

## Chapitre 3: Transformations

Tasnime Hamdeni

February 15, 2024

- 1 Introduction
- 2 Sommes de variables aléatoires
- 3 Algorithmes de Box-Muller
- 4 Simulation d'un vecteur gaussien
- 5 Simulation d'un mouvement Brownien
- 6 Méthodes d'acceptation-rejet
- 7 Conclusion:

# 1- Introduction

# 1- Introduction

Lorsque  $F^{-1}$  n'a pas d'expression explicite ou lorsque l'on cherche des méthodes de simulation alternatives plus rapide, on peut essayer de trouver des transformations entre une loi et des lois dont on dispose déjà de générateurs.

## 2- Sommes de variables aléatoires

## 2- Sommes de variables aléatoires

### 2-1 Simulation d'une v.a. suivant la loi binomiale

Un exemple concret de cette approche se trouve dans la simulation d'une variable aléatoire  $X$  suivant une loi binomiale  $\mathcal{B}(n, p)$  à l'aide de la méthode de la fonction inverse.

## 2- Sommes de variables aléatoires

### 2-1 Simulation d'une v.a. suivant la loi binomiale

Un exemple concret de cette approche se trouve dans la simulation d'une variable aléatoire  $X$  suivant une loi binomiale  $\mathcal{B}(n, p)$  à l'aide de la méthode de la fonction inverse.

Cependant, cette méthode implique le calcul de la fonction de répartition, incluant des coefficients binomiaux et des puissances de  $p$  et  $1 - p$ , ce qui peut être **coûteux en termes de calculs**.

## 2- Sommes de variables aléatoires

### 2-1 Simulation d'une v.a. suivant la loi binomiale

Un exemple concret de cette approche se trouve dans la simulation d'une variable aléatoire  $X$  suivant une loi binomiale  $\mathcal{B}(n, p)$  à l'aide de la méthode de la fonction inverse.

Cependant, cette méthode implique le calcul de la fonction de répartition, incluant des coefficients binomiaux et des puissances de  $p$  et  $1 - p$ , ce qui peut être **coûteux en termes de calculs**.





## 2- Sommes de variables aléatoires

### 2-1 Simulation d'une v.a. suivant la loi binomiale

#### Définition (Rappel)

Une définition de la loi binomiale, de paramètres  $n$  et  $p$ , est la loi de probabilité d'une variable aléatoire  $X$  telle que :

$$X = Y_1 + Y_2 + \cdots + Y_n,$$

où  $Y_1, Y_2, \dots, Y_n$ , sont des variables aléatoires indépendantes de loi de Bernoulli de même paramètre  $p$ .

## 2- Sommes de variables aléatoires

### 2-1 Simulation d'une v.a. suivant la loi binomiale

#### Définition (Rappel)

Une définition de la loi binomiale, de paramètres  $n$  et  $p$ , est la loi de probabilité d'une variable aléatoire  $X$  telle que :

$$X = Y_1 + Y_2 + \cdots + Y_n,$$

où  $Y_1, Y_2, \dots, Y_n$ , sont des variables aléatoires indépendantes de loi de Bernoulli de même paramètre  $p$ .

En utilisant la méthode de la fonction inverse pour générer des variables de Bernoulli, où  $U_1, \dots, U_n$  sont des variables aléatoires i.i.d. suivant la loi uniforme  $\mathcal{U}([0, 1])$ , on peut générer la somme

$$X_n = \sum_{i=1}^n 1_{\{U_i \leq p\}} \sim \mathcal{B}(n, p).$$

## 2- Sommes de variables aléatoires

### 1-2 Simulation d'une v.a. suivant la loi de Poisson

La simulation d'une loi de Poisson à partir de lois exponentielles est basée sur le résultat suivant :

#### Théorème

Soit  $(E_i)_{i \geq 1}$  une suite de variables aléatoires indépendantes et de même loi exponentielle de paramètre  $\lambda$ . On pose  $S_1 = E_1$  et pour  $n \geq 2$ ,  $S_n = E_1 + \dots + E_n$ . Alors, pour tout  $n \geq 1$ ,

$$\mathbb{P}(S_n \leq 1 < S_{n+1}) = \frac{e^{-\lambda} \lambda^n}{n!}.$$

S

## 2- Sommes de variables aléatoires

### 1-2 Simulation d'une v.a. suivant la loi de Poisson

- Soit  $U_1, U_2, \dots, U_n$  des v.a. uniformes indépendantes sur l'intervalle  $[0, 1]$ .
- Utilisez **la méthode de la transformée inverse** : pour chaque  $i$ ,  $E_i = -\frac{1}{\lambda} \ln(U_i)$  suit une loi exponentielle de paramètre  $\lambda$ .
- On a ainsi

$$S_n = -\frac{1}{\lambda} \ln \left( \prod_{i=1}^n U_i \right)$$

En notant  $P_n := \prod_{i=1}^n U_i$ ,  $n \geq 1$  on a

$$S_n \leq 1 < S_{n+1} \Leftrightarrow P_n \geq e^{-\lambda} > P_{n+1} \quad \forall n \in \mathbb{N}^*$$

- Trouvez la plus petite valeur de  $n$  telle que  $S_n$  dépasse un nombre donné  $x$  (où  $x$  est un entier positif pour simuler une loi de Poisson).
- La valeur de  $n$  obtenue dans l'étape précédente est la réalisation de la variable de Poisson.

---

**Algorithm 1** PoissonSimulation( $\lambda$ )

---

1: $k \leftarrow 0$	▷ Initialisation du compteur d'événements
2: $p \leftarrow 1$	▷ Initialisation de la probabilité accumulée
3: <b>while</b> $p > \exp(-\lambda)$ <b>do</b>	
4: $u \leftarrow \text{UniformRandom}()$	▷ Générer une variable uniforme sur $[0, 1]$
5: $p \leftarrow p \times u$	▷ Mettre à jour la probabilité accumulée
6: $k \leftarrow k + 1$	▷ Incrémenter le compteur d'événements
7: <b>end while</b>	
8: <b>return</b> $k - 1$	▷ Renvoyer le nombre d'événements générés - 1

---

## 2- Algorithmes de Box-Muller

## 2- Algorithmes de Box-Muller

L'algorithme de Box-Muller est une méthode pour générer des paires de variables aléatoires indépendantes et identiquement distribuées (i.i.d.) suivant une distribution normale standard ( $\mathcal{N}(0, 1)$ ) à partir de paires de variables aléatoires uniformes indépendantes  $\mathcal{U}(0, 1)$ . Il existe deux versions de cet algorithme:

## 2- Algorithmes de Box-Muller

L'algorithme de Box-Muller est une méthode pour générer des paires de variables aléatoires indépendantes et identiquement distribuées (i.i.d.) suivant une distribution normale standard ( $\mathcal{N}(0, 1)$ ) à partir de paires de variables aléatoires uniformes indépendantes  $\mathcal{U}(0, 1)$ . Il existe deux versions de cet algorithme:

### Proposition: Algorithme de Box-Muller Version n°1

Soient  $U_1$  et  $U_2$  des variables i.i.d. de loi  $U([0, 1])$ . Alors les variables  $X_1$  et  $X_2$  définies ci-dessous sont i.i.d. de loi  $\mathcal{N}(0, 1)$  :

$$X_1 \equiv \sqrt{-2 \ln(U_1)} \cos(2\pi U_2)$$

$$X_2 \equiv \sqrt{-2 \ln(U_1)} \sin(2\pi U_2)$$



## 2- Algorithme de Box-Muller

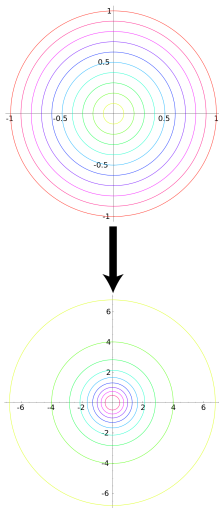
### Proposition: Algorithmes de Box-Muller Version n°2

Soit  $(U_1, U_2)$  une variable aléatoire de loi uniforme sur le disque unité  $D_1 \equiv \{(u_1, u_2) \in \mathbb{R}^2 : u_1^2 + u_2^2 \leq 1\}$ . Alors les variables  $X_1$  et  $X_2$  définies ci-dessous sont i.i.d. de loi  $N(0, 1)$  :

$$X_1 \equiv U_1 \sqrt{\frac{-2 \ln(\sqrt{U_1^2 + U_2^2})}{U_1^2 + U_2^2}}$$

$$X_2 \equiv U_2 \sqrt{\frac{-2 \ln(\sqrt{U_1^2 + U_2^2})}{U_1^2 + U_2^2}}$$

## 2- Algorithmes de Box-Muller



### Illustration graphique de la transformation:

les cercles initiaux, uniformément répartis autour de l'origine, se transforment en un ensemble de nouveaux cercles centrés, dont la densité est plus élevée près de l'origine et diminue rapidement en s'éloignant.

Les cercles de départ les plus grands correspondent aux cercles d'arrivée les plus petits, et réciproquement.

## 2- Algorithmes de Box-Muller

### Démonstration de l'algorithme de Box-Muller

# 3- Simulation d'un vecteur gaussien

### 3- Simulation d'un vecteur gaussien

La simulation d'un vecteur gaussien peut être réalisée à l'aide d'un générateur de la loi  $\mathcal{N}(0, 1)$ , permettant ainsi de simuler un vecteur aléatoire  $X = (X_1, \dots, X_d)$  dans  $\mathbb{R}^d$  suivant une loi normale multivariée  $\mathcal{N}(\mu, \Sigma)$ . Ainsi, on distingue deux cas:

### 3- Simulation d'un vecteur gaussien

La simulation d'un vecteur gaussien peut être réalisée à l'aide d'un générateur de la loi  $\mathcal{N}(0, 1)$ , permettant ainsi de simuler un vecteur aléatoire  $X = (X_1, \dots, X_d)$  dans  $\mathbb{R}^d$  suivant une loi normale multivariée  $\mathcal{N}(\mu, \Sigma)$ . Ainsi, on distingue deux cas:

#### 1<sup>er</sup> cas: Matrice de covariance $\Sigma$ diagonale

Si la matrice de covariance  $\Sigma$  est diagonale, il suffit de générer  $d$  variables  $X_i$ ,  $i \in \{1, \dots, d\}$ , indépendantes, chacune suivant une loi normale  $\mathcal{N}(\mu_i, \sigma_i^2)$ . Pour ce faire, on simule  $Z_1, \dots, Z_d$  variables aléatoires i.i.d. suivant la loi  $\mathcal{N}(0, 1)$  et on utilise la transformation  $X_i \equiv \mu_i + \sqrt{\sigma_i^2} \cdot Z_i$ .

### 3- Simulation d'un vecteur gaussien

La simulation d'un vecteur gaussien peut être réalisée à l'aide d'un générateur de la loi  $\mathcal{N}(0, 1)$ , permettant ainsi de simuler un vecteur aléatoire  $X = (X_1, \dots, X_d)$  dans  $\mathbb{R}^d$  suivant une loi normale multivariée  $\mathcal{N}(\mu, \Sigma)$ . Ainsi, on distingue deux cas:

#### 1<sup>er</sup> cas: Matrice de covariance $\Sigma$ diagonale

Si la matrice de covariance  $\Sigma$  est diagonale, il suffit de générer  $d$  variables  $X_i$ ,  $i \in \{1, \dots, d\}$ , indépendantes, chacune suivant une loi normale  $\mathcal{N}(\mu_i, \sigma_i^2)$ . Pour ce faire, on simule  $Z_1, \dots, Z_d$  variables aléatoires i.i.d. suivant la loi  $\mathcal{N}(0, 1)$  et on utilise la transformation  $X_i \equiv \mu_i + \sqrt{\sigma_i^2} \cdot Z_i$ .

#### 2<sup>ème</sup> cas: Matrice de covariance $\Sigma$ non diagonale

si la matrice de covariance  $\Sigma$  n'est pas diagonale, la méthode précédente n'est pas applicable. Dans ce cas, on recourt à la décomposition de Cholesky de  $\Sigma$ .

### 3- Simulation d'un vecteur gaussien

#### Rappel: Théorème de décomposition de Cholesky

Soit  $A$  une matrice symétrique définie positive. Alors il existe une matrice triangulaire inférieure  $L$  tel que  $A = L^T L$ . De plus, si on impose aux coefficients diagonaux de  $L$  d'être positifs, cette factorisation est unique.



### 3- Simulation d'un vecteur gaussien

#### Rappel: Théorème de décomposition de Cholesky

Soit  $A$  une matrice symétrique définie positive. Alors il existe une matrice triangulaire inférieure  $L$  tel que  $A = L^T L$ . De plus, si on impose aux coefficients diagonaux de  $L$  d'être positifs, cette factorisation est unique.

#### Proposition:

Soit  $Z$  un vecteur aléatoire de loi  $\mathcal{N}(0_d, I_d)$ . Étant donnée la décomposition de Cholesky de  $\Sigma$ , i.e.,  $\Sigma \equiv LL^T$  avec  $L$  une matrice triangulaire inférieure, la variable définie par  $X \equiv \mu + LZ$ ,  $\mu \in \mathbb{R}^d$ , suit une loi normale multivariée  $\mathcal{N}(\mu, \Sigma)$ .

### 3- Simulation d'un vecteur gaussien

#### Remarques

- 1  $Z$  peut être simulé en utilisant l'algorithme de Box-Muller (les composantes sont i.i.d. suivant  $\mathcal{N}(0, 1)$ ).

### 3- Simulation d'un vecteur gaussien

#### Remarques

- 1  $Z$  peut être simulé en utilisant l'algorithme de Box-Muller (les composantes sont i.i.d. suivant  $\mathcal{N}(0, 1)$ ).
- 2 La fonction `L = np.linalg.cholesky(A)` de python permet d'obtenir la décomposition de Cholesky d'une matrice symétrique définie-positive.

### 3- Simulation d'un vecteur gaussien

#### Exemple:

En dimension 2, soit  $Z \equiv (Z_1, Z_2) \sim \mathcal{N}(0_2, I_2)$ ,  $\rho \in [-1, 1]$  une constante.  
En posant,

$$X_1 \equiv \mu_1 + \sigma_1 Z_1$$

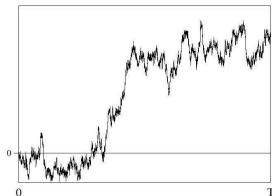
$$X_2 \equiv \mu_2 + \sigma_2 \left( \rho Z_1 + \sqrt{1 - (\rho)^2} Z_2 \right),$$

on a

$$X \equiv (X_1, X_2) \sim \mathcal{N} \left( \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \sigma_1^2 & \rho \sigma_1 \sigma_2 \\ \rho \sigma_1 \sigma_2 & \sigma_2^2 \end{bmatrix} \right).$$

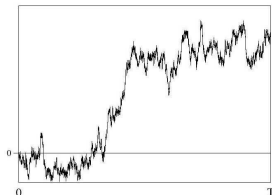
# 4- Simulation d'un mouvement Brownien

## 4- Simulation d'un mouvement Brownien



Le mouvement brownien, ou processus de Wiener est un processus stochastique.

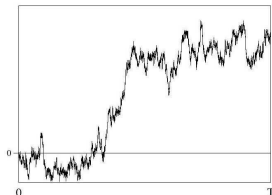
## 4- Simulation d'un mouvement Brownien



Le mouvement brownien, ou processus de Wiener est un processus stochastique.

Décrit pour la première fois en 1827 par le botaniste Robert Brown.

## 4- Simulation d'un mouvement Brownien



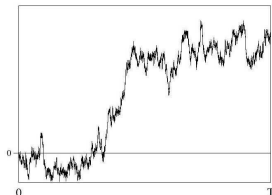
Le mouvement brownien, ou processus de Wiener est un processus stochastique.

Décrit pour la première fois en 1827 par le botaniste Robert Brown.

On ne s'intéresse qu'à la simulation.



## 4- Simulation d'un mouvement Brownien



Le mouvement brownien, ou processus de Wiener est un processus stochastique.

Décrit pour la première fois en 1827 par le botaniste Robert Brown.

On ne s'intéresse qu'à la simulation.

Basé sur la génération de variables aléatoires gaussiennes.

## 4- Simulation d'un mouvement Brownien

### Définition: Mouvement Brownien

Un processus stochastique  $B=(B_t)_{t\in\mathbb{R}}$  est un mouvement brownien standard si, et seulement si,

- 1  $B$  est issu de 0,

## 4- Simulation d'un mouvement Brownien

### Définition: Mouvement Brownien

Un processus stochastique  $B=(B_t)_{t\in\mathbb{R}}$  est un mouvement brownien standard si, et seulement si,

- 1  $B$  est issu de 0,
- 2  $B$  a des trajectoires continues,

## 4- Simulation d'un mouvement Brownien

### Définition: Mouvement Brownien

Un processus stochastique  $B=(B_t)_{t\in\mathbb{R}}$  est un mouvement brownien standard si, et seulement si,

- ①  $B$  est issu de 0,
- ②  $B$  a des trajectoires continues,
- ③  $\forall 0 = t_0 \leq t_1 \leq \dots \leq t_n$ , les variables

$$B_{t_1}, B_{t_2} - B_{t_1}, \dots, B_{t_n} - B_{t_{n-1}}$$

sont indépendantes et on dit que  $B$  est à accroissements indépendants.

## 4- Simulation d'un mouvement Brownien

### Définition: Mouvement Brownien

Un processus stochastique  $B=(B_t)_{t\in\mathbb{R}}$  est un mouvement brownien standard si, et seulement si,

- ①  $B$  est issu de 0,
- ②  $B$  a des trajectoires continues,
- ③  $\forall 0 = t_0 \leq t_1 \leq \dots \leq t_n$ , les variables

$$B_{t_1}, B_{t_2} - B_{t_1}, \dots, B_{t_n} - B_{t_{n-1}}$$

sont indépendantes et on dit que  $B$  est à accroissements indépendants.

- ④  $B_t - B_s \sim \mathcal{N}(0, t - s)$  c'est-à-dire que, pour tous  $s \leq t$ , la valeur de  $B_t - B_s$  ne dépend que de la valeur de  $t - s$ .

On dit que  $B$  a des accroissements stationnaires

## 4- Simulation d'un mouvement Brownien

### Forward simulation

Soit  $(t_n)_{n \in \mathbb{N}}$  une suite de  $\mathbb{R}_+$  et  $(Z_n)_{n \in \mathbb{N}}$  une suite de variables i.i.d. de loi  $\mathcal{N}(0, 1)$ . On définit le processus  $B = (B_t)_{t \in \mathbb{R}_+}$  par

$$B_0 = 0 \quad \text{et} \quad B_{t_{n+1}} = B_{t_n} + \sqrt{t_{n+1} - t_n} Z_n.$$

Alors,  $B$  est une réalisation de trajectoires du mouvement brownien aux instants  $(t_n)_{n \in \mathbb{N}}$ .

## 4- Simulation d'un mouvement Brownien

### Modèle de Black-Scholes

En finance, un exemple classique d'utilisation du mouvement brownien  $(B_t)_{t \in \mathbb{R}_+}$  est le calcul de la valeur d'une option avec le modèle de Black-Scholes, c'est-à-dire le processus  $(S_t)_{t \in \mathbb{R}}$  défini par

$$S_t = S_0 \exp \left( \left( \frac{r - \sigma^2}{2} \right) t + \sigma B_t \right)$$

avec

- $r$  : le taux d'intérêt sans risque,
- $\sigma$  : la volatilité du prix de l'option.

# 5- Méthodes d'acceptation-rejet



## 5- Méthode d'acceptation-rejet

### Proposition: Méthode d'acceptation-rejet

Soit  $X=(X_1, \dots, X_d)$  une variable aléatoire de densité  $f$  sur  $\mathbb{R}^d$  et  $g$  une densité sur  $\mathbb{R}^d$  telle qu'il existe une constante  $c \geq 1$  telle que

$$f(x) \leq cg(x), \quad \text{pour tout } x \text{ dans } \mathbb{R}^d.$$

Soient  $(U_n)_{n \geq 1}$  une suite de variables i.i.d. de loi  $\mathcal{U}([0, 1])$  et  $(Y_n)_{n \geq 1} = (Y_{1,n}, \dots, Y_{d,n})_{n \geq 1}$  une suite de variables i.i.d. de loi de densité  $g$  telles que ces deux suites soient indépendantes.

Alors, pour  $T$  défini par:  $T = \inf\{n \geq 1 : U_n \leq \frac{f(Y_n)}{cg(Y_n)}\}$ ,  $Y_T$  suit la loi de densité  $f$ .

Autrement dit, pour simuler  $X \sim f$ , il suffit de simuler  $Y \sim g$  et  $U \sim \mathcal{U}([0, 1])$  jusqu'à ce que  $ucg(y) < f(y)$ .

## 5- Méthode d'acceptation-rejet

### Proposition: Méthode d'acceptation-rejet

Soit  $X=(X_1, \dots, X_d)$  une variable aléatoire de densité  $f$  sur  $\mathbb{R}^d$  et  $g$  une densité sur  $\mathbb{R}^d$  telle qu'il existe une constante  $c \geq 1$  telle que

$$f(x) \leq cg(x), \quad \text{pour tout } x \text{ dans } \mathbb{R}^d.$$

Soient  $(U_n)_{n \geq 1}$  une suite de variables i.i.d. de loi  $\mathcal{U}([0, 1])$  et  $(Y_n)_{n \geq 1} = (Y_{1,n}, \dots, Y_{d,n})_{n \geq 1}$  une suite de variables i.i.d. de loi de densité  $g$  telles que ces deux suites soient indépendantes.

Alors, pour  $T$  défini par:  $T = \inf\{n \geq 1 : U_n \leq \frac{f(Y_n)}{cg(Y_n)}\}$ ,  $Y_T$  suit la loi de densité  $f$ .

Autrement dit, pour simuler  $X \sim f$ , il suffit de simuler  $Y \sim g$  et  $U \sim \mathcal{U}([0, 1])$  jusqu'à ce que  $ucg(y) < f(y)$ .

**Démonstration:** Nous allons prouver que  $Y_T$  suit la loi de densité  $f$ .

## 5- Méthode d'acceptation-rejet

### Remarques:

- ①  $\frac{f(Y_n)}{cg(Y_n)}$  est bien défini presque partout, car

$$P(Y_n \in \{x : g(x) = 0\}) = 0.$$

- ② On admet que  $T$  est presque sûrement fini, et l'algorithme ne peut donc pas tourner indéfiniment.
- ③  $f$  et  $g$  étant des densités, on a nécessairement  $c \geq 1$ .
- ④ Ce résultat est valable pour une densité  $f$  par rapport à une mesure  $\mu$  quelconque. Pour une variable aléatoire discrète sur un ensemble fini ou dénombrable,  $\mu$  est la mesure de comptage.

## 5- Méthodes d'acceptation-rejet

### Corollaire

La probabilité d'acceptation dans cette méthode est exactement  $\frac{1}{c}$ .  
Autrement dit,  $T$  suit une loi géométrique de paramètre  $\frac{1}{c}$  et le nombre d'essais moyen jusqu'à ce qu'une variable soit acceptée est  $c$ .

**Preuve:**

# 6- Conclusion:

## 6- Conclusion

La plupart des langages comme R ou Python proposent des générateurs aléatoires pour les lois de probabilité usuelles.

Par ailleurs ce chapitre ne fait pas un état de l'art exhaustif des méthodes existantes. Certaines ont été omises comme le rapport d'uniformes et les méthodes de Monte Carlo par chaînes de Markov (MCMC)