INSA

Année Universitaire 2025-2026

3ème Année Mathématiques Appliquées

Probabilités Projet N° 1 :
Méthodes de Monte Carlo pour la simulation probabiliste

C'est quoi la simulation probabiliste et on fait comment ?

1 La simulation et les méthodes de Monte Carlo

Définition 1.1 (Larousse/Simulation) *Représentation du comportement d'un processus physique, industriel, biologique, économique ou militaire au moyen d'un modèle matériel dont les paramètres et les variables sont les images de ceux du processus étudié. (Les modèles de simulation prennent le plus souvent la forme de programmes d'ordinateurs auxquels sont parfois associés des éléments de calcul analogique).*

Définition 1.2 (Larousse/Simulation numérique) *Une simulation numérique repose toujours sur un modèle, sur la représentation simplifiée d'un objet, d'un système ou d'un processus physique, chimique ou biologique. Les scientifiques élaborent ces modèles en tenant compte des lois et des expériences : ils peuvent représenter les courants marins, la trajectoire d'un avion ou l'étude des flux migratoires à l'aide de chiffres et d'équations mathématiques. Un tel modèle met, en quelque sorte, la réalité en formules.*

Ensuite, ces formules permettent de réaliser un programme informatique. On obtient ainsi une simulation numérique de la réalité qui remplace parfois même l'expérimentation. Ainsi la simulation permet-elle de réduire les coûts et les délais. Les simulations servent au dimensionnement et fournissent des informations difficiles à obtenir par les mesures. On peut simuler une explosion de gaz dans un immeuble, ou les effets d'une arme nucléaire, sans commettre le moindre dégât. Les simulations permettent ensuite d'extrapoler en se projetant en grandeur réelle. Enfin, pour les scientifiques, une simulation numérique rend possible l'élaboration de nouveaux modèles physiques. Il est impossible par exemple d'écrire une théorie qui rendrait compte des mouvements climatiques à l'échelle de la planète. Mais des simulations, qui intègrent les effets d'un grand nombre de phénomènes plus ou moins simples, peuvent approcher cette réalité complexe. Trop de variables entrent en compte pour qu'un homme seul puisse analyser les données météorologiques. Seule une machine peut prévoir le temps.

1. Simulation déterministe : toute la dynamique d'évolution (ou le modèle choisi) est spécifiée par des paramètres, des variables et des règles de décision ne soufrant d'aucune incertitude.
2. Simulation probabiliste : certains événements, éléments dans la dynamique (ou son modèle) ne sont spécifiés que sous la forme de lois de probabilité.

Notion d'échantillon et convention

Un échantillon de taille $n \geq 1$ en statistique correspond au résultat de n répétitions indépendantes de l'expérience aléatoire d'espace fondamental Ω. De manière générale, un échantillon est

1. une suite $\{\omega_n\}_{n\in\mathbb{N}}$ d'éléments de Ω, c'est à dire un élément de $\Omega_\infty = \Omega^{\mathbb{N}}$, tel que les événements $\{\omega_n\}_{n\in\mathbb{N}}$ sont mutuellement indépendants
2. ou une suite de valeurs $\{X_n(\omega)\}_{n\in\mathbb{N}}$ d'une suite $\{X_n\}_{n\in\mathbb{N}}$ v.a. indépendantes et de même loi définies sur Ω_∞ par $X_n(\omega) = \omega_n$. On parle de $\{X_n(\omega)\}_{n\in\mathbb{N}}$ comme une réalisation de la suite $\{X_n\}_{n\in\mathbb{N}}$ de v.a.

On se placera systématiquement dans le second format.

2 Simuler un échantillon de nombres choisis au hasard entre 0 et 1.

1. Le modèle mathématique :

1.a) le tirage au hasard d'un réel compris entre 0 et 1 est représenté par la valeur x d'une v.a. réelle X telle que $X(\Omega) = [0,1]$ et de loi définie par

$$\text{quelque soit l'intervalle } I \text{ de } \mathbb{R} : \quad \mathbb{P}\{X \in I\} = \text{longueur}(I \cap [0,1]);$$

1.b) un échantillon de n réels pris au hasard dans l'intervalle $[0,1]$ sont les n valeurs $x_1, \ldots, x_n$ d'une suite de n v.a. $X_1, \ldots, X_n$ de même loi que X et **indépendantes** les unes des autres :

$$\mathbb{P}\{X_1 \in I_1, \ldots, X \in I_n\} = \prod_{i=1}^{n} \mathbb{P}\{X_i \in I_i\} = \prod_{i=1}^{n} \text{longueur}(I_i \cap [0,1]).$$

2. La simulation : une suite déterministe $\{x_1, \ldots, x_n\}$ obtenue par une relation de récurrence : x_0 le germe ou la « graine de départ » du générateur (seed) et pour $0 \leq i \leq n-1$: $x_{i+1} = f(x_i)$ où f est une fonction mathématique codée par une fonction d'un langage informatique.

Code R1: set.seed et runif

```
set.seed(1) # sélection de x_0
runif(2) # donne 2 tirages x_1, x_2
[1] 0.2655087 0.3721239
runif(2)
[1] 0.5728534 0.9082078
runif(2)
[1] 0.2016819 0.8983897
set.seed(1) # sélection du même x_0 que dans la première ligne
runif(2)
[1] 0.2655087 0.3721239
```

Nous ne nous intéresserons pas ici à l'évaluation approfondie de la qualité de la simulation d'un échantillon d'une loi uniforme sur $[0,1]$. On pourra consulter par exemple [L'E04] https://www.iro.umontreal.ca/~lecuyer/myftp/papers/handstat2.pdf et les pages web http://www.iro.umontreal.ca/~lecuyer/indexf.html et https://www.random.org/. Cependant, plusieurs idées sont naturelles pour « vérifier » que la simulation d'une suite de nombres compris entre 0 et 1 correspond à une séquence de tirages au hasard indépendants de réels dans $[0,1]$. Nous allons aller au plus direct avant d'introduire la notion de quantile.

Comparer les fonctions de répartition empirique et théorique : Pour une suite de v.a. à valeurs réelles $\{X_n\}_{n\geq 1}$, la v.a. appelée la fonction de répartition empirique

$$\forall \omega \in \Omega, \forall x \in \mathbb{R} : \quad F_n(x,\omega) := \frac{1}{n}\sum_{k=1}^{n} 1_{\{X_k \leq x\}}(\omega) = \frac{1}{n}\sum_{k=1}^{n} 1_{]-\infty,x]}\big(X_k(\omega)\big)$$

calcule la proportion de valeurs inférieures ou égales à x d'un n-échantillon. La fonction de répartition théorique de la loi commune de l'échantillon est

$$\forall x \in \mathbb{R} \quad F(x) := \mathbb{P}\{X_1 \leq x\} = \begin{cases} \displaystyle\sum_{x \in X_1(\Omega), x \leq t} \mathbb{P}\{X_1 = x\} & \text{cas discret} \\ \displaystyle\int_{-\infty}^{x} f_{X_1}(u)du & \text{cas à densité.} \end{cases}$$

Code R2: Superposition des deux fonctions de répartition

La fonction ***ecdf(x)*** construit la fonction de répartition empirique des données stockées dans la suite ***x***. On peut alors superposer la courbe théorique de la fonction de répartition. Faire des essais avec des échantillons de taille n croissante.

```
x=runif(n)
plot(ecdf(x))
curve(punif,0,1,add=TRUE,col=2)
```

Commentaire 2.1 *La LFGN établit la convergence (uniforme) de $F_n(\cdot,\omega)$ vers $F(\cdot)$ pour presque tout $\omega \in \Omega$. Il est possible de quantifier la distance entre les deux fonctions de répartition et de réaliser des tests statistiques d'adéquation des données à une loi théorique comme celui de Kolmogorov-Smirnov (voir Projet N° 2).*

Code R3: Simulation de v.a. finies : fonction `sample`

Pour simuler une valeur ou un n-échantillon d'une v.a. suivant une loi finie, il est recommandé d'utiliser la fonction sample qui effectue des tirages sans remise (par défaut) ou avec remise dans un ensemble fini $x := \{x_1, \ldots, x_k\}$ suivant le vecteur des probabilités loi := $(\mathbb{P}\{X = x_1\} = p1, \ldots, \mathbb{P}\{X = x_k\} = pk)$

```
x  = c(x1,...,xk)
loi = c(p1,...,pk)
sample(x,n,replace=TRUE,prob=loi)
```

3 Deux méthodes classiques de générations d'échantillon d'une loi

3.1 Fonction quantile

Une fonction importante en statistique :

Définition 3.1 (Fonction quantile) *Soit X une v.a. de fonction de répartition $F_X(\cdot) := \mathbb{P}\{X \leq \cdot\}$. Pour tout $\alpha \in]0,1[$, le quantile d'ordre α de F_X, noté $Q_X(\alpha)$, est le réel défini par*

$$Q_X(\alpha) := \min\{x \in \mathbb{R} \mid F_X(x) \geq \alpha\}. \tag{1}$$

La fonction $\alpha \mapsto Q_X(\alpha)$ est appelée la fonction quantile associée à la loi de X (on parle également de fonction inverse généralisée). Pour $\alpha := 1/2$, $Q_X(1/2)$ est appelée la valeur médiane de X. Pour $\alpha := 1/4$ et $\alpha := 3/4$ on parle de premier et troisième quartile.

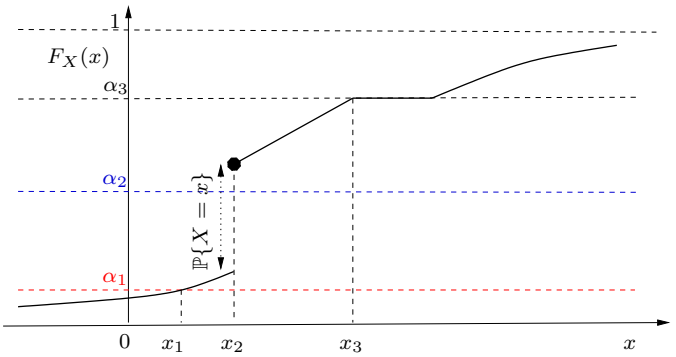

Figure 1 – la plus petite valeur x de X telle que $\mathbb{P}\{X \leq x\} = F_X(x) \geq \alpha$

À partir de la Figure 1, on constate que trois cas de figure se présentent :

α_1 : F_X continue en x_1 et strictement croissante au voisinage de x_1 alors $Q_X(\alpha_1) = x_1$;

α_2 : F_X admet une discontinuité en x_2 : $Q_X(\alpha_2) = x_2$;

α_3 : F_X continue en x_3, croissante au voisinage de x_3 mais avec un « plateau » : $Q_X(\alpha_3) = x_3$.

Commentaire 3.1

1. *$]-\infty, x_\alpha := Q_X(\alpha)]$ est le plus petit intervalle de la forme $]-\infty, x]$ qui « contienne au moins $\alpha \times 100\%$ des valeurs de X », ou encore au moins « $\alpha \times 100\%$ des valeurs de X » sont plus petites (ou égale) à x.*
2. *Dans le cas standard où la v.a. X admet une densité, F_X est continue sur $\mathbb{R}$. Alors*

$$Q_X(\alpha) = \min\{x \in \mathbb{R} \mid F_X(x) = \alpha\}.$$

Si $X(\Omega)$ est un intervalle et F_X établit une bijection de $X(\Omega)$ dans $]0,1[$, alors Q_X n'est rien d'autre que la fonction réciproque ${F_X}^{-1}$ de F_X. On détermine les quantiles d'une loi à partir de la lecture d'une table des valeurs de F_X. L'exemple le plus courant est celui d'une v.a. X de loi normale centrée-réduite pour laquelle on ne dispose d'aucune expression explicite de F_X (voir la Table 1).

α	0.5	0.6	0.7	0.8	0.9	0.95	0.975	0.99
$Q_X(\alpha) = F_X^{-1}(\alpha)$ ou `qnorm`(α)	0	0.2533471	0.5244005	0.8416212	1.2815516	1.6448536	1.9599640	2.3263479

Table 1 – Des quantiles d'une loi normale centrée-réduite

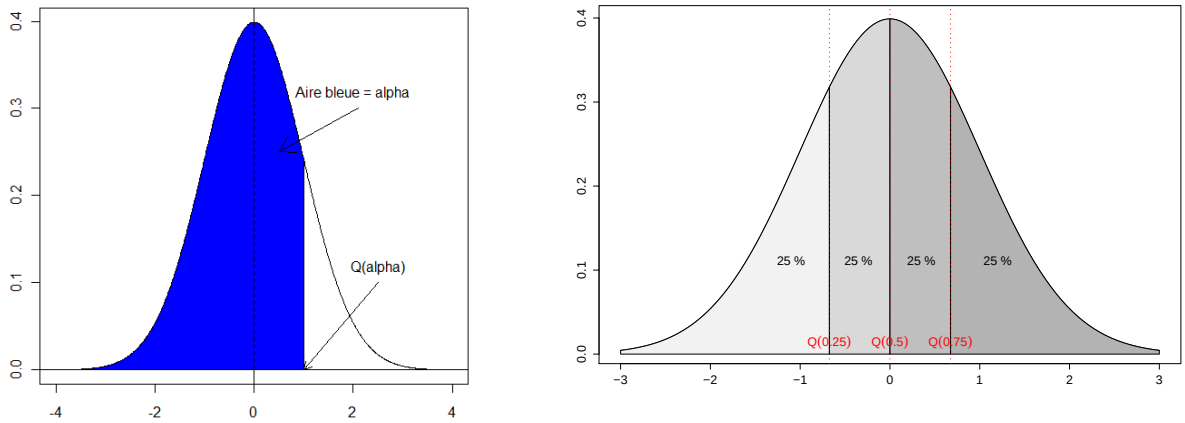

Figure 2 – $X \sim \mathcal{N}(0,1)$: graphe de la densité f_X et quantile d'ordre α / Différents quartiles.

3. *Notons que l'expression de Q_X comme un « min » au lieu d'un « inf » est valable car la fonction de répartition F_X est toujours continue à droite.*

Code R4

Les fonctions de répartition et quantile des lois disponibles sous R s'obtiennent par les commandes `pnomloi` et `qnomloi`, constituées de l'abréviation utilisée par R pour la loi, précédée de la lettre p pour probability function et q pour quantile (voir p.8 pour la liste des lois disponibles extraite de [LDL11]). Attention `dnomloi` donne la density function de la loi et non sa « distribution function » appellation classique en anglais de la fonction de répartition. Par ailleurs, `rnomloi` donne le générateur.

3.2 Méthode de l'inverse

Lemma 3.2 *Soit U une v.a. de loi uniforme sur $[0,1]$. Alors pour toute v.a. X, la v.a. $Q_X(U)$ admet la même loi que X.*

La preuve de ce résultat repose sur le calcul de la fonction de répartition de la v.a. $Q_X(U)$ et l'équivalence suivante (cf Ex 3. D.L. 2)

$$Q_X(U) \leq x \Longleftrightarrow U \leq F_X(x).$$

Le cas des v.a. à densité

1. Si $X \sim \text{Unif}([a,b])$ avec $a < b$ alors $F_X(x) = 1_{]a,b[}(x)(x-a)/(b-a)$ établit une bijection de l'intervalle $X(\Omega) =]a,b[$ vers $]0,1[$. Alors, pour $\alpha \in]0,1[$ donné, en résolvant l'équation $F_X(x_\alpha) = \alpha$, on trouve $x_\alpha \equiv Q_X(\alpha) = {F_X}^{-1}(\alpha) = (b-a)\alpha + a$.

Ainsi, à partir d'un n-échantillon $(U_1, \ldots, U_n)$ d'une loi uniforme sur $[0,1]$, la famille de v.a.

$$(Q_X(U_1), \ldots, Q_X(U_n)) = \big((b-a)U_1 + a, \ldots, (b-a)U_n + a\big)$$

est un n-échantillon d'une loi uniforme sur $[a,b]$.

2. $X \sim \text{Exp}(\lambda)$ avec $\lambda > 0$. Alors $F_X(x) = (1 - \exp(-\lambda x))1_{]0,+\infty[}(x)$ établit une bijection de $X(\Omega) =]0,+\infty[$ sur $]0,1[$. Alors, pour $\alpha \in]0,1[$ donné, en résolvant l'équation $F_X(x_\alpha) = \alpha$, on trouve $x_\alpha \equiv Q_X(\alpha) = {F_X}^{-1}(\alpha) = -\ln(1-\alpha)/\lambda$.

Ainsi, à partir d'un n-échantillon $(U_1, \ldots, U_n)$ d'une loi uniforme sur $[0,1]$, la famille de v.a.

$$(Q_X(U_1), \ldots, Q_X(U_n)) = \big(-\ln(1-U_1)/\lambda, \ldots, -\ln(1-U_n)/\lambda\big)$$

est un n-échantillon d'une loi exponentielle de paramètre λ.

Exercice 1 *Vérifier que* $\big(-\ln(U_1)/\lambda, \dots, -\ln(U_n)/\lambda\big)$ *est également un* n*-échantillon d'une loi exponentielle de paramètre* λ*.*

Code R5: Loi de Laplace

Une v.a. Y suit une loi de Laplace de paramètre $a > 0$ si elle admet pour densité

$$\forall x \in \mathbb{R}, \quad g_a(x) = \frac{a}{2}\exp(-a|x|)$$

associée à une loi dite doublement exponentielle ou de Laplace.

1. Montrer que l'on peut générer une valeur d'une v.a. de loi de Laplace par la méthode de l'inverse (Ex 2, D.L. n° 2).
2. Construire une fonction *rlaplace(n,a)*, où n est un entier, qui simule la réalisation d'un n-échantillon $(Y_1, \dots, Y_n)$ de la loi de Laplace par la méthode de l'inverse.

3.3 Méthode de l'acceptation-rejet (AR) ou « Hit and Miss »

On souhaite générer une réalisation d'une v.a. X admettant pour densité f_X. Supposons qu'il existe une densité de probabilité g et une constante c telles que :

– on dispose de l'inégalité (pourquoi a-t-on alors $c \geq 1$?)

$$\forall x \in X(\Omega), \quad f_X(x) \leq c\, g(x); \tag{2}$$

– ET la simulation d'une réalisation y d'une v.a. Y de densité g est « facile » (car un générateur R est disponible, la méthode de l'inverse peut être mise en place...)

La densité f_X est appelée « la densité cible » et g la « densité instrumentale ».

L'algorithme AR suivant donne une simulation d'une réalisation x de X :

1. Générer une réalisation y d'une v.a. admettant pour densité g.
2. Générer un nombre u pris au hasard entre 0 et 1, puis calculer $ucg(y) \in [0, cg(y)]$
3. Si $u\, c\, g(y) \leq f_X(y)$ alors accepter la valeur de y comme une réalisation de X (poser $x := y$)
   Si non, rejet de la valeur y et on repart en 1.

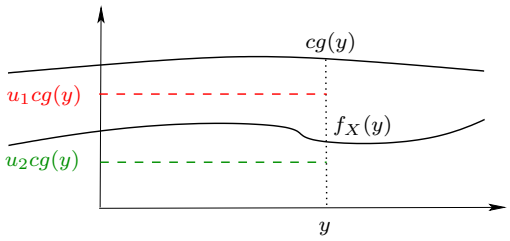

Figure 3 – Pour $X(\Omega) \subset \mathbb{R}$: Hit pour $u_2 cg(y) \leq f_X(y)$ et Miss pour $u_1 cg(y) > f_X(y)$.

Lemma 3.3 *Si* Y *est une v.a. de densité* g *et si* $U \sim \mathrm{Unif}([0,1])$ *indépendante de* Y *alors :*

1. *la probabilité d'accepter à l'étape 3 la valeur produite à l'étape 1 de l'algorithme est*

$$\mathbb{P}\big\{Ucg(Y) \leq f_X(Y)\big\} = \frac{1}{c}. \tag{3}$$

La loi du nombre de répétitions de l'algorithme AR pour accepter une valeur candidate est géométrique de paramètre $1/c$*. Ainsi l'algorithme est d'autant plus « efficace » que* c *est proche de* 1*.*

2. *La loi de* Y *sachant que l'événement* $\big\{Ucg(Y) \leq f_X(Y)\big\}$ *est réalisé admet pour densité* f_X*.*

La preuve du second point s'appuie sur le calcul de la fonction de répartition conditionnelle de Y sachant $A := \{Ucg(Y) \leq f_X(Y)\}$ à l'aide de la formule de Bayes et l'indépendance des deux v.a. Y et U :

$$F_{Y|A}(y) = \mathbb{P}\{Y \leq y \mid A\} = \frac{\mathbb{P}\{Y \leq y, A\}}{\mathbb{P}(A)} = \cdots = \int_{-\infty}^{y} f_X(x)dx = F_X(y).$$

Commentaire 3.2

1. *La méthode AR est encore valide même si f_X dans (2) n'est plus une densité. En effet, si (2) est remplacée par $0 \le f(x) \le cg(x)$ avec $\int_{\mathbb{R}} f(x)\,dx \neq 1$ alors on vérifie facilement que*

   1.a) la probabilité d'accepter un candidat quand $ucg(y) \le f(y)$ est $\mathbb{P}\{Ucg(Y) \le f(Y)\} = \int_{\mathbb{R}} f(x)\,dx/c$.

   1.b) la loi de Y sachant que l'événement $\{Ucg(Y) \le f(Y)\}$ est réalisé admet pour densité $f_X(\cdot) = f(\cdot)/\int_{\mathbb{R}} f(x)\,dx$.

   Ainsi l'algorithme peut être utilisé même si la densité f_X est connue seulement à une constante multiplicative près. Cette remarque est fondamentale dans le cadre de la statistique bayésienne qui met en jeu le plus souvent la génération d'échantillons associés à une densité dont l'expression est connue à une constante multiplicative près.
2. *Si l'inégalité (2) est valide pour une certaine constante c, il n'est pas nécessaire que c soit la constante optimale pour que l'algorithme fonctionne. Autrement dit, l'algorithme génère des réalisations de X de densité f_X même s'il existe un $c' < c$ telle que $f_X \le c'\,g$ sur $X(\Omega)$. Cependant, il y a une contre-partie en termes de probabilité d'acceptation d'un candidat y (voir le lemme qui suit) donc de « coût » de simulation.*
3. *L'algorithme est valable pour $X(\Omega)$ discret (avec $f_X(x) \equiv \mathbb{P}\{X = x\}$) ou $X(\Omega) \subset \mathbb{R}^d$.*

Code R6

On souhaite simuler une réalisation d'une v.a. $X \sim \mathcal{N}(0,1)$. On « ne peut pas utiliser » la méthode de l'inverse. On considère la densité instrumentale suivante : pour $a > 0$

$$\forall x \in \mathbb{R}, \quad g_a(x) = \frac{a}{2}\exp(-a|x|)$$

associée à une loi dite doublement exponentielle ou de Laplace.

1. À l'aide de la fonction curve, superposer les densités dnorm et g_a pour quelques valeurs de a.
2. Montrer que
$$\forall a > 0,\ \forall x \in \mathbb{R}, \quad \frac{f_X(x)}{g_a(x)} \le c_a := \sqrt{\frac{2}{\pi}}\frac{1}{a}\exp(a^2/2).$$
Montrer que c_a est minimale sur $]0, +\infty[$ pour $a := 1$.
3. En déduire que la probabilité d'acceptation de l'algorithme AR est alors de $\sqrt{\pi/2e}$ et qu'il faut en moyenne la simulation de $\sqrt{2e/\pi}$ tirages au hasard dans $[0,1]$ pour obtenir une valeur de X.
4. Donner un algorithme (et un code R) qui simule la réalisation d'un n-échantillon $(X_1, \ldots, X_n)$ d'une loi $\mathcal{N}(0,1)$ à l'aide de votre générateur `rlaplace` du code R5.
5. On réalisera des comparaisons, à l'aide de la fonction `system.time`, de temps de génération de l'algorithme obtenu avec la fonction `rnorm`. Par ailleurs, si rgaussAR permet la génération d'un n échantillon d'une loi $\mathcal{N}(0,1)$ suivant la méthode AR avec la loi de Laplace comme modèle instrumental, on pourra apprécier la qualité des échantillons générés avec un test de Kolmogorov-Smirnov ks.test

```
x=rgaussAR(100)
ks.test(x,"pnorm")
```

Des explications sur l'utilisation du test sont données dans un document sous moodle.

Le modèle mathématique d'un tirage au hasard d'un point dans le carré $[0,1]^2$ est donné par la probabilité sur le plan définie par

$$\forall S \subset \mathbb{R}^2, \quad \mathbb{P}\{(X,Y) \in S\} = \text{Aire}(S \cap [0,1]^2).$$

On peut vérifier que les deux v.a. X et Y sont indépendantes et de même loi uniforme sur $[0,1]$. Pour générer une réalisation de (X,Y), il suffit donc de faire un appel `runif(2)`.

Code R7: Simulation d'un tirage au hasard d'un point dans une surface $S \subset [0,1]^2$

1. On considère un couple $(X,Y) \sim \text{Unif}([0,1]^2)$.

   1.a) Montrer que l'inégalité de base (2) de la méthode AR est satisfaite avec les éléments :

$$f(x,y) := \frac{1}{\text{Aire}(S)}\,1_S(x,y), \quad g(x,y) := 1_{[0,1]^2}(x,y), \quad c := \frac{1}{\text{Aire}(S)}.$$

1.b) Montrer que la méthode AR conduit alors à accepter un couple (x, y), tiré au hasard dans le carré $[0, 1]^2$, comme une réalisation d'un tirage au hasard d'un point dans S ssi $(x, y) \in S$.

2. En déduire un algorithme qui génère un tirage au hasard dans une surface $S := \{(x, y) \in \mathbb{R}^2 : s(x, y) \leq 0\}$ (avec $s(\cdot, \cdot)$ donnée) incluse dans le carré $[0, 1]^2$.

3. Mettre en œuvre le cas où S est un disque inscrit dans le carré unité. Visualiser l'échantillon produit dans le plan et superposer le cercle sur votre graphique. Discuter qualitativement les résultats obtenus.

Le code suivant prend dix points pris au hasard dans le domaine $[-1, 1]^2$, superpose le cercle de rayon 1 centré en $(0, 0)$ et trace le contour du domaine $[-1, 1]^2$

```
x=runif(10,-1,1)
y=runif(10,-1,1)
par(pty="s") # oblige a respecter une échelle identique en x et y
plot(x,y,xlim=c(-1,1), ylim=c(-1,1))
symbols(0,0,circles=1, inches=F, add=T)
abline(v=c(-1,1), h=c(-1,1))
```

Références

[LDL11] P. Lafaye de Micheaux, Rémy Drouilhet, and B. Liquet. *Le logiciel R*. Springer, 2011.

[L'E04] P. L'Ecuyer. *Handbook of Computational Statistics*, chapter Random Number Generation, pages 35–70. Springer-Verlag, 2004.

[RC04] C. P. Robert and P. Casella. *Monte Carlo Statistical methods*. Springer, 2004.

A Et en MatLab ?

Les fonctions `rand`, `randn` permettent d'obtenir des réalisations de v.a. suivant une loi uniforme sur l'intervalle $[0, 1]$ et une loi normale centrée réduite respectivement.

Comme souvent avec MatLab, des outils supplémentaires sont proposés dans des toolbox spécialisées. La toolbox `Stats` offre une bibliothèque de fonctions liées aux probabilités et statistiques (https://fr.mathworks.com/help/stats/). Il existe une alternative « libre », la toolbox `Stixbox`. Une version récente est intégrée comme un atoms de SciLab (voir http://forge.scilab.org/index.php/p/stixbox/ et https://atoms.scilab.org/toolboxes/stixbox).

Dans ce cas très simple, la théorie nous dit que le biais est nul et que la variance vaut $p(1-p)/n = 0.00694$.

Astuce

Notez l'existence du *package* `boot` qui facilite la pratique du *bootstrap* :
`boot(xvec,function(x,w) sum(x[w]==4)/n,B)`

SECTION 10.7

Lois usuelles et moins usuelles

10.7.1 Lois usuelles

Les lois de probabilité courantes sont implémentées dans R. Nous donnons, dans les tableaux 10.1 et 10.2, les fonctions permettant de calculer la densité (ou la fonction de masse), la fonction de répartition et la fonction quantile de ces lois. Nous donnons également l'instruction permettant de générer des nombres pseudo-aléatoires issus de ces lois.

TAB. 10.1: Lois discrètes usuelles. Fonctions R pour la fonction de masse (`d-`), de répartition (`p-`) et quantile (`q-`). Instruction pour générer (`r-`) des nombres pseudo-aléatoires issus de ces lois.

Lois discrètes	**Fonctions R**	**Espérance** **Variance**	**Fonction** **de masse** $P(\mathrm{X}=x)$

Binomiale(m,α)	`dbinom(x,size=`m`,prob=`α`)`
	`pbinom(q,size=`m`,prob=`α`)`
	`qbinom(p,size=`m`,prob=`α`)`
	`rbinom(n,size=`m`,prob=`α`)`
		$m\alpha$
		$m\alpha(1-\alpha)$
			$\binom{m}{x}\alpha^x(1-\alpha)^{m-x}$

Poisson(λ)	`dpois(x,lambda=`λ`)`
	`ppois(q,lambda=`λ`)`
	`qpois(p,lambda=`λ`)`
	`rpois(n,lambda=`λ`)`
		λ
		λ
			$e^{-\lambda}\frac{\lambda^x}{x!}$

Géométrique(α)	`dgeom(x,prob=`α`)`
	`pgeom(q,prob=`α`)`
	`qgeom(p,prob=`α`)`
	`rgeom(n,prob=`α`)`
		$\frac{1}{\alpha}$
		$\frac{1-\alpha}{\alpha^2}$
			$(1-\alpha)^{x-1}\alpha$

Hyper-géométrique(m,n,k)	`dhyper(x,m=`m`,n=`n`,k=`k`)`
	`phyper(q,m=`m`,n=`n`,k=`k`)`
	`qhyper(p,m=`m`,n=`n`,k=`k`)`
	`rhyper(nn,m=`m`,n=`n`,k=`k`)`
		$\frac{nm}{N}$ (avec $N=n+m$)
		$\frac{n(m/N)(1-(m/N))(N-n)}{(N-1)}$
			$\frac{\binom{m}{x}\binom{n}{k-x}}{\binom{m+n}{k}}$

Binomiale négative(m,α)	`dnbinom(x,size=`m`,prob=`α`)`
	`pnbinom(q,size=`m`,prob=`α`)`
	`qnbinom(p,size=`m`,prob=`α`)`
	`rnbinom(n,size=`m`,prob=`α`)`
		$m\frac{1-\alpha}{\alpha}$
		$m\frac{1-\alpha}{\alpha^2}$
			$\binom{x+m-1}{m-1}\alpha^m(1-\alpha)^x$

Uniforme discrète$\{1,\ldots,m\}$	`(x %in% 1:`m`)/`m
	`sum(1:`m`<=q)/`m
	`match(1,1:`m`/`m`>=p)`
	`sample(x=1:`m`,size=n,TRUE)`
		$\frac{m+1}{2}$
		$\frac{m^2-1}{12}$
			$\frac{1}{m}\mathbb{1}_{\{1,\ldots,m\}}(x)$

TAB. 10.2: Lois continues usuelles. Fonctions **R** pour la fonction de densité (`d-`), de répartition (`p-`) et quantile (`q-`). Instruction pour générer (`r-`) des nombres pseudo-aléatoires issus de ces lois (notations : $B(\cdot,\cdot)$: fonction bêta, $I(\cdot)$: fonction de Bessel modifiée, $\Gamma(\cdot)$: fonction gamma, $P(\cdot;\lambda)$: fonction de masse d'une Poisson(λ), $I'_x(\cdot,\cdot)$: dérivée de la fonction bêta incomplète, $\text{sech}(x) = \frac{2}{e^x+e^{-x}}$).

Lois continues	**Fonctions R**	**Espérance Variance**	**Densité**

Normale(μ,σ^2)	`dnorm(x,mean=`μ`,sd=`σ`)`
	`pnorm(q,mean=`μ`,sd=`σ`)`
	`qnorm(p,mean=`μ`,sd=`σ`)`
	`rnorm(n,mean=`μ`,sd=`σ`)`
		μ
		σ^2
			$\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$

Student(ν,μ)	`dt(x,df=`ν`,ncp=`μ`)`
	`pt(q,df=`ν`,ncp=`μ`)`
	`qt(p,df=`ν`,ncp=`μ`)`
	`rt(n,df=`ν`,ncp=`μ`)`
		$\mu\sqrt{\frac{\nu}{2}}\frac{\Gamma((\nu-1)/2)}{\Gamma(\nu/2)}$
		$(\nu>1)$
		$\frac{\nu(1+\mu^2)}{\nu-2} - \frac{\mu^2\nu}{2}\times$
		$\left(\frac{\Gamma((\nu-1)/2)}{\Gamma(\nu/2)}\right)^2$, $(\nu>2)$
			$\frac{\nu^{\nu/2}e^{-\nu\mu^2/2(x^2+\nu)}}{\sqrt{\pi}\Gamma(\nu/2)2^{(\nu-1)/2}(x^2+\nu)^{(\nu+1)/2}}$
			$\times\int_0^\infty t^\nu e^{\frac{\mu x}{2\sqrt{x^2+\nu}}-\frac{t}{2}}dt$

Khi-deux(k,λ)	`dchisq(x,df=`k`,ncp=`λ`)`
	`pchisq(q,df=`k`,ncp=`λ`)`
	`qchisq(p,df=`k`,ncp=`λ`)`
	`rchisq(n,df=`k`,ncp=`λ`)`
		$k+\lambda$
		$2(k+2\lambda)$
			$\frac{1}{2}e^{-(x+\lambda)/2}\left(\frac{x}{\lambda}\right)^{k/4-1/2}$
			$\times I_{k/2-1}(\sqrt{\lambda x})$

Fisher(ν_1,ν_2,λ)	`df(x,df1=`ν_1`,df2=`ν_2`,ncp=`λ`)`
	`pf(q,df1=`ν_1`,df2=`ν_2`,ncp=`λ`)`
	`qf(p,df1=`ν_1`,df2=`ν_2`,ncp=`λ`)`
	`rf(n,df1=`ν_1`,df2=`ν_2`,ncp=`λ`)`
		$\frac{\nu_2(\nu_1+\lambda)}{\nu_1(\nu_2-2)}$
		$(\nu_2>2)$
		$2\frac{(\nu_1+\lambda)^2+(\nu_1+2\lambda)(\nu_2-2)}{(\nu_2-2)^2(\nu_2-4)}$
		$\times\left(\frac{\nu_2}{\nu_1}\right)^2$, $(\nu_2>4)$
			$\sum_{k=0}^{\infty}\frac{e^{-\lambda/2}(\lambda/2)^k}{B\left(\frac{\nu_2}{2},\frac{\nu_1}{2}+k\right)k!}\left(\frac{\nu_1}{\nu_2}\right)^{\frac{\nu_1}{\nu_2}+k}$
			$\times\left(\frac{\nu_2}{\nu_2+\nu_1 x}\right)^{\frac{\nu_1+\nu_2}{2}+k} x^{\frac{\nu_1}{2}-1+k}$

Exponentielle(λ)	`dexp(x,rate=`λ`)`
	`pexp(q,rate=`λ`)`
	`qexp(p,rate=`λ`)`
	`rexp(n,rate=`λ`)`
		$\frac{1}{\lambda}$
		$\frac{1}{\lambda^2}$
			$\lambda e^{-\lambda x}\mathbb{1}\{x\geq 0\}$

Uniforme(a,b)	`dunif(x,min=`a`,max=`b`)`
	`punif(q,min=`a`,max=`b`)`
	`qunif(p,min=`a`,max=`b`)`
	`runif(n,min=`a`,max=`b`)`
		$\frac{a+b}{2}$
		$\frac{(b-a)^2}{12}$
			$\frac{1}{b-a}\mathbb{1}\{a\leq x\leq b\}$

Bêta(α,β,λ)	`dbeta(x,shape1=`α`,shape2=`β`,ncp=`λ`)`
	`pbeta(q,shape1=`α`,shape2=`β`,ncp=`λ`)`
	`qbeta(p,shape1=`α`,shape2=`β`,ncp=`λ`)`
	`rbeta(n,shape1=`α`,shape2=`β`,ncp=`λ`)`
		$\approx 1-\frac{\beta}{C}\left(1+\frac{\lambda}{2C^2}\right)$
		avec $C=\alpha+\beta+\frac{\lambda}{2}$
		$\frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$ si $\lambda=0$
			$\sum_{i=0}^{\infty}P(i;\frac{\lambda}{2})I'_x(\alpha+i,\beta)$

Cauchy(x_0,γ)	`dcauchy(x,location=`x_0`,scale=`γ`)`
	`pcauchy(q,location=`x_0`,scale=`γ`)`
	`qcauchy(p,location=`x_0`,scale=`γ`)`
	`rcauchy(n,location=`x_0`,scale=`γ`)`
		Non définie
		Non définie
			$\frac{1}{\pi}\left[\frac{\gamma}{(x-x_0)^2+\gamma^2}\right]$

Logistique(μ,s)	`dlogis(x,location=`μ`,scale=`s`)`
	`plogis(q,location=`μ`,scale=`s`)`
	`qlogis(p,location=`μ`,scale=`s`)`
	`rlogis(n,location=`μ`,scale=`s`)`
		μ
		$\frac{\pi^2}{3}s^2$
			$\frac{1}{4s}\text{sech}^2\left(\frac{x-\mu}{2s}\right)$

Log-Normale(μ,σ)	`dlnorm(x,meanlog=`μ`,sdlog=`σ`)`
	`plnorm(q,meanlog=`μ`,sdlog=`σ`)`
	`qlnorm(p,meanlog=`μ`,sdlog=`σ`)`
	`rlnorm(n,meanlog=`μ`,sdlog=`σ`)`
		$e^{\mu+\sigma^2/2}$
		$(e^{\sigma^2}-1)e^{2\mu+\sigma^2}$
			$\frac{1}{x\sigma\sqrt{2\pi}}e^{-\frac{(\ln(x)-\mu)^2}{2\sigma^2}}$

Gamma(α,β)	`dgamma(x,shape=`α`,rate=`β`)`
	`pgamma(q,shape=`α`,rate=`β`)`
	`qgamma(p,shape=`α`,rate=`β`)`
	`rgamma(n,shape=`α`,rate=`β`)`
		$\alpha\beta$
		$\alpha\beta^2$
			$x^{\alpha-1}\frac{e^{-x/\beta}}{\beta^\alpha\Gamma(\alpha)}\mathbb{1}_{x>0}$

Weibull(λ,k)	`dweibull(x,shape=`λ`,scale=`k`)`
	`pweibull(q,shape=`λ`,scale=`k`)`
	`qweibull(p,shape=`λ`,scale=`k`)`
	`rweibull(n,shape=`λ`,scale=`k`)`
		$\lambda\Gamma\left(1+\frac{1}{k}\right)$
		$\lambda^2\Gamma\left(1+\frac{2}{k}-\mu^2\right)$
			$\frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1}e^{-(x/\lambda)^k}\mathbb{1}_{x\geq 0}$

Gumbel(μ,β)
(*Package* `evd`)
	`dgumbel(x,loc=`μ`,scale=`β`)`
	`pgumbel(q,loc=`μ`,scale=`β`)`
	`qgumbel(p,loc=`μ`,scale=`β`)`
	`rgumbel(n,loc=`μ`,scale=`β`)`
		$\mu+\beta$
		$\frac{\pi^2}{6}\beta^2$
			$\frac{ze^{-z}}{\beta}$ avec $z=e^{-\frac{x-\mu}{\beta}}$