in $\mathbb{R}^d$ such that $D \subseteq D'$ et $0 < vol(D) \leq vol(D') < +\infty$, suppose that you know how to sample the uniform law in D'.

Algo : repeat draw a random point X in D' until $X \in D$.

Sampling by conditional rejection (II)

Example1 : let $\alpha \in]0,1[$, repeat $X \leftarrow \texttt{Random}$ until $X \leq \alpha$.

$$F_{\tilde{X}}(x) = \mathbb{P}(X \leq x | X \leq \alpha) = \begin{cases} 0 & \text{si } x < 0 \\ x/\alpha & \text{si } x \in [0,\alpha] \\ 1 & \text{si } x > \alpha \end{cases}$$

Densité $f_{\tilde{X}}(x) = \frac{1}{\alpha}\mathbb{1}_{[0,\alpha]}(x)$ càd uniforme sur $[0,\alpha]$

Comparer avec l'algo : $X \leftarrow \alpha \times \texttt{Random}$.

Compare to this algorithm (that we have seen before), t

Example2 : let D and D' two measurable areas in $\mathbb{R}^d$ such that $D \subseteq D'$ et $0 < vol(D) \leq vol(D') < +\infty$, suppose that you know how to sample the uniform law in D'.

Algo : repeat draw a random point X in D' until $X \in D$.

Sampling by conditional rejection (III)

Example2 : repeat draw a random point X in D' until $X \in D$.

Proposition

If $D \subseteq D'$ measurable areas in $\mathbb{R}^d$ où $0 < vol(D) \leq vol(D') < \infty$, and X r.v. in $\mathbb{R}^d$ of uniform law over D', then the conditional law of X given $X \in D$ is the uniform law over D.

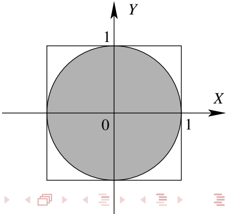
Time complexity : r.v. with geometric law of $\frac{vol(D)}{vol(D')}$ $\rightarrow$ choose D' close to D and where uniform law is simple to simulate, e.g. union of disjoint boxes.

Uniform law over the unit disk

$D = \{(x,y) | x^2 + y^2 \leq 1\}$ **via** $D' = [-1,1]^2$

Repeat $X \leftarrow 2 \times \texttt{Random} - 1$
$\quad Y \leftarrow 2 \times \texttt{Random} - 1$

until $X^2 + Y^2 \leq 1$



Sampling by conditional rejection (IV)

Example 3 : Rejection method by von Neumann (1951)

Proposition

Let f, g proba densities over $\mathbb{R}^d$ with constant c such that $\forall x \in \mathbb{R}^d$, $f(x) \le cg(x)$, and X r.v. over $\mathbb{R}^d$ of density g, and U real r.v. of uniform law over $[0,1]$, independent of X, then the conditional law of X given "$cUg(X) < f(X)$ has density f.

Algo : repeat $U \leftarrow \texttt{Random}$, $X \leftarrow$ sample for density g
until $cUg(X) < f(X)$

we want to sample f, using g (that is small and we kno

Example : $f(x) = \frac{2}{\pi}\sqrt{1-x^2}\mathbb{1}_{[-1,1]}(x)$ (abscisse pt unif in disk)

Choose e.g. $g(x) = \frac{1}{2}\mathbb{1}_{[-1,1]}(x)$, and adjust small $c = \frac{4}{\pi}$

U is a rv that is only use

Algo : repeat $\begin{cases} U \leftarrow \texttt{Random} \\ X \leftarrow 2 \times \texttt{Random} - 1 \end{cases}$ until $\underbrace{\left(\frac{4}{\pi} U \frac{1}{2} < \frac{2}{\pi}\sqrt{1-\boxed{X^2}}\right)}_{U^2 \le 1 - X^2}$

X: we sample random numbers in [-1,1]

Sampling by decomposition (I)

Ingredients : f proba density over $\mathbb{R}^d$ which can be written $f = \sum_{n\in\mathbb{N}} p_n f_n$ where $(p_n)_{n\in\mathbb{N}}$ mass over $\mathbb{N}$ and $\forall n \in \mathbb{N}$, f_n density over $\mathbb{R}^d$.

the formula with sum just means that f can be decomposed i

Proposition

Let $(X_n)_{n\in\mathbb{N}}$ family of r.v. over $\mathbb{R}^d$ with respective densities f_n, and N r.v. over $\mathbb{N}$ with mass $(p_n)_{n\in\mathbb{N}}$ where N independent from $(X_n)_{n\in\mathbb{N}}$, then $\widetilde{X} = X_N$ has density f.

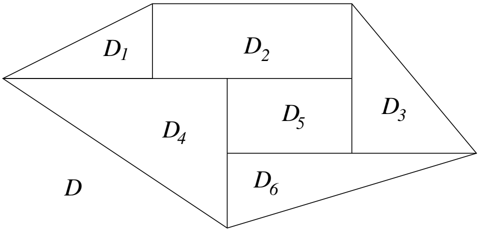
Algorithm

Draw $n \in \mathbb{N}$ with proba $(p_n)_{n\in\mathbb{N}}$, then draw $X \in \mathbb{R}^d$ with density f_n.

Suppose that for each fn, we know how to sample. Then one

Sapling by decomposition (II)

Example : Uniforme law over $D = \bigcup_{i=1}^{n} D_i$ disjoint and measurable in $\subseteq \mathbb{R}^d$



Sapling by decomposition (II)

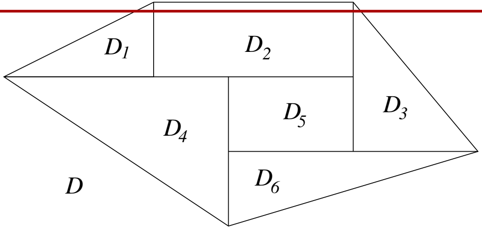
Example : Uniforme law over $D = \bigcup_{i=1}^{n} D_i$ disjoint and measurable in $\subseteq \mathbb{R}^d$

Formula :

$$\frac{1}{vol(D)} \mathbb{1}_D(x) = \sum_{i=1}^{n} \frac{vol(D_i)}{vol(D)} [\frac{1}{vol(D_i)} \mathbb{1}_{D_i}(x)]$$

Algo : draw i avec proba $\frac{vol(D_i)}{vol(D)}$, then draw X at random over D_i.

an explanation: suppose that we have an area D that can be decomposed into some Di's. TO sampling, we choose one of



Sampling by change of variables (I)

Reminder : let φ bijection $(x_1,x_2) \mapsto (y_1,y_2)$ between $D \subseteq \mathbb{R}^2$ and $D' \subseteq \mathbb{R}^2$, with (abusive) notations $y_1 = y_1(x_1,x_2), y_2 = y_2(x_1,x_2)$ for φ, and $x_1 = x_1(y_1,y_2), x_2 = x_2(y_1,y_2)$ for φ^{-1}. Assuming the existence of partial deriv, one define the Jacobien of φ^{-1} as :

$$J_{\varphi^{-1}}(y_1,y_2) = \begin{vmatrix} \frac{\partial x_1}{\partial y_1} & \frac{\partial x_2}{\partial y_1} \\ \frac{\partial x_1}{\partial y_2} & \frac{\partial x_2}{\partial y_2} \end{vmatrix} = \frac{\partial x_1}{\partial y_1}\frac{\partial x_2}{\partial y_2} - \frac{\partial x_1}{\partial y_2}\frac{\partial x_2}{\partial y_1}$$

Theorem (integration & change of variables)

Let $f : \mathbb{R}^2 \to \mathbb{R}$ integrable, $\varphi : D \to D'$ bijection and $A \subseteq \mathbb{R}^2$, then

$$\iint_A f(x_1,x_2)dx_1 dx_2 = \iint_{\varphi(A)} f(x_1(y_1,y_2),x_2(y_1,y_2))|J_{\varphi^{-1}}(y_1,y_2)|dy_1 dy_2$$

Sampling by change of variables (II)

Corollary

Let (X_1,X_2) r.v. of continuous joint distrib f, of support $D \in \mathbb{R}^2$, and φ bijection $D \to D'$, then $(Y_1,Y_2) = \varphi((X_1,X_2))$ has a continous joint distrib :

$$f_{Y_1,Y_2}(y_1,y_2) = \begin{cases} f(x_1(y_1,y_2), x_2(y_1,y_2))|J_{\varphi^{-1}}(y_1,y_2)| & si\ (y_1,y_2) \in D' \\ 0 & sinon \end{cases}$$

Example : Box-Muller algorithm (1958)

$$\begin{cases} R \leftarrow \sqrt{-2\ln(\texttt{Random})},\ \Theta \leftarrow 2\pi \times \texttt{Random} \\ X \leftarrow R\cos\Theta,\ Y \leftarrow R\sin\Theta \end{cases}$$

explanation: - we draw to random values, R and theta, then we do "change of variables" (notice that here we go from polar

Sampling by change of variables (II)

Corollary

Let (X_1,X_2) r.v. of continuous joint distrib f, of support $D \in \mathbb{R}^2$, and φ bijection $D \to D'$, then $(Y_1,Y_2) = \varphi((X_1,X_2))$ has a continous joint distrib :

$$f_{Y_1,Y_2}(y_1,y_2) = \begin{cases} f(x_1(y_1,y_2),x_2(y_1,y_2))|J_{\varphi^{-1}}(y_1,y_2)| & si\ (y_1,y_2) \in D' \\ 0 & sinon \end{cases}$$

Example : Box-Muller algorithm (1958)

$$\begin{cases} R \leftarrow \sqrt{-2\ln(\texttt{Random})},\ \Theta \leftarrow 2\pi \times \texttt{Random} \\ X \leftarrow R\cos\Theta,\ Y \leftarrow R\sin\Theta \end{cases}$$

$\Rightarrow$ Θ of uniform law over $[0,2\pi]$, indep from R of density $re^{-\frac{r^2}{2}}\mathbb{1}_{\mathbb{R}_+}(r)$ $\Rightarrow$ X et Y are independent, of normal law $\mathcal{N}(0,1)$

Generators : true randomness vs pseudo-randomness

Question : how to implement `Random` ?

Generators : true randomness vs pseudo-randomness

Question : how to implement `Random` ?

→ how to generate sequences of random numbers/bits

Generators : true randomness vs pseudo-randomness

Question : how to implement `Random` ?
$\rightarrow$ how to generate sequences of random numbers/bits

Pseudo-random generator

Deterministic algo generating a sequence of numbers, with some parameters to fix, often defined as $x(n+1) = f(x(n))$ with a "seed" $x(0)$, predictable e.g. if knowledge of initial parameters.

"true" random generator

Sequence obtained by physical measures of phenomena with intrinsec probabilities (e.g. quantum effects) or complex behaviors (e.g. chaotic).

Random sequences

Question : what is a truly random sequence of numbers ? how to evaluate randomness of a sequence of numbers ?

Random sequences : statistical test

example of statistical tests to measure the uniformity of the results of our random generator

Definition (d-uniform real sequences)

A sequence $(x_n)_{n\in\mathbb{N}}$ with values in $[0,1]$ is d-uniform if for any box $D =]a_1, b_1] \times \cdots \times]a_d, b_d]$, we have :

in this formula, I count all the points in the box D

$$\lim_{n\to+\infty} \frac{1}{n} \sum_{i=0}^{n-1} \mathbb{1}_D((x_{di}, x_{di+1}, \ldots, x_{d(i+1)-1})) = (b_1 - a_1) \cdots (b_d - a_d)$$

Definition (d-uniform boolean sequences)

A sequence $(x_n)_{n\in\mathbb{N}}$ with values in $\{0,1\}$ is d-uniforme if for any pattern $(\varepsilon_1, \ldots, \varepsilon_d) \in \{0,1\}^d$, we have :

$$\lim_{n\to+\infty} \frac{1}{n} \sum_{i=0}^{n-1} \mathbb{1}_{(\varepsilon_1,\ldots,\varepsilon_d)}((x_{di}, x_{di+1}, \ldots, x_{d(i+1)-1})) = \frac{1}{2^d}$$

statistical tests do not give the best criteria for randomness

Random sequences : statistical tests

Exemple 1 :

$$000000000000000000000000\ldots$$

Exemple 2 :

$$010101010101010101010101\ldots$$

Exemple 3 :

$$000110110001101100011 0\ldots$$

Exemple 4 : Champernowne sequence(1933)

$$\underbrace{0}_{0}\underbrace{1}_{1}\underbrace{10}_{2}\underbrace{11}_{3}\underbrace{100}_{4}\underbrace{101}_{5}\underbrace{110}_{6}\underbrace{111}_{7}\underbrace{1000}_{8}\ldots$$

Random sequences : statistical tests

Exemple 1 : not 1-uniform

$$000000000000000000000\ldots$$

Exemple 2 :

$$010101010101010101010\ldots$$

Exemple 3 :

$$000110110001101100011 0\ldots$$

Exemple 4 : Champernowne sequence(1933)

$$\underbrace{0}_{0}\underbrace{1}_{1}\underbrace{10}_{2}\underbrace{11}_{3}\underbrace{100}_{4}\underbrace{101}_{5}\underbrace{110}_{6}\underbrace{111}_{7}\underbrace{1000}_{8}\ldots$$

Random sequences : statistical tests

Exemple 1 : not 1-uniform

$$000000000000000000000\ldots$$

Exemple 2 : 1-uniform

$$010101010101010101010\ldots$$

Exemple 3 :

$$000110110001101100011 0\ldots$$

Exemple 4 : Champernowne sequence(1933)

$$\underbrace{0}_{0}\underbrace{1}_{1}\underbrace{10}_{2}\underbrace{11}_{3}\underbrace{100}_{4}\underbrace{101}_{5}\underbrace{110}_{6}\underbrace{111}_{7}\underbrace{1000}_{8}\ldots$$

Random sequences : statistical tests

Exemple 1 : not 1-uniform

$$000000000000000000000000\ldots$$

Exemple 2 : 1-uniform

$$010101010101010101010101\ldots$$

Exemple 3 : 2-uniform

$$\underbrace{00}\,\underbrace{01}\,\underbrace{10}\,\underbrace{11}\,\underbrace{00}\,\underbrace{01}\,\underbrace{10}\,\underbrace{11}\,\underbrace{00}\,\underbrace{01}\,\underbrace{10}\ldots$$

Exemple 4 : Champernowne sequence(1933)

$$\underbrace{0}_{0}\,\underbrace{1}_{1}\,\underbrace{10}_{2}\,\underbrace{11}_{3}\,\underbrace{100}_{4}\,\underbrace{101}_{5}\,\underbrace{110}_{6}\,\underbrace{111}_{7}\,\underbrace{1000}_{8}\ldots$$

Random sequences : statistical tests

Exemple 1 : not 1-uniform

because the frequency of 11...1 is 0. If we want it to be 1-uniform, the f

$$0000000000000000000000\ldots$$

Exemple 2 : 1-uniform

this is not 2-uniform

$$0101010101010101010101\ldots$$

Exemple 3 : 2-uniform

2-uniform because all pattern of size 2 appear in frequency which tends to 4but it

$$\underbrace{00}\ \underbrace{01}\ \underbrace{10}\ \underbrace{11}\ \underbrace{00}\ \underbrace{01}\ \underbrace{10}\ \underbrace{11}\ \underbrace{00}\ \underbrace{01}\ \underbrace{10}\ldots$$

Exemple 4 : Champernowne sequence(1933)

d-uniform, for any d

$$\underbrace{0}_{0}\ \underbrace{1}_{1}\ \underbrace{10}_{2}\ \underbrace{11}_{3}\ \underbrace{100}_{4}\underbrace{101}_{5}\underbrace{110}_{6}\underbrace{111}_{7}\underbrace{1000}_{8}\ldots$$

∞-uniform, but simple to compute $\rightarrow$ limits of d-uniformity to define randomness. Many other statistical criteria, but with the same limits.

Random sequences : Kolmogorov complexity

this is one way (also classical way) to define formally the notion of "randomness"

Algorithm : function ϕ from $\{0,1\}^*$ to $\{0,1\}^*$, encoded with $|\phi|$ bits.

Complexity of word x relatively to algo ϕ

$$K_\phi(x) \stackrel{\text{def}}{=} |\phi| + \inf\{|z|,\ \phi(z) = x\}$$

Kolmogorov complexity of word x

$$K(x) \stackrel{\text{def}}{=} \inf_\phi K_\phi(x)$$

Definition (Random sequence)

A sequence $(x_n)_{n\in\mathbb{N}} \in \{0,1\}^{\mathbb{N}}$ is called random if it exists a constant c such that $\forall n \geq 1,\ K(x_1 \cdots x_n) \geq n - c$.

Random = information can not be compressed, no simple rule of generation (thus "unpredictable")

Random generator : physical methods

Several ways to get random bits :

- dices, coins, cards, loto, marc de café, ...
- quantum phenomena : electronic noise in circuits, radioactivity ...
- other physical phenomena : thermal noise, radio noise, read/write moves of heads in hard disks ...

Selling randomness :

- `RANDOM.ORG` : atmospheric noise measured through radio (www.random.org)
- `HotBits` : measures from a radioactive source (www.fourmilab.ch/hotbits)
- `Intel` : Intel 810, 810E, 810E2 Chipsets
- `The Marsaglia Random Number CDROM` : 4.10^9 random bits mixing several processes (i.cs.hku.hk/~diehard)

Pseudo-random generators : linear congruence

this slide is about: be careful of the choice of the random gerators. An example that is bad is RANDU (given below). Using this r

Linear congruence generators

- Integer parameters : $m > 0,\ a > 0,\ b \geq 0$, seed $0 \leq x_0 < m$.
- Sequence $(u_n)_{n\in\mathbb{N}} \in [0,1]^{\mathbb{N}}$: $x_{n+1} = ax_n + b \mod m,\ u_n = x_n/m$.

⚠ **Beware of the choice of parameters :**

- risk of short periodic behavior
- uniformity sometimes poor
- risk of correlation between successive values

Examples :

- RANDU (IBM 1960) : $x_{n+1} = 65539x_n \mod 2^{31}$, x_0 odd
- MINSTD called "Minimal Standard" (Park, Miller 1988) : $x_{n+1} = 16807x_n \mod 2^{31} - 1$

Pseudo-random generators : linear congruence

Linear congruence generators

- Integer parameters : $m > 0$, $a > 0$, $b \geq 0$, seed $0 \leq x_0 < m$.
- Sequence $(u_n)_{n \in \mathbb{N}} \in [0,1]^{\mathbb{N}}$: $x_{n+1} = ax_n + b \mod m$, $u_n = x_n/m$.

⚠ **Beware of the choice of parameters :**

- risk of short periodic behavior
- uniformity sometimes poor
- risk of correlation between successive values

Examples :

- RANDU (IBM 1960) : $x_{n+1} = 65539 x_n \mod 2^{31}$, x_0 odd ☹
- MINSTD called “Minimal Standard” (Park, Miller 1988) : $x_{n+1} = 16807 x_n \mod 2^{31} - 1$ ☺

Pseudo-random generators : quadratic congruence

BBS Generator (Blum, Blum, Shub 1986)

- Integer parameters : $m = pq$, with p, q prime and $\equiv 3 \mod 4$, x_0 prime with m.
- Sequence $(u_n)_{n\in\mathbb{N}} \in \{0,1\}^{\mathbb{N}}$: $x_{n+1} = x_n^2 \mod m$, $u_n = x_n \mod 2$.

Remark : slow for simulation, but strong for cryptography assuming that factorisation is hard.

Example : $n = 7 \times 19 = 133$

$$\begin{array}{lcl}
x_0 = 100 & \xrightarrow{\text{parité}} & u_0 = 0 \\
x_1 = 100^2 \mod 133 = 25 & \xrightarrow{\text{parité}} & u_1 = 1 \\
x_2 = 25^2 \mod 133 = 93 & \xrightarrow{\text{parité}} & u_2 = 1 \\
x_3 = 93^2 \mod 133 = 4 & \xrightarrow{\text{parité}} & u_3 = 0
\end{array}$$

Different types of computer simulation *for performance evaluation*

Simulations to analyse the dynamics :

- Equational simulation (recurrences)
- Trace simulation
- Discrete event simulation

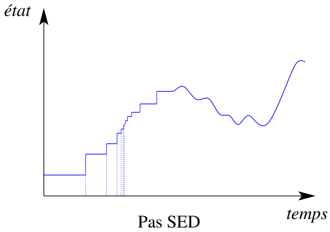
Simulations as algorithms to compute some functions :

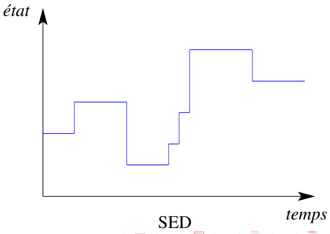
- Monte Carlo simulation (a class of randomized algorithms)
- Sampling using simulation : "to the future" (classical) or "from the past" (coupling from the past)

Discrete event simulation

Definition

- *event/transition/jump : state of the system chainging at some instant.*
- *discrete event system (DES) : dynamics described by a sequence of discrete events (time & space discrete or continuous)..*
- *discrete event simulation : simulation of a DES.*





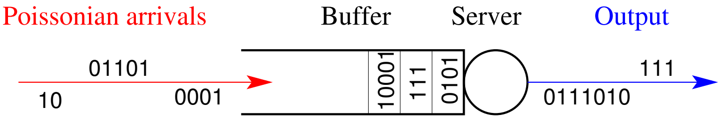
Discrete event simulation

Algorithm (DES Simulation)

1. **Initialisation** *{create the 1st event and insert it in the schedule}*
2. **Repeat until** *some stopping criteria is satisfied*
   - *Move the clock to instant t of next event e ;*
   - *Update variables depending on time t ;*
   - *Execute e {action over the state and insertion/suppression of events in the schedule} ;*
   - *Suppress e from the schedule ;*
3. **Ending** *{compute final statistics and produce final report}*

Schedule : dynamic set of incoming next events

Example : a communication channel in isolation (I)



Model of the channel (continuous time and discrete data) :

- Input traffic : packets of random length (with uniform law over $\{1,\ldots,M\}$) with T_n arrival date of n-th packet following a Poisson process of intensity λ, i.e. $T_0 = 0$ and inter-arrivals $(T_n - T_{n-1})_{n\in\mathbb{N}^*}$ i.i.d. of law $Exp(\lambda)$.
- Server : FIFO with rate $= 1$ if there is work (transmission time of a packet $=$ its length).
- Queue : storage with ∞ memory.

Example : a communication channel in isolation (II)

Variable(s) for system states ? Simulation algo ?

Example : a communication channel in isolation (II)

System state : $X(t)$ = nb o f packets waiting or being transmitted at time t (state space : $\mathbb{N}$).

arrival

departure

Two types of events (“sources”)	Active source if
α : packet arrival	always
β : transmission end of a packet	$X(t) > 0$

departure only happe

Residual times : $Y_\alpha(t)$ (resp. $Y_\beta(t)$) time from t to the first type α event (resp. β).
Set of active sources for state i **:** $Active(i) \subseteq \{\alpha, \beta\}$.

Example : a communication channel in isolation (III)

t is the current time

Algorithm (Simulation of $M(\lambda)/D(1)/1$ queue)

1. $t \leftarrow 0$; $X(t) \leftarrow 0$; $Y_\alpha(t) \leftarrow -\frac{1}{\lambda}\ln(\mathtt{Random})$; the event (arrival/depart) follows exp
2. $V \leftarrow \min_{\gamma \in Active(X(t))} Y_\gamma(t)$; $\overline{\gamma} \leftarrow \arg\min_{\gamma \in Active(X(t))} Y_\gamma(t)$
3. **If** $\overline{\gamma} = \alpha$, $\left[\begin{array}{l} X(t+V) \leftarrow X(t)+1; \\ Y_\alpha(t+V) \leftarrow -\frac{1}{\lambda}\ln(\mathtt{Random}); \\ \textbf{If } X(t+V) > 1, \textbf{ then } Y_\beta(t+V) \leftarrow Y_\beta(t) - V; \\ \textbf{Else } Y_\beta(t+V) \leftarrow \lceil M \times \mathtt{Random} \rceil; \end{array}\right.$

   V is the time from t to the first event (arrival

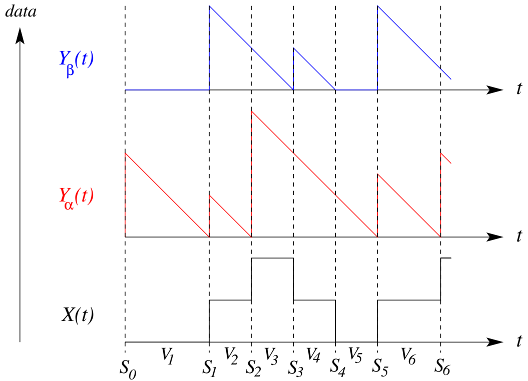
   arrival

   Y_\beta(t+V) : departure timeM x

   If $\overline{\gamma} = \beta$, $\left[\begin{array}{l} X(t+V) \leftarrow X(t)-1; \\ Y_\alpha(t+V) \leftarrow Y_\alpha(t) - V; \\ \textbf{If } X(t+V) > 0, \; Y_\beta(t+V) \leftarrow \lceil M \times \mathtt{Random} \rceil; \end{array}\right.$

   departure
4. $t \leftarrow t + V$; **Goto** 2;

Example : a communication channel in isolation (III)



- $V_n \in \mathbb{R}_+$, $n \geq 1$, consecutive values of V : delay between each state transition ("jump") $\rightarrow S_n = \sum_{i=1}^{n} V_i$ date of n-th jump.

Matthes scheme : ingredients

- E : countable set of system states, $X(t)$ state at time t.
- S : set of sources (inducing state transitions).
- State $x \in E \rightarrow$ active sources : $Active(x) \subseteq S$.
- Source $\alpha \in S \rightarrow Y_\alpha(t)$ delay from t to next event α.
  Computed from :
  - F_α cumulative distrib fct for the "size" of event α.
  - $C(\alpha, x)$ "decrease" speed of $Y_\alpha(t)$ when state is x.
- Jump : when $Y_\alpha(t)$ reaches 0, α occurs and system jumps from current state x to new state y with proba $p(\alpha, x, y)$.

Matthes scheme : simulation algo

Algorithm (Simulation “à la Matthes”)

1. $t \leftarrow 0$; $X(t) \leftarrow x_0$; $Y_\alpha(t) \leftarrow y_{0,\alpha}$, $\forall \alpha \in Active(x_0)$;
2. $V \leftarrow \min\limits_{\alpha \in Active(X(t))} \frac{Y_\alpha(t)}{C(\alpha, X(t))}$; $\overline{\alpha} \leftarrow \arg\min\limits_{\alpha \in Active(X(t))} \frac{Y_\alpha(t)}{C(\alpha, X(t))}$
3. *Draw* x *with law* $(p(\overline{\alpha}, X(t), e))_{e \in E}$;
   $X(t+V) \leftarrow x$;
   $Y_\alpha(t+V) \leftarrow Y_\alpha(t) - V \times C(\alpha, X(t))$,
   $\forall \alpha \in Active(x) \cap Active(X(t)) \setminus \{\overline{\alpha}\}$;
   $Y_\alpha(t+V) \leftarrow F_\alpha^{-1}(\mathtt{Random})$, $\forall \alpha \in Active(x) \setminus Active(X(t))$;
   $Y_{\overline{\alpha}}(t+V) \leftarrow F_{\overline{\alpha}}^{-1}(\mathtt{Random})$, *si* $\overline{\alpha} \in Active(x)$;
4. $t \leftarrow t + V$; **Goto** 2;

Quantify/qualify transitory/asymptotic behaviour

Stationarity / Stability :

- For deterministic system ($F_\alpha = \mathbb{1}_{[T_\alpha,+\infty[}$), si E est ∞, Does $X(t)$ remain if a finite subset of E when $t \to +\infty$?
- For probabilistic system (F_α random), which conditions make $X(t)$ tends to a limit r.v. X_∞ ?

Characterization of processes :

- For deterministic system, $X(t)$ becomes periodic ?
- For probabilistic system, which conditions make $X(t)$ markovian ? and make this Markov chain positive recurrent (probabilistic analog of periodicity) ?

Characterization of laws : what are the laws of $X(t)$ (transitory law) and X_∞ (asymptotic/stationary law) ?

Simulation in pratice

Use of a simulator → estimate behaviour/laws via observations and statistics

- ⚠ Choice of initial conditions ?
- ⚠ Stopping criteria for each simulation ?
- ⚠ Stopping criteria over number of simulation runs ?
- ⚠ Compromise between simulator sharpness / simulation complexity.

to be continued ...