

COMPLÉMENT III

Simulation de variables aléatoires

Références : *Modélisation stochastique et simulation*, B. Bercu et D. Chafai

Pour Scilab, non testé : *Introduction à scilab*, P. Chancelier, ed. Springer.

Algorithmique et mathématiques : travaux pratiques et applications scilab pour le lycée et la licence, J. Ouin, ed. Ellipses

On trouve aussi facilement des tutoriels et manuels d'introduction à scilab sur internet.

Le problème qu'on se pose dans ce chapitre est l'écriture d'algorithmes qui simulent une expérience aléatoire : une sorte de "dé informatique". La brique de base pour cela est le générateur aléatoire de loi uniforme sur $[0, 1]$ (sous scilab, commande **rand**). En général, ce générateur n'est que "pseudo-aléatoire" au sens où il est construit sur un algorithme déterministe. Mais on suppose qu'il est parfait, et que différents appels à cet algorithme fournissent des variables aléatoires de loi uniforme sur $[0, 1]$, indépendantes.

A partir de cette brique de base, on peut obtenir toutes les lois, par des considérations assez simples.

1. Utilisation de propriétés de certaines lois

- (1) La loi de Bernoulli de paramètre p .

C'est aussi une brique de base. L'algorithme le plus simple consiste à couper l'intervalle $[0, 1]$ en deux sous-intervalles, l'un de taille p , l'autre de taille $1 - p$:

```
U = rand(1);
if(U < p) then X = 1; else X = 0;
end
```

Ou encore sous scilab

```
U = rand(1);
X = 1 * (U < p)
```

En effet on vérifie bien que X est à valeurs dans $\{0; 1\}$ et que $\mathbb{P}(X = 1) = \mathbb{P}(U < p) = p$

- (2) Loi discrète (Par exemple) Loi sur $\{1, 34, 48\}$ de loi $\mathbb{P}(1) = \frac{2}{3}$, $\mathbb{P}(34) = \frac{1}{6}$, $\mathbb{P}(48) = \frac{1}{6}$
On partitionne l'intervalle $[0, 1]$ en trois sous-intervalles de taille respective $\frac{2}{3}$, $\frac{1}{6}$ et $\frac{1}{6}$. Une manière de faire cela consiste à prendre $[0, \frac{2}{3}]$, $[\frac{2}{3}, \frac{2}{3} + \frac{1}{6}]$, $[\frac{2}{3} + \frac{1}{6}, 1]$. L'algorithme est alors :

```
U = rand(1);
if (U < 2/3) then X = 1;
elseif (U < 5/6) then X = 34;
else X = 48
end
```

Ou encore sous scilab :

```
U = rand(1);
X = (U < 2/3) + 34 * ((U > 2/3) & (U < 5/6)) + 48 * (U > 5/6)
```

- (3) Loi discrète uniforme sur $\{1, 2, \dots, k\}$.

On va maintenant partitionner l'intervalle $[0, 1]$ dans lequel U prend ses valeurs en k sous-intervalles de taille $\frac{1}{k}$. Il faut ensuite déterminer dans lequel de ces sous-intervalles se trouve U . Une manière de faire cela est d'utiliser la partie entière supérieure :

```
U = rand(1);
X = ceil(k * U)
```

- (4) Loi binomiale de paramètre
- (n, p)

Il suffit d'ajouter des lois de Bernoulli : sous scilab, par exemple

$U = \text{rand}(1, n)$

$Y = (U < p)$;

$X = \text{sum}(Y)$

- (5) Loi géométrique de paramètre
- p

Il suffit de répéter des Bernoulli jusqu'à obtenir un succès :

$X = 1$;

$U = \text{rand}(1)$;

while($U > p$) $X = X + 1$; $U = \text{rand}(1)$; *end*

X

- (6) Loi uniforme sur un intervalle
- I

On utilise le fait qu'une transformation affine d'une loi uniforme sur un intervalle est une loi uniforme sur l'image de l'intervalle. Ainsi il suffit de trouver a et b tel que l'image par $x \mapsto ax + b$ de $[0, 1]$ soit I puis :

$U = \text{rand}(1)$;

$X = a * U + b$

2. Inversion de la fonction de répartition

Rappel du théorème fondamental :

Proposition 1. Soit F une fonction de répartition. On appelle fonction quantile la fonction

$$F^{-}(u) = \inf \{x : F(x) \geq u\}, \quad u \in]0, 1[$$

Alors si U est une variable de loi uniforme sur $[0, 1]$, la variable $X = F^{-}(U)$ a pour fonction de répartition F .

Ainsi, à partir d'un générateur de loi uniforme sur $[0, 1]$, on peut en principe simuler n'importe quelle loi, **à condition d'avoir un algorithme pour calculer F^{-}** . Rappelons que lorsque X est à densité, F^{-} est la fonction inverse (au sens fonction réciproque) de la fonction de répartition.

Exemple : simulation d'une loi exponentielle de paramètre λ .

La fonction de répartition est $x \mapsto 1 - \exp(-\lambda x)$, sa réciproque est $x \mapsto -\frac{1}{\lambda} \ln(1 - x)$. D'où l'algorithme (car U et $1 - U$ ont la même loi) :

$U = \text{rand}(1)$;

$X = -\log(U)/\lambda$;

3. Méthode du rejet

Il y en a plusieurs versions. La plus simple consiste à donner une méthode pour simuler des points suivant une distribution uniforme sur un borélien de \mathbb{R}^2 de mesure finie.

Théorème 2.**Méthode du rejet 1**

Soient $B \subset A$ deux boréliens de \mathbb{R}^2 de mesure de Lebesgue finies $0 < \lambda(B) \leq \lambda(A)$. Soit (X_n) une suite de vecteurs aléatoires indépendants de loi uniforme sur A . On définit pour tout entier n $Y_n = 1_B(X_n)$. Alors

- La suite (Y_n) est une suite de variables aléatoires indépendantes suivant une loi de Bernoulli de paramètre $p = \frac{\lambda(B)}{\lambda(A)}$
- $T = \min\{n > 0, X_n \in B\}$ suit une loi géométrique de paramètre p et est en particulier p.s. finie
- les variables T et X_T sont indépendantes
- X_T suit une loi uniforme sur B .

Ainsi à partir d'une loi uniforme sur un borélien A il est assez facile de simuler une loi uniforme sur un borélien $B \subset A$. Cela prendra par cette méthode d'autant plus de temps que A est plus gros que B . Pour A on prend souvent un pavé $[a, b] \times [c, d]$: en effet simuler une loi uniforme sur ce pavé se fait en simulant les deux coordonnées de manière indépendante, l'abscisse uniforme sur $[a, b]$ et l'ordonnée uniforme sur $[c, d]$.

Exemple : l'algorithme scilab :

```
X = rand(1)
Y = rand(1)
while((X^2 + Y^2) > 1)
X = rand(1); Y = rand(1);
end
(X, Y)
```

fournit un point uniforme sur un quart de disque unité, au bout d'un nombre géométrique d'itérations, de paramètre $\pi/4$.

On peut aussi sous scilab obtenir un échantillon d'un coup : l'algorithme

```
U = rand(1000, 1)
V = rand(1000, 1)
J = find(U^2 + V^2 < 1)
X = U(J); Y = V(J)
```

simule un échantillon de points uniformes indépendants sur un quart de disque unité. Le nombre de points de l'échantillon est aléatoire : c'est ici une binomiale de paramètres $(1000, \pi/4)$.

DÉMONSTRATION. – Les Y_n sont indépendantes et de Bernoulli puisqu'à valeurs dans $\{0, 1\}$; leur paramètre est $P(X_n \in B) = \frac{\lambda(B)}{\lambda(A)}$

– Se déduit du point précédent.

– Soit $C \subset B$ un borélien de \mathbb{R}^2 et $n \geq 1$.

$$\mathbb{P}(T = n \text{ et } X_T \in C) = \mathbb{P}(X_1 \notin B, X_2 \notin B, \dots, X_n \in C) = \mathbb{P}(X_1 \notin B)^{n-1} \mathbb{P}(X_n \in C) = (1-p)^{n-1} \frac{\lambda(C)}{\lambda(A)}$$

On obtient bien un produit d'une fonction de n par une fonction de C : X_T et T sont donc bien indépendantes

– Comme on connaît la loi de T on en déduit immédiatement $\mathbb{P}(X_T \in C) = \frac{\lambda(C)}{\lambda(B)}$: X_T suit donc bien une loi uniforme sur B .

□

Cette méthode s'utilise facilement comme on l'a vu pour simuler par exemple une loi uniforme sur un disque ; mais on peut aussi l'utiliser pour simuler une variable réelle de densité donnée, grâce à la proposition suivante :

Théorème 3.

Méthode du rejet 2 Soit f une densité de probabilité sur \mathbb{R} . Si le vecteur $(X, Y) \in \mathbb{R}^2$ suit une loi uniforme sur $\{(x, y) \in \mathbb{R}^2, 0 < y < f(x)\}$, alors X a pour densité f .

Ainsi pour simuler une v.a. ayant une densité donnée, il peut suffire en vertu des propositions précédentes de trouver un pavé qui contienne $\{(x, y) \in \mathbb{R}^2, 0 < y < f(x)\}$ et d'appliquer la méthode du rejet (ce qui nécessite d'avoir un algorithme pour calculer f). On peut noter que le premier point est impossible si f a un support non borné...

DÉMONSTRATION. On commence par calculer $\lambda(\{(x, y) \in \mathbb{R}^2, 0 < y < f(x)\}) = \int_{\mathbb{R}} f(x)dx = 1$ (application directe de Fubini). Ensuite si $x \in \mathbb{R}$, on obtient encore par Fubini $\mathbb{P}(X \leq x) = \int_{-\infty}^x (\int_0^{f(t)} dy)dt = \int_{-\infty}^x f(t)dt$. \square

Comment faire lorsque la loi qu'on souhaite simuler n'a pas un support borné? On va généraliser les principes ci-dessus. Le lemme ci-dessous est la réciproque du théorème précédent : il dit que si on sait simuler une variable aléatoire de densité g (par exemple par inversion de la fonction de répartition), alors on peut pour tout $c > 0$ facilement simuler un vecteur aléatoire de loi uniforme sur $\{(x, y) \in \mathbb{R}^2, 0 < y < cg(x)\}$.

Lemme 4. Soit g une densité de probabilité sur \mathbb{R} . Soient X et U deux variables aléatoires indépendantes, telles que X a pour densité g et U suit une loi uniforme sur $[0, 1]$. Alors pour tout $c > 0$, le vecteur $(X, cg(X)U)$ suit une loi uniforme sur $\{(x, y) \in \mathbb{R}^2, 0 < y < cg(x)\}$.

DÉMONSTRATION. Soit $c > 0$. Il est clair que $(X, cg(X)U)$ est à valeurs dans $A = \{(x, y) \in \mathbb{R}^2, 0 < y < cg(x)\}$, et on vérifie facilement que $\lambda(A) = c$. Soit ϕ une fonction borélienne bornée de \mathbb{R}^2 .
$$\mathbb{E}[\phi(X, cg(X)U)] = \int_{\mathbb{R} \times [0, 1]} \phi(x, cg(x)u) dx du = \int_{\mathbb{R}} (\int_0^{cg(x)} \frac{\phi(x, y)}{c} dx dy) = \frac{\int_A \phi(x, y) dx dy}{\lambda(A)}$$
. \square

Enfin on obtient une méthode assez générale de simulation par rejet :

Théorème 5.

Méthode du rejet 3 Soit f une densité de probabilités sur \mathbb{R} . On suppose qu'il existe $c > 0$ et une densité de probabilités g telle que pour tout réel x $f(x) \leq cg(x)$. Soient (U_n) une suite de variables aléatoires indépendantes de loi uniforme sur $[0, 1]$ et (X_n) une suite de variables aléatoires indépendantes de densité g , indépendantes de (U_n) . Alors $T = \min\{n > 0, cg(X_n)U_n < f(X_n)\}$ suit une loi géométrique de paramètre $\frac{1}{c}$ et $cg(X_T)U_T$ a pour densité f .

Ainsi, si on veut simuler une v.a. de densité f , il suffit de trouver un réel $c > 0$ et une densité g que l'on sait simuler telles que $f(x) \leq cg(x)$.

DÉMONSTRATION. D'après le lemme précédent le vecteur $(X_n, cg(X_n)U_n)$ suit une loi uniforme sur $A = \{(x, y) \in \mathbb{R}^2, 0 < y < cg(x)\}$. Il suffit donc d'appliquer le premier théorème concernant la méthode du rejet avec $B = \{(x, y) \in \mathbb{R}^2, 0 < y < f(x)\}$, $\lambda(B) = 1$, $\lambda(A) = c$. \square

4. Box Muller

C'est une méthode pour simuler un couple de variables aléatoires indépendantes de loi normale centrée réduite. Elle est basée sur le théorème suivant :

Théorème 6.

Box Muller Soient R et Θ deux variables aléatoires indépendantes telles que r^2 suit une loi exponentielle de paramètre $\frac{1}{2}$ et Θ suit une loi uniforme sur $[0, 2\pi]$. Alors les variables aléatoires $X = R \cos \Theta$ et $Y = R \sin \Theta$ sont indépendantes de loi $\mathcal{N}(0, 1)$.

Il suffit donc pour simuler deux variables aléatoires de loi normale centrée réduite de savoir simuler une loi exponentielle et une loi uniforme.

DÉMONSTRATION. On pose $S = R^2$. Soit f une fonction borélienne bornée.

$\mathbb{E}[f(X, Y)] = \mathbb{E}[f(\sqrt{S} \cos \Theta, \sqrt{S} \sin \Theta)] = \int_{\mathbb{R}^+ \times [0, 2\pi]} f(\sqrt{s} \cos \theta, \sqrt{s} \sin \theta) \frac{\exp(-s/2)}{4\pi} ds d\theta$ En faisant le changement de variables $x = \sqrt{s} \cos \theta, y = \sqrt{s} \sin \theta$ il vient

$$\mathbb{E}[f(X, Y)] = \int_{\mathbb{R}^2} f(x, y) \frac{\exp(-x^2/2)}{\sqrt{2\pi}} \frac{\exp(-y^2/2)}{\sqrt{2\pi}} dx dy$$

On en déduit que X et Y sont indépendantes de loi normale centrée réduite. \square

Ainsi un algorithme pour simuler un couple $[x, y]$ de variables aléatoires normales centrées réduites indépendantes est :

$t = 2 * \%pi * rand(1);$

$s = -2 * log(rand(1));$

$x = sqrt(s) * cos(t); y = sqrt(s) * sin(t);$