



**Université Mohamed BOUDIAF
M'Sila**

**Faculté des Mathématiques et de
l'Informatique.**



**Département de Mathématiques, 3ème Année Licence
Mathématiques Appliquées Semestre 6**

Module : Simulation et pratique de logiciels

Responsable du module :

Dr. BOUHADJAR El Mountasar Billah

Chapitre 2 : Génération d'échantillon suivant différentes lois de probabilités

Année Universitaire : 2024-2025

Table des matières

1	Génération d'échantillon suivant différentes lois de probabilités	3
1.1	Introduction	3
1.2	La méthode de l'inverse	4
1.3	Méthode de rejet	12

Génération d'échantillon suivant différentes lois de probabilités

1.1 Introduction

Dans ce chapitre, nous allons explorer des méthodes pour générer des échantillons aléatoires suivant différentes lois de probabilités. Ces techniques sont essentielles en simulation et en modélisation, car elles permettent de reproduire des comportements aléatoires dans des systèmes complexes.

Nous aborderons trois méthodes principales :

- **La méthode de l'inverse** : Cette méthode, applicable aux cas discret et continu, utilise la fonction de répartition pour générer des échantillons à partir d'une loi de probabilité donnée.
- **La méthode de rejet** : Cette approche permet de générer des échantillons en "rejetant" les valeurs qui ne respectent pas certaines conditions, tout en garantissant que les échantillons retenus suivent la loi souhaitée.
- **La méthode de composition** : Cette méthode combine plusieurs lois de probabilité pour générer des échantillons plus complexes.

À la fin de ce chapitre, vous serez capable de choisir et d'appliquer la méthode la plus adaptée pour générer des échantillons selon une loi de probabilité donnée, une compétence clé en simulation et en analyse statistique.

1.2 La méthode de l'inverse

Fonction de répartition : Soit X une v.a discrète ou continue. La fonction de répartition de X est : F tq : $\forall n \in \mathbb{R}$:

$$\begin{aligned} F(n) &= P(X \leq n) \\ &= \int_{-\infty}^n f(t)dt. \end{aligned}$$

qui vérifie :

- $F(n) \in [0; 1], \forall n \in \mathbb{R}$
- F est croissante sur $\mathbb{R}$
- $\lim_{-\infty} F(n) = 0, \lim_{+\infty} F(n) = 1$
- F est continue à droite.

Exp1 : Soit $X \sim U[a; b]$, alors $f(n) = \begin{cases} 0, & n \notin [a; b] \\ \frac{1}{b-a}, & n \in [a; b] \end{cases} = \frac{1}{b-a} \mathbf{1}_{[a; b]}$

$$F(x) = \begin{cases} \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & x \geq b \end{cases}$$

Exemple 1.2.1. Soit $T \sim U[0, 1]$

$$\begin{aligned} f(t) = I_{[0,1]}(x) &= \begin{cases} 0, & x \notin [0, 1] \\ 1, & x \in [0, 1] \end{cases} \\ F(t) &= \begin{cases} 0, & x < 0 \\ x, & 0 \leq x \leq 1 \\ 1, & x \geq 1 \end{cases} \end{aligned}$$

Remarque 1. Si $X \sim U[a, b]$, alors $T = \frac{X-a}{b-a}$ suit $U[0, 1]$. Inversement, si $T \sim U[0, 1]$, alors $X = a + (b - a)T$ suit $U[a, b]$.

Exemple 1.2.2. si $X \sim \mathcal{N}(0, 1)$, alors la densité X est :

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}, x \in \mathbb{R},$$

la fonction de répartition de F est :

$$F(x) = \int_{-\infty}^x f(t)dt$$

f n'a pas de forme explicite (c-à-d on ne peut pas l'écrire sous forme de fonctions élémentaires).

2.1 Principe de la méthode

Soit X une variable aléatoire réelle de fonction de répartition F . On rappelle que F est croissante (donc en particulier admet un nombre fini ou dénombrable de point de discontinuité). Elle est de plus continue à droite, limitée en tout point à gauche. En général, elle n'est cependant pas bijective. On peut toutefois introduire la notion suivante.

Definition 1.2.3. On définit la fonction inverse généralisée ou fonction quantile F^- de F par

$$F^-(u) = \inf\{x \in \mathbb{R}, F(x) \geq u\}, u \in [0; 1].$$

Cette fonction coïncide avec F^{-1} lorsque F est bijective.

Lemme 1.2.4. Quels que soient $x \in \mathbb{R}$ et $u \in [0; 1]$, on a l'équivalence

$$u \leq F(x) \iff F^-(u) \leq x$$

En particulier, $F \circ F^-(u) \geq u$, avec égalité si et seulement si $u \in F(\mathbb{R})$.

Exemple 1.2.5. Soit F défini comme suit :

$$F(x) = \begin{cases} \frac{x}{3} & \text{si } x \in [0; 1[\\ \frac{x}{3} + \frac{1}{3} & \text{si } x \in [1; 2] \\ 1 & \text{si } x \geq 1 \\ 0 & \text{si } x \leq 0 \end{cases}$$

Alors :

$$F^{-1}(t) = \begin{cases} 3t & \text{si } t \in \left[0; \frac{1}{3}\right] \\ 1 & \text{si } t \in \left[\frac{1}{3}; \frac{2}{3}\right] \\ 3t - 1 & \text{si } t \in \left[\frac{2}{3}; 1\right] \end{cases},$$

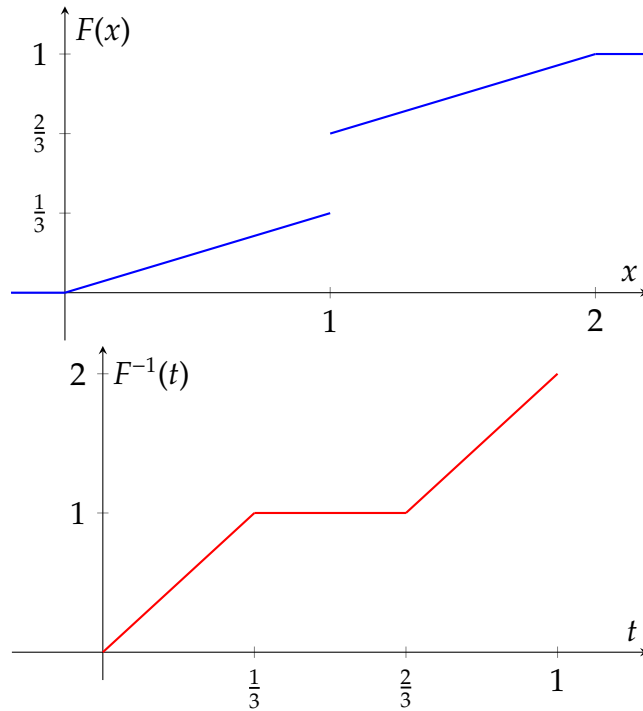


FIGURE 1.1 – Graphiques de $F(x)$ et $F^{-1}(t)$

Simulation par inversion de la fonction de répartition

Programme sur R :

```

finv<-function(u)
{
  if (u<1/3)
  {z=3*u }
  else if (u<2/3)
  {z=1 }
  else
  { z=3*u-1 }
  return(z)
}
u=runif(1,0,1) #simulation d'une variable uniforme (dans [0;1])
# Imprimer le résultat de l'application de finv à u
cat("Valeur simulée de u :", u, "\n" ) # La valeur uniforme aléatoire
cat("Valeur simulée de  $F^{-1}(u)$  :", finv(u), "\n" ) # Le résultat

```

Programme sur Matlab :

```

    u = rand(); % Générer une variable uniforme aléatoire
    fprintf('Valeur simulée de u : %f\n', u);
    % Compute finv
    if u < 1/3
        z = 3 * u;
    elseif u < 2/3
        z = 1;
    else
        z = 3 * u - 1;
    end

```

```

    % Display the result
    fprintf('Valeur simulée de  $F^{(-1)}(u)$  : %f\n', z);

```

Proposition 1.2.6 (Méthode d'inversion pour les variables aléatoires discrètes). Soient U une variable aléatoire suivant la loi uniforme sur $[0, 1]$ et X une variable aléatoire discrète de fonction de répartition F_X . On note $X(\Omega) = \{x_1, x_2, x_3, \dots\}$ avec $x_1 < x_2 < x_3 < \dots$, on a alors $\forall k \geq 2$:

- $p_1 := \mathbb{P}(X = x_1) = \mathbb{P}(U \leq F_X(x_1))$
- $p_k := \mathbb{P}(X = x_k) = \mathbb{P}(F_X(x_{k-1}) < U \leq F_X(x_k))$

Preuve 1. Comme pour tout k , $F_X(x_k) \in [0, 1]$, alors d'après la fonction de répartition de la loi uniforme

$$\mathbb{P}(U \leq F_X(x_1)) = F_X(x_1)$$

- Puisque $p_1 = F_X(x_1)$, on a

$$p_1 = \mathbb{P}(U \leq F_X(x_1))$$

- Puisque pour tout $k \geq 2$, $\mathbb{P}(X = x_k) = F_X(x_k) - F_X(x_{k-1})$, on a également

$$\mathbb{P}(X = x_k) = \mathbb{P}(U \leq F_X(x_k)) - \mathbb{P}(U \leq F_X(x_{k-1})) = \mathbb{P}(F_X(x_{k-1}) < U \leq F_X(x_k))$$

Exemple 1.2.7. On se propose de simuler la variable aléatoire X suivant la loi de Bernoulli de paramètre $p \in]0, 1[$.

Solution. D'après la proposition précédente, si U est une variable aléatoire suivant la loi uniforme sur $[0, 1]$, on a :

$$\mathbb{P}(X = 1) = \mathbb{P}(F_X(0) < U \leq F_X(1)) = \mathbb{P}(1 - p < U \leq 1) = \mathbb{P}(U > 1 - p)$$

La probabilité de l'événement $[X = 1]$ est la même que celle de l'événement $[U > 1 - p]$. On peut donc proposer la simulation suivante :

Programme sur Matlab

```
% Demander à l'utilisateur la valeur de p
p = input('Entrez la valeur de p (0 < p < 1) : ');
% Simuler la variable aléatoire de Bernoulli X
if rand(1, 1) > 1 - p
X = 1;
else
X = 0;
end
% Afficher le résultat
disp(['La valeur de X est : ', num2str(X)]);
```

Programme sur R

```
p <- as.numeric(readline(prompt = "Entrez la valeur de p (0 < p < 1) : "))
if (runif(1, 0, 1) > 1 - p) {
X <- 1
} else {
X <- 0
}
cat("La valeur de X est :", X, "\n")
```

exercice 1.2.8. Utilisez la méthode d'inversion pour simuler une variable aléatoire X suivant une loi géométrique de paramètre $p \in]0, 1[$.

Solution Pour une loi géométrique, la fonction de répartition est donnée par

$$F_X(k) = \sum_{i=1}^k q^{i-1} p = p \sum_{i=0}^{k-1} q^i = 1 - (1 - p)^k$$

D'après la proposition précédente, si X est une variable aléatoire suivant la loi uniforme sur $[0, 1]$, on a pour n :

$$p_k = \mathbb{P}(F_X(k-1) < U \leq F_X(k)) = \mathbb{P}(1 - q^{k-1} < U \leq 1 - q^k)$$

On a la même formule pour $k = 1$,

$$p_1 = \mathbb{P}(0 < U \leq F_X(1)) = \mathbb{P}(1 - q^{1-1} < U \leq 1 - q^1).$$

L'événement $[X = k]$ est donc réalisé si, et seulement si, k est le plus petit entier tel que U est inférieur ou égal à $1 - q^k$. Tant que U dépasse $1 - q^k$, on ajoute 1. On peut donc proposer la solution en pseudo-code :

1. Demander à l'utilisateur de saisir la valeur de p .
2. Initialiser $k \leftarrow 1$ et générer $U \sim \mathcal{U}([0, 1])$.
3. Tant que $U > 1 - (1 - p)^k$, incrémenter k .
4. Afficher la valeur finale de k , qui correspond à X .

En **MATLAB**, ce programme peut s'écrire comme suit :

```
p = input('Entrez la valeur de p (0 < p < 1) : ');
k = 1;
u = rand(1, 1);
while u > 1 - (1 - p)^k
    k = k + 1;
end
disp(['La valeur de X est : ', num2str(k)]);
```

Remarque 2. Une autre manière de simuler la loi géométrique On peut poursuivre le calcul, ce qui donnerait pour $k \in \mathbb{N}^*$

$$\begin{aligned} \mathbb{P}(X = k) &= \mathbb{P}(1 - q^{k-1} < U \leq 1 - q^k) \\ &= \mathbb{P}(q^{k-1} > 1 - U \geq q^k) \\ &= \mathbb{P}((k-1)\ln(q) > \ln(1-U) \geq k\ln(q)) \\ &= \mathbb{P}\left(k-1 < \frac{\ln(1-U)}{\ln(q)} \leq k\right) \text{ car } \ln(q) < 0 \\ &= \mathbb{P}\left(\left\lfloor \frac{\ln(1-U)}{\ln(q)} \right\rfloor = k-1\right) \\ &= \mathbb{P}\left(\left\lfloor \frac{\ln(1-U)}{\ln(q)} \right\rfloor + 1 = k\right) \end{aligned}$$

On peut donc proposer la simulation suivante :

En Matlab :

```
p = input('Entrez la valeur de p (0 < p < 1) : ');
X = floor(log(1 - rand(1, 1)) / log(1 - p)) + 1;
disp(['La valeur de X est : ', num2str(X)]);
```

En R :

```
p <- as.numeric(readline(prompt="Entrez la valeur de p (0 < p < 1):"))
X <- floor(log(1 - runif(1)) / log(1 - p)) + 1
cat("La valeur de X est :", X, "\n")
```

Proposition 1.2.9. *Si F est continue et strictement croissante sur $\mathbb{R}$ et vérifie $\lim_{x \rightarrow -\infty} F(x) = 0$, $\lim_{x \rightarrow +\infty} F(x) = 1$, et $U \sim \mathcal{U}[0, 1]$, alors $X = F^{-1}(U)$ possède la densité f .*

Preuve 2. *Soit $a < b \in \mathbb{R}$. Comme F est strictement croissante, $\{a < X \leq b\} = \{a < F^{-1}(U) \leq b\} = \{F(a) < U \leq F(b)\}$. Donc*

$$\mathbb{P}(a < X \leq b) = \int_0^1 1_{\{F(a) < u \leq F(b)\}} du = F(b) - F(a) = \int_a^b f(y) dy.$$

Ce qui signifie que X possède la densité f .

Exemple 1.2.10. *La loi de Cauchy de paramètre $a > 0$ est la loi de densité $f(y) = \frac{a}{\pi(y^2 + a^2)}$ sur $\mathbb{R}$. La fonction de répartition associée est*

$$\begin{aligned} F(x) &= \int_{-\infty}^x f(y) dy = \frac{1}{\pi} \int_{-\infty}^x \frac{1}{(y/a)^2 + 1} \frac{dy}{a} \\ &= \frac{1}{\pi} \int_{-\infty}^{x/a} \frac{1}{1 + z^2} dz = \frac{1}{\pi} [\arctan(z)]_{-\infty}^{x/a} = \frac{1}{\pi} \arctan(x/a) + 1/2 \end{aligned}$$

Pour inverser la fonction de répartition, on se donne $u \in]0, 1[$ et on cherche x t.q. $F(x) = u$:

$$F(x) = u \Leftrightarrow x = a \tan(\pi(u - 1/2))$$

En conséquence si $U \sim \mathcal{U}[0, 1]$, $X = a \tan(\pi(U - 1/2)) \sim C(a)$. Comme $\pi(U - 1/2) \sim \mathcal{U}\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ et comme la fonction $\tan$ est périodique de période π , pour $f : \mathbb{R} \rightarrow \mathbb{R}$ bornée,

$$\begin{aligned}\mathbb{E}(f(\tan(\pi(U - 1/2)))) &= \frac{1}{\pi} \int_{-\pi/2}^0 f(\tan(\pi + x))dx + \frac{1}{\pi} \int_0^{\pi/2} f(\tan x)dx \\ &= \frac{1}{\pi} \int_0^{\pi} f(\tan x)dx = \mathbb{E}(f(\tan(\pi U)))\end{aligned}$$

Ainsi, $\tan(\pi(U - 1/2)) \stackrel{d}{=} \tan(\pi U)$ et

$$X = a \tan(\pi U) \sim C(a)$$

exercice 1.2.11. Utilisez la méthode d'inversion pour simuler une variable aléatoire suivant une loi exponentielle de paramètre $\lambda > 0$, dont la densité est donnée par :

$$\lambda e^{-\lambda z} 1_{\{z>0\}}.$$

Solution La fonction de répartition associée est

$$F(x) = \int_{-\infty}^x \lambda e^{-\lambda z} 1_{\{z>0\}} dz = 1_{\{x>0\}} (1 - e^{-\lambda x})$$

Pour $u \in]0, 1[$, on a

$$F(x) = u \Leftrightarrow x = -\frac{1}{\lambda} \ln(1 - u)$$

On en déduit que si $U \sim \mathcal{U}[0, 1]$, $-\frac{1}{\lambda} \ln(1 - U) \sim \mathcal{E}(\lambda)$. On peut apporter une simplification en remarquant que $1 - U \stackrel{\mathcal{L}}{=} U$ ($1 - U$ a même loi que U), ce qui entraîne que $-\frac{1}{\lambda} \ln(1 - U) \stackrel{\mathcal{L}}{=} -\frac{1}{\lambda} \ln(U)$ et permet de conclure que

$$S = -\frac{1}{\lambda} \ln(U) \sim \mathcal{E}(\lambda)$$

Programme sur Matlab

```
function Y = rexp(m, n, lambda)
    % Générer une matrice m*n de variables aléatoires
    exponentielles
    Y = -log(rand(m, n)) / lambda;
end
```

Programme sur R

```

rexp\_matrix <- function(m, n, lambda) {
  # Générer une matrice m $ \times $ n de v.a exponentielles
  Y <- -log(matrix(runif(m * n), nrow = m, ncol = n)) / lambda
  return(Y)
}
result <- rexp\_matrix(3, 4, 2)
print(result).

```

1.3 Méthode de rejet

On veut simuler une variable de densité f (dans $\mathbb{D}$) et on suppose qu'il existe une loi de densité g facile à simuler telle que

$$\forall x \in \mathbb{D}, f(x) \leq kg(x)$$

avec k une constante. Posons $\alpha(x) = \frac{f(x)}{kg(x)}$ sur l'ensemble $\{x, g(x) > 0\}$.

Proposition 1.3.1. Soit $(X_n, U_n)_{n \geq 1}$ une suite de variable i.i.d. avec pour tout n , X_n est de loi de densité g et est indépendante de U_n , qui est de loi uniforme sur $[0; 1]$. Soit $T = \inf \{n, U_n \leq \alpha(X_n)\}$ (T est un entier aléatoire). Soit $X = X_T$.

Alors la v.a. X ainsi simulée est de loi de densité f .

Preuve 3. Remarquons que $\alpha(X_n)$ est défini pour presque tout ω . En effet, $\mathbb{P}(X_n \in \{x : g(x) = 0\}) = 0$.

Soit $\varphi \in C_b^+(\mathbb{D})$ (l'ensemble des fonctions continues bornées positives). Nous avons $\varphi(X_T) = \sum_{n=1}^{+\infty} \varphi(X_n) \mathbf{1}_{n=T}$ (pour tout ω , cette somme contient au plus un terme non nul).

$$\begin{aligned}
\mathbb{E}(\varphi(X)) &= \sum_{n=1}^{+\infty} \mathbb{E}(\varphi(X_n) \mathbf{1}_{n=T}) \\
&= \sum_{n=1}^{+\infty} \mathbb{E} \left(\varphi(X_n) \prod_{i=1}^{n-1} \mathbf{1}_{U_i > \alpha(X_i)} \times \mathbf{1}_{U_n \leq \alpha(X_n)} \right) \\
&= \sum_{n=1}^{+\infty} \mathbb{E} \left(\varphi(X_n) \mathbf{1}_{U_n \leq \alpha(X_n)} \right) \prod_{i=1}^{n-1} \mathbb{E}(\mathbf{1}_{U_i > \alpha(X_i)}).
\end{aligned}$$

La dernière étape est obtenue par indépendance des (X_i, U_i) .

Calculons, pour tout n ,

$$\begin{aligned}\mathbb{E}(\varphi(X_n) \mathbf{1}_{U_n \leq \alpha(X_n)}) &= \int_{x \in \mathbb{D}} \left(\int_{u \in [0;1]} \mathbf{1}_{u \leq \alpha(x)} du \right) \varphi(x) g(x) dx \\ &= \int_{x \in \mathbb{D}} \frac{f(x)}{kg(x)} \varphi(x) g(x) dx \\ &= \frac{1}{k} \int_{x \in \mathbb{D}} \varphi(x) f(x) dx.\end{aligned}$$

Nous en déduisons, pour tout n ,

$$\begin{aligned}\mathbb{E}(\mathbf{1}_{U_n > \alpha(X_n)}) &= 1 - \mathbb{E}(\mathbf{1}_{U_n \leq \alpha(X_n)}) \\ &= 1 - \frac{1}{k};\end{aligned}$$

Remarquons que l'inégalité $f \leq kg$ implique

$$\int_{\mathbb{D}} f(x) dx \leq k \int_{\mathbb{D}} g(x) dx,$$

donc $k \geq 1$. Nous avons donc

$$\begin{aligned}E(\varphi(X)) &= \sum_{n=1}^{+\infty} \int_{\mathbb{D}} \varphi(x) f(x) dx \times \frac{1}{k} \left(1 - \frac{1}{k}\right)^{n-1} \\ &= \int_{\mathbb{D}} \varphi(x) f(x) dx \times \frac{1}{k} \times \frac{1}{1 - \left(1 - \frac{1}{k}\right)} \\ &= \int_{\mathbb{D}} \varphi(x) f(x) dx.\end{aligned}$$

D'où le résultat.

Exemple 1.3.2. Soit $f(x) = \frac{e^{-x^3}}{Z} \mathbf{1}_{x \geq 1}$ avec $Z = \int_1^{\infty} e^{-x^3} dx$ (la constante Z n'est pas connue mais nous n'en aurons pas besoin). Soit $g(x) = \frac{1}{x^2} \mathbf{1}_{x \geq 1}$. Les fonctions f et g sont des densités de probabilité. Il est facile de simuler suivant la loi de densité g en utilisant la technique d'inversion de la fonction de répartition.

Nous commençons par étudier la fonction $h : x \geq 1 \mapsto x^2 e^{-x^3}$. Nous avons $h'(x) = (2x - 3x^4) e^{-x^3} \leq 0$ pour tout $x \geq 1$. Donc, pour tout $x \geq 1$, $h(x) < h(1) = e^{-1}$. Nous en déduisons (pour $x \geq 1$)

$$f(x) \leq \frac{1}{eZ}g(x).$$

Nous posons $k = 1/(eZ)$. Soit

$$\alpha(x) = \frac{f(x)}{kg(x)} = x^2 e^{-x^3+1}.$$

Nous ne connaissons pas Z mais nous pouvons calculer $\alpha(x)$ pour tout $x \geq 1$ donc nous pouvons utiliser l'algorithme de simulation par rejet pour simuler suivant la densité f (voir le programme suivant).

Programme sur R

Méthode 1

```
b <- 0
while (b == 0) {
  u <- runif(1, 0, 1)
  v <- runif(1, 0, 1)
  y <- 1 / (1 - v)
  if (u <= y^2 * exp(-y^3) * exp(1)) {
    b <- 1
  }
}
print(y)
```

par inversion de la fonction de répartition :

$$\text{pour } x \geq 1, G(x) = \int_1^x g(u)du = \left[-\frac{1}{u}\right]_1^x = 1 - \frac{1}{x}$$

$$\text{pour } u \in [0; 1], G^{-1}(u) = \frac{1}{1-u}.$$

Méthode 2

```
while (TRUE) {
  u <- runif(1)
  v <- runif(1)
  y <- 1 / (1 - v)
  if (u <= y^2 * exp(-y^3) * exp(1)) {
    break
  }
}
print(y)
```