

## Simulations de Variables Aléatoires

### Table des matières

<b>1</b>	<b>A partir de l'inverse généralisée de la fonction de répartition</b>	<b>2</b>
1.1	Un peu de théorie . . . . .	2
1.2	Développez votre savoir-faire . . . . .	3
<b>2</b>	<b>Méthode Accept - Reject</b>	<b>3</b>
2.1	Un peu de théorie . . . . .	3
2.2	Développez votre savoir-faire . . . . .	4
<b>3</b>	<b>Méthode de Box-Muller pour des lois normales univariées</b>	<b>5</b>
3.1	Un peu de théorie . . . . .	5
3.2	Développez votre savoir-faire . . . . .	6
<b>4</b>	<b>Générer des échantillons d'une loi normale multivariée</b>	<b>7</b>
4.1	Un peu de théorie . . . . .	7
4.1.1	La loi normale multivariée . . . . .	7
4.1.2	Description de la méthode . . . . .	7
4.2	Développez votre savoir-faire . . . . .	8
<b>5</b>	<b>Echantillonner suivant une loi de Bernouilli</b>	<b>8</b>
5.1	Un peu de théorie . . . . .	8
5.2	Développez votre savoir-faire . . . . .	9
<b>6</b>	<b>Les méthodes Monte Carlo Markov Chain : MCMC</b>	<b>9</b>
6.1	Principe général . . . . .	9
6.2	L'algorithme de Metropolis-Hastings (MH) . . . . .	10
6.3	Algorithme de MH : développez votre savoir-faire . . . . .	13
6.3.1	Influence du support de la loi de proposition . . . . .	13
6.3.2	Echantillonner suivant une loi possédant deux modes . . . . .	13
<b>7</b>	<b>Echantillonnage de Monte Carlo parfait</b>	<b>13</b>
7.1	Problématique . . . . .	13
7.2	Un peu de théorie . . . . .	14
7.3	Développer votre savoir faire . . . . .	15

## Liste des Algorithmes

1	Générateur fondé sur l'inverse généralisée . . . . .	2
2	Générateur Accept-Reject . . . . .	4
3	Algorithme de Box-Muller . . . . .	6
4	Génération d'un échantillon normal multivarié de dimension $n : \mathcal{N}(\mu, \Sigma)$ . .	8
5	Génération d'un échantillon d'une loi de Bernoulli de paramètre $p$ . . . . .	9
6	Principe de l'algorithme de Metropolis-Hastings . . . . .	11
7	Algorithme de Metropolis-Hastings . . . . .	11
8	Algorithme de Metropolis-Hastings : cas d'une loi de proposition symétrique	12
9	Algorithme de Metropolis-Hastings : cas d'une loi de proposition indépendante de $x_t$ . . . . .	12

## 1 A partir de l'inverse généralisée de la fonction de répartition

### 1.1 Un peu de théorie

Soit  $F$  une fonction de répartition définie sur  $\mathbb{R}$ , on définit son inverse généralisée par :

$$F^{-1}(u) = \inf\{x | F(x) \geq u\} \quad (1)$$

**Propriété 1.1.** On appelle  $U$  une variable aléatoire répartie uniformément sur  $[0, 1]$ . La fonction de répartition de la variable aléatoire  $F^{-1}(U)$  est  $F(x)$ .

*Démonstration.* Une fonction de répartition étant croissante :

$$P(F^{-1}(u) \leq x) = P(\inf\{y | F(y) \geq u\} \leq x) = P(u \leq F(x)) = \int_0^{F(x)} dx = F(x) \quad (2)$$

□

Donc si on connaît l'inverse généralisée d'un fonction de répartition  $F$ , on peut générer une variable aléatoire selon la densité  $f(x) = F'(x)$  selon l'algorithme :

---

#### Algorithme 1 : Générateur fondé sur l'inverse généralisée

---

```

1 Require :  $F^{-1}$ 
2 Require : Un générateur de nombre pseudo-aléatoire uniforme entre 0 et 1
    $\mathcal{U}(0, 1)$ 
3 begin
4    $u \sim \mathcal{U}(0, 1)$ 
5    $x \leftarrow F^{-1}(u)$ 
6 end
```

---

## 1.2 Développez votre savoir-faire

**Définition 1.1.** Une variable aléatoire  $X$  suit une loi exponentielle si sa densité de probabilité s'écrit :

$$f(x; \lambda) = \lambda \exp(-\lambda x) \quad (3)$$

pour  $x > 0$  et  $\lambda > 0$

La fonction de répartition de  $X$  est alors égale à :

$$F(x) = 1 - \exp(-\lambda x) \quad (4)$$

1. Calculer l'inverse généralisée de  $F$ .
2. En utilisant la méthode de l'inverse généralisée, simuler 10000 échantillons issus d'une loi exponentielle avec  $\lambda = 2$ . Tracer le résultat.
3. Tracer sur une autre figure la courbe de la densité de probabilité normalisée de  $X$ .
4. Sur cette même figure tracer l'histogramme normalisé des échantillons de la première question. Que constatez-vous ?
5. Quelle est selon vous la principale limitation de cette méthode ?

## 2 Méthode Accept - Reject

### 2.1 Un peu de théorie

On cherche à échantillonner  $X$  suivant la densité  $f(x)$ . On suppose que l'on connaît une densité  $g(x)$  à partir de laquelle on sait échantillonner et qui est telle que :

$$f(x) \leq M g(x) \quad (5)$$

sur le support de  $f$  avec  $M$  une constante. On suppose également qu'il est possible d'échantillonner suivant une loi uniforme  $\mathcal{U}(0, 1)$ . Il est alors possible d'échantillonner suivant  $f$  grâce à l'algorithme suivant :

**Propriété 2.1.** La probabilité que  $x$  soit accepté est égale à  $\frac{1}{M}$ .

*Démonstration.*

$$\begin{aligned} P(x \text{ accepte}) &= \int_x \int_0^{\frac{f(x)}{Mg(x)}} p(u, x) du dx \\ &= \int_x \left( \int_0^{\frac{f(x)}{Mg(x)}} p(u|x) du \right) g(x) dx \end{aligned}$$

Les variables  $X$  et  $U$  étant indépendantes  $p(u|x) = p(u) = \mathcal{U}(0, 1)$  donc :

---

**Algorithme 2 : Générateur Accept-Reject**

---

```
1 Require :  $g$  telle que  $f(x) \leq Mg(x)$ 
2 Require : un générateur de nombre pseudo-aléatoire uniforme entre 0 et 1  $\mathcal{U}(0, 1)$ 
3 begin
4    $u \sim \mathcal{U}(0, 1)$ 
5    $x \sim g(x)$ 
6   while  $(\frac{f(x)}{Mg(x)} < u)$  do
7      $u \sim \mathcal{U}(0, 1)$ 
8      $x \sim g(x)$ 
9   end
10  Accepter  $x$ 
11 end
```

---

$$\begin{aligned} P(x \text{ accepte}) &= \int_x \frac{f(x)}{Mg(x)} g(x) dx \\ &= \frac{1}{M} \end{aligned}$$

□

**Propriété 2.2.** La fonction de répartition de  $X$  est  $F(x)$ .

*Démonstration.* D'après la règle de Bayes

$$\begin{aligned} P(X < x | X \text{ accepte}) &= \frac{P(X < x, X \text{ accepte})}{\frac{1}{M}} \\ &= M \int_{-\infty}^x \left( \int_0^{\frac{f(x)}{Mg(x)}} p(u|x) du \right) g(x) dx \\ &= M \int_{-\infty}^x \frac{f(x)}{Mg(x)} g(x) dx \\ &= F(x) \end{aligned}$$

□

## 2.2 Développez votre savoir-faire

On cherche à échantillonner suivant une loi normale centrée réduite en utilisant la méthode accept-reject. On rappelle que la densité de probabilité de cette loi est

$$f(x) = \frac{1}{\sqrt{2\pi}} \exp\left(\frac{-x^2}{2}\right) \quad (6)$$

On met en œuvre la méthode accept reject à partir de la loi de cauchy de densité de probabilité :

$$g(x) = \frac{1}{\pi (1 + x^2)} \quad (7)$$

La mise en œuvre de cette méthode nécessite de connaître  $M$ . On montre pour cela que le rapport

$$\frac{f(x)}{g(x)} = \sqrt{\frac{\pi}{2}} (1 + x^2) \exp\left(\frac{-x^2}{2}\right) \quad (8)$$

est maximal en  $x = \pm 1$  et atteint approximativement la valeur  $\sqrt{\frac{2\pi}{e}} \simeq 1.52$ .

1. Mettre en œuvre la méthode accept-reject pour échantillonner 10000 échantillons suivant une loi normale centrée réduite. On prendra  $M = 5$ . Tracer le résultat.
2. Tracer sur une autre figure la courbe de la densité de probabilité normalisée de  $X$ .
3. Sur cette même figure tracer l'histogramme normalisé des échantillons de la première question. Que constatez-vous ?
4. A partir des échantillons précédents, calculer 10000 échantillons distribués suivant une loi normale de moyenne 5. Vérifier qualitativement le résultat à l'aide d'un histogramme.

### 3 Méthode de Box-Muller pour des lois normales univariées

#### 3.1 Un peu de théorie

C'est certainement la méthode la plus simple à implémenter même si ce n'est pas forcément la plus rapide. Cet algorithme génère un échantillon à partir de la loi normale bivariée réduite, c'est-à-dire la loi normale de moyenne nulle et de matrice de covariance la matrice identité en dimension deux  $I_2$ . On appelle  $Z = [Z_1, Z_2]^T$  le vecteur aléatoire qui suit une loi normale bivariée réduite. La densité de probabilité de  $Z$  est :

$$f(z) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{z^T z}{2}\right) \quad (9)$$

avec  $z = [z_1, z_2]^T$ . On le note  $Z \sim \mathcal{N}_2(0, I_2)$ . Chaque composante  $Z_i$  suit ainsi une loi normale réduite univariée. L'algorithme de Box Muller repose sur deux propriétés de la loi normale bivariée :

- $R = Z_1^2 + Z_2^2$  suit une loi exponentielle de moyenne 2 :

$$P(R \leq x) = 1 - \exp\left(-\frac{x}{2}\right) \quad (10)$$

- Pour  $R$  donné, le point  $(Z_1, Z_2)$  est distribué uniformément sur le cercle centré à l'origine et de rayon  $\sqrt{R}$ .

Pour générer  $Z_1$  et  $Z_2$  on commence donc par générer  $R$  et ensuite on choisit de manière uniforme un point sur le cercle de rayon  $\sqrt{R}$ . Pour générer  $R$ , on utilise la méthode de l'inverse de la fonction de répartition.  $U$  étant distribué selon une loi uniforme  $\mathcal{U}(0, 1)$ , alors  $1 - U$  est aussi distribué selon  $\mathcal{U}(0, 1)$ . On calcule donc  $R = -2 \ln(U_1)$  avec  $U_1 \sim \mathcal{U}(0, 1)$ . Pour obtenir un point positionné aléatoirement sur le cercle de rayon  $R$  on génère un angle  $V$  de façon uniforme sur  $[0, 2\pi]$ . En utilisant de nouveau la méthode de l'inverse de la fonction de répartition, on calcule en fait  $V = 2\pi U_2$  avec  $U_2 \sim \mathcal{U}(0, 1)$ .  $Z_1$  et  $Z_2$  sont alors obtenus par :

$$Z_1 = \sqrt{R} \cos(V) \quad (11)$$

$$Z_2 = \sqrt{R} \sin(V) \quad (12)$$

Au total, l'algorithme à mettre en œuvre est le suivant :

---

**Algorithme 3 :** Algorithme de Box-Muller

---

```

1 Require : un générateur de nombre pseudo-aléatoire uniforme entre 0 et 1  $\mathcal{U}(0, 1)$ 
2 begin
3    $u1 \sim \mathcal{U}(0, 1)$ 
4    $u2 \sim \mathcal{U}(0, 1)$ 
5    $R = -2 \ln(u1)$ 
6    $V = 2\pi u2$ 
7    $z_1 = \sqrt{R} \cos(V)$ 
8    $z_2 = \sqrt{R} \sin(V)$ 
9 end
```

---

### 3.2 Développez votre savoir-faire

1. Mettre en œuvre la méthode de Box-Muller pour échantillonner 10000 échantillons suivant une loi normale centrée réduite. Tracer le résultat.
2. Tracer sur une autre figure la courbe de la densité de probabilité normalisée de  $X$ .
3. Sur cette même figure, tracer l'histogramme normalisé des échantillons de la première question. Que constatez-vous ?
4. Relancer plusieurs fois les programmes des questions précédentes. Que remarquez-vous sur les deuxièmes figures quant à la coïncidence des courbes ?
5. A partir des échantillons précédents, calculer 5000 échantillons distribués suivant une loi normale de moyenne -3 et de variance 3. Vérifier qualitativement le résultat à l'aide d'un histogramme.

## 4 Générer des échantillons d'une loi normale multivariée

### 4.1 Un peu de théorie

#### 4.1.1 La loi normale multivariée

Une loi normale multivariée est caractérisée par un vecteur moyenne  $\mu$  et par une matrice de covariance  $\Sigma$ . Si  $X$  suit une loi normale multivariée alors sa densité de probabilité  $p(x)$  ( $x \in \mathbb{R}^n$ ) est égale :

$$p(x) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} \exp \left( -\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \right) \quad (13)$$

Lorsque  $\Sigma = I_n$  avec  $I_n$  la matrice identité de dimension  $n$ , alors la  $i$ ème composante  $X_i$  suit une loi normale  $\mathcal{N}(\mu_i, \sigma_i)$  avec :

$$\Sigma_{i,j} = E[(X_i - \mu_i)(X_j - \mu_j)] \quad (14)$$

et

$$\sigma_i = \Sigma_{ii} \quad (15)$$

On appelle coefficient de corrélation la quantité  $\rho_{ij}$  avec :

$$\rho_{ij} = \frac{\Sigma_{ij}}{\sigma_i \sigma_j} \quad (16)$$

#### 4.1.2 Description de la méthode

Une loi normale multivariée est caractérisée par un vecteur moyenne  $\mu$  et par une matrice de covariance  $\Sigma$ . La méthode de génération d'un vecteur distribué selon une loi normale  $\mathcal{N}(\mu, \Sigma)$  repose sur la propriété de transformation linéaire intuitée dans les manipulations précédentes pour une loi univariée.

**Propriété 4.1.** *Toute transformation linéaire d'un vecteur normal est encore un vecteur normal :*

$$X \sim \mathcal{N}(\mu, \Sigma) \quad \Rightarrow \quad AX \sim \mathcal{N}(A\mu, A\Sigma A) \quad (17)$$

pour toute matrice  $A$ .

Par conséquent, si on génère un vecteur  $Z \sim \mathcal{N}(0, I_n)$  le vecteur  $X = AZ + \mu$  suit une loi normale  $\mathcal{N}(\mu, AA^T)$ . Le vecteur  $Z$  peut être généré en utilisant l'une des méthodes précédentes comme par exemple l'algorithme de Box-Muller. Supposons alors que l'on trouve une matrice  $A$  telle que  $\Sigma = AA^T$ , alors  $X = AZ + \mu$  suit une loi  $\mathcal{N}(\mu, \Sigma)$ . Le problème consiste donc à déterminer cette matrice  $A$ . Parmi toutes les matrices possibles le choix d'une matrice triangulaire inférieure est particulièrement intéressant car il réduit la complexité des calculs. On utilise alors la décomposition de Choleski qui permet justement de déterminer pour toute matrice  $\Sigma$  définie positive une matrice triangulaire inférieure telle que  $\Sigma = AA^T$ . L'algorithme final est donc l'algorithme 4 :

---

**Algorithme 4 :** Génération d'un échantillon normal multivarié de dimension  $n$  :  $\mathcal{N}(\mu, \Sigma)$

---

```

1 Require : un générateur de nombre pseudo-aléatoire suivant une loi normale
   centrée réduite  $\mathcal{N}(0, 1)$ 
2 begin
3   Générer  $n$  échantillons gaussiens  $\mathcal{N}(0, 1) : Z_i$ 
4   Former  $Z = [Z_1, \dots, Z_n]$ 
5   Calculer  $A = \text{Choleski}(\Sigma)$ 
6   Calculer  $X = \mu + AZ$ 
7 end

```

---

## 4.2 Développez votre savoir-faire

1. Mettre en œuvre la méthode précédente pour générer 1000 échantillons d'une loi normale multivariée pour :

$$\mu = [0 \quad 50 \quad 100 \quad -50 \quad -100 \quad 200]^T \quad (18)$$

$$\Sigma = \begin{pmatrix} 11 & 10 & 5 & 9 & 4 & 2 \\ 10 & 13 & 9 & 15 & 5 & 3 \\ 5 & 9 & 15 & 11 & 3 & 1 \\ 9 & 15 & 11 & 21 & 6 & 4 \\ 4 & 5 & 3 & 6 & 5 & 1 \\ 2 & 3 & 1 & 4 & 1 & 1 \end{pmatrix} \quad (19)$$

Vous utiliserez pour cela la fonction  $\text{chol}(M)$  qui retourne une matrice  $C$  telle que  $M = C^T C$ .

2. Tracer sur une même figure l'évolution de chacune des composantes.
3. Comment expliquez-vous les différences entre les amplitudes des variations de ces composantes ?

## 5 Échantillonner suivant une loi de Bernoulli

### 5.1 Un peu de théorie

On cherche à échantillonner suivant une loi de Bernoulli de paramètre  $p$ . Soit  $X \in \{0, 1\}$  avec

$$\begin{cases} X = 0 & \text{avec la prob } p \\ X = 1 & \text{avec la prob } 1 - p \end{cases} \quad (20)$$

Soit  $U$  une variable uniforme  $\mathcal{U}(0, 1)$  alors :



$$\begin{cases} \text{Prob}(U < p) = p \\ \text{Prob}(U > p) = 1 - p \end{cases} \quad (21)$$

On en déduit l'algorithme suivant :

---

**Algorithme 5 :** Génération d'un échantillon d'une loi de Bernouilli de paramètre  $p$

---

```

1 Require : un générateur de nombre pseudo-aléatoire uniforme entre 0 et 1  $\mathcal{U}(0, 1)$ 
2 begin
3   Générer un échantillon  $u \sim \mathcal{U}(0, 1)$ 
4   if  $u < p$  then
5      $X = 0$ 
6   else
7      $X = 1$ 
8   end
9 end

```

---

## 5.2 Développez votre savoir-faire

1. Générer 1000 échantillons suivant une loi de Bernouilli de paramètre  $p = 0.7$ . Tracer le résultat.
2. Calculer la fréquence du nombre de zéros générés. Que remarquez-vous ?

## 6 Les méthodes Monte Carlo Markov Chain : MCMC

### 6.1 Principe général

Le principe des méthodes Monte Carlo Markov Chain (MCMC) est de construire une chaîne de Markov ergodique  $\{x_t\}$  dont la loi marginale converge vers  $f(x)$  au fur et à mesure que  $t$  augmente.

idée de base dans le cas discret. l'idée de base de l'algorithme de Metropolis-Hastings est que à chaque étape on propose de passer de l'état courant  $x$  à un nouvel état  $x'$  avec la probabilité  $q(x'|x)$  avec  $q$  appelée la distribution instrumentale (proposal distribution) aussi appelée le noyau.

Si on utilise une distribution instrumentale  $q(x'|x) = q(x')$  pour laquelle le nouvel état est indépendant de l'ancien état, nous obtenons un échantillonneur qui est similaire à l'échantillonnage d'importance.

Après avoir proposé le passage de l'état  $x$  à  $x'$  il faut alors décider d'accepter cette proposition ou non selon une expression qui garanti que la fraction de temps passé dans chaque état est proportionnelle à  $f(x)$ , la densité cible. Si la proposition est acceptée, le nouvel état est  $x'$ , sinon le nouvel état est le même que l'état courant  $x$ .

Si la densité instrumentale est symétrique, donc  $q(x'|x) = q(x|x')$ , la probabilité d'acceptation est donnée par :

$$r = \min \left( 1, \frac{f(x')}{f(x)} \right)$$

On peut constater que si  $x'$  est plus probable que  $x$ , on s'y déplace définitivement (puisque  $\frac{f(x')}{f(x)} > 1$ ), mais si  $x'$  est moins probable, on peut quand même s'y déplacer selon les probabilités relatives. Ainsi, au lieu de passer systématiquement aux états les plus probables, nous autorisons parfois des mouvements vers des états moins probables. On peut montrer que cette procédure garantit que la fraction du temps que nous passons dans chaque état  $x$  est proportionnel à  $f(x)$ .

Si la proposition n'est pas symétrique, donc  $q(x'|x) \neq q(x|x')$ , nous avons besoin de la correction de Hastings :

$$r = \min(1, \alpha)$$

$$\alpha = \frac{f(x')q(x|x')}{f(x)q(x'|x)} = \frac{f(x')q(x'|x)}{f(x)q(x|x')}$$

Cette correction est nécessaire pour compenser le fait que la distribution instrumentale elle-même (plutôt que simplement la distribution cible) pourrait favoriser certains États. Une raison importante pour laquelle l'algorithme de Metropolis-Hastings est un algorithme pratique est que, lors de l'évaluation de  $\alpha$ , nous avons seulement besoin de connaître la densité cible à une constante de normalisation près. En effet, supposons que  $f(x) = \frac{1}{Z}\tilde{f}(x)$  avec  $\tilde{f}(x)$  une distribution non normalisée et  $Z$  une constante de normalisation. Il vient :

$$\alpha = \frac{\frac{\tilde{f}(x')}{Z}q(x|x')}{\frac{\tilde{f}(x)}{Z}q(x'|x)}$$

On peut remarquer que les  $Z$  se simplifient et donc on peut échantillonner suivant  $f(x)$  même si  $Z$  est inconnue. Le principe de l'algorithme de Metropolis-Hastings est décrit par l'algorithme 6.

## 6.2 L'algorithme de Metropolis-Hastings (MH)

D'une manière générale les méthodes MCMC définissent au préalable une loi de proposition, aussi appelée loi instrumentale,  $q(y|x)$  à partir de laquelle on sait échantillonner. L'algorithme de Metropolis-Hastings (algorithme 7) définit alors les conditions pour que la loi marginale de  $x_t$  converge vers  $f(x)$  en fonction de  $t$ .

**Remarque 6.1.** *Compte-tenu de la définition de la fonction  $\rho$ , la loi de proposition ne doit pas être nulle sur le support de  $f$ .*

**Remarque 6.2.** *La loi cible peut être connue à une constante multiplicative près*

---

**Algorithme 6** : Principe de l'algorithme de Metropolis-Hastings

---

```
1 begin
2   Choisir une valeur initiale  $x^0$ 
3   Initialiser :  $s = 0$ 
4   while (la convergence n'est pas assurée) do
5     Définir :  $x = x^s$ 
6     Echantillonner :  $x' \sim q(x'|x)$ 
7     Calculer la probabilité d'acceptation :  $\alpha = \frac{f(x')q(x|x')}{f(x)q(x'|x)}$ 
8     Calculer :  $r = \min(1, \alpha)$ 
9     Echantillonner :  $u \sim \mathcal{U}(0, 1)$ 
10    Mettre à jour le nouvel échantillon :  $x^{s+1} = \begin{cases} x' & \text{si } u < r \\ x^s & \text{si } u \geq r \end{cases}$ 
11    Passer à l'échantillon suivant :  $s = s + 1$ 
12  end
13 end
```

---

---

**Algorithme 7** : Algorithme de Metropolis-Hastings

---

```
1 begin
2   Choisir une valeur initiale  $x_0$ 
3   Générer  $y_t \sim q(y_t|x_t)$ 
4   while (la convergence n'est pas assurée) do
5      $x_{t+1} = \begin{cases} y_t & \text{avec probabilité } \rho(x_t, y_t) \\ x_t & \text{avec probabilité } 1 - \rho(x_t, y_t) \end{cases}$ 
6   end
7 end
8 Ensure :  $x_t \sim f(x)$ 
```

---

**Remarque 6.3.** *La chaîne peut prendre plusieurs fois la même valeur ; les échantillons successifs ne sont pas iid.*

Lorsque la loi de proposition est symétrique  $q(y|x) = q(x|y)$  l'algorithme 7 se simplifie en l'algorithme 8.

---

**Algorithme 8 :** Algorithme de Metropolis-Hastings : cas d'une loi de proposition symétrique

---

```

1 begin
2   Choisir une valeur initiale  $x_0$ 
3   Générer  $y_t \sim q(y_t|x_t)$ 
4   while (la convergence n'est pas assurée) do
5      $x_{t+1} = \begin{cases} y_t & \text{avec probabilité } \rho(x_t, y_t) \\ x_t & \text{avec probabilité } 1 - \rho(x_t, y_t) \end{cases} \quad \rho(x, y) = \min \left\{ \left( \frac{f(y)}{f(x)} \right), 1 \right\}$ 
6   end
7 end
8 Ensure :  $x_t \sim f(x)$ 

```

---

On peut également choisir une loi de proposition indépendante de  $x_t$  :

$$q(y|x_t) = q(y) \tag{22}$$

L'algorithme devient alors (Algorithme 9) :

---

**Algorithme 9 :** Algorithme de Metropolis-Hastings : cas d'une loi de proposition indépendante de  $x_t$

---

```

1 begin
2   Choisir une valeur initiale  $x_0$ 
3   Générer  $y_t \sim q(y_t|x_t)$ 
4   while (la convergence n'est pas assurée) do
5      $x_{t+1} = \begin{cases} y_t & \text{avec probabilité } \rho(x_t, y_t) \\ x_t & \text{avec probabilité } 1 - \rho(x_t, y_t) \end{cases} \quad \rho(x, y) = \min \left\{ \left( \frac{f(y)q(x)}{f(x)q(y)} \right), 1 \right\}$ 
6   end
7 end
8 Ensure :  $x_t \sim f(x)$ 

```

---

Dans tous ces algorithmes se pose la question suivante : comment traduire le fait que la convergence soit assurée ? Ces méthodes reposent sur la construction d'une suite  $x_t$  dont la loi converge vers celle recherchée au bout d'un certain temps. En pratique, cela a pour conséquence de ne considérer les échantillons valides que pour  $t \geq t_0$  avec  $t_0$  fixé. Cette période durant laquelle les échantillons ne sont pas considérés comme valides est appelée *burn-in*.

## 6.3 Algorithme de MH : développez votre savoir-faire

### 6.3.1 Influence du support de la loi de proposition

Nous allons commencer par utiliser l'algorithme de Metropolis-Hastings avec loi de proposition indépendante pour échantillonner à partir d'une loi normale de moyenne nulle et de variance unité. La loi de proposition choisie est une loi uniforme  $\mathcal{U}(a, b)$ .

1. On choisit une loi de proposition  $\mathcal{U}(-1, 1)$ . Calculer 20000 échantillons d'une loi  $\mathcal{N}(0, 1)$ . On utilisera un nombre d'échantillons de burn-in de 500.
2. Tracer sur une autre figure la courbe de la densité de probabilité normalisée de  $X$ .
3. Sur cette même figure, tracer l'histogramme normalisé des échantillons de la première question. Que constatez-vous ?
4. Reprendre les questions précédentes avec une loi de proposition  $\mathcal{U}(-20, 20)$  et un burn-in de 1000. Analyser les résultats.

### 6.3.2 Echantillonner suivant une loi possédant deux modes

Nous allons maintenant chercher à échantillonner suivant une loi plus complexe comportant deux modes :

$$f(x) = p \frac{1}{2a_1} \exp\left(-\left|\frac{x - \mu_1}{a_1}\right|\right) + \frac{1}{2a_2} \exp\left(-\left|\frac{x - \mu_2}{a_2}\right|\right)(1 - p) \quad (23)$$

$X$  est donc une variable aléatoire de Laplace de paramètre  $(\mu_1, a_1)$  avec la probabilité  $p$  ou une variable aléatoire de Laplace de paramètre  $(\mu_2, a_2)$  avec la probabilité  $1 - p$ .

1. Tracer la densité de probabilité cible sur le support  $[-20, 20]$  pour  $a_1 = 1$ ,  $a_2 = 2$ ,  $\mu_1 = 10$ ,  $\mu_2 = -5$  et  $p = 0.3$ .
2. Mettre en œuvre un algorithme de Metropolis Hastings pour générer 20000 échantillons suivant  $f$  en utilisant une loi normale convenablement choisie comme loi candidate. Tracer l'histogramme des résultats sur la figure précédente pour valider votre résultat. (Vous pouvez utiliser la fonction `randn()` pour générer les échantillons gaussiens).

## 7 Echantillonnage de Monte Carlo parfait

### 7.1 Problématique

Dans un raisonnement probabiliste on est généralement amené à calculer une intégrale de la forme :

$$I(f) = \mathbb{E}_{p(x)}[f(x)] = \int f(x)p(x)dx \quad (24)$$

C'est la cas, par exemple, lorsque l'on veut estimer le prix initial d'une option en finance. Pour un certain nombre de densités classiques, il est possible de la calculer analytiquement. Comment fait-on dans les autres cas ? La solution qui vient immédiatement

est de conduire le calcul de manière numérique sur un ordinateur. La question devient alors : comment faire ? Il est possible, dans certains cas, de mettre en œuvre des méthodes d'intégrations numériques déterministes fondées sur une discrétisation de l'espace. Elles sont cependant d'une complexité importante en grande dimension. Les méthodes de Monte Carlo sont un ensemble de méthodes fondées elles aussi sur des simulations numériques permettant de calculer la distribution de probabilité. Mais, contrairement aux méthodes déterministes qui utilisent une grille fixe de l'espace des paramètres, les méthodes de Monte Carlo séquentielles se fondent sur une grille *adaptée* à chaque forme de densité de probabilité.

## 7.2 Un peu de théorie

On suppose dans cette section que l'on est capable de simuler  $N$  échantillons indépendants et identiquement distribués (i.i.d) à partir de la densité  $p(x)$ . Chaque échantillon est appelé une *particule* et est noté  $\tilde{x}^{(i)}$ . La distribution  $p(x)$  peut alors être approchée par la quantité :

$$P_N(dx) = \frac{1}{N} \sum_{i=1}^N \delta_{\tilde{x}^{(i)}}(dx) \quad (25)$$

avec  $\delta_{\tilde{x}^{(i)}}(dx)$  la distribution de dirac en  $\tilde{x}^{(i)}$ . L'intégrale

$$I(f) = \mathbb{E}_{p(x)}[f(x)] = \int f(x)p(x)dx \quad (26)$$

avec  $f$  une fonction de  $x$ , peut donc elle aussi être approchée par :

$$I_N(f) = \int f(x)P_N(dx) = \frac{1}{N} \sum_{i=1}^N f(\tilde{x}^{(i)}) \quad (27)$$

$I_N(f)$  est ce que l'on appelle un *estimateur* de  $I(f)$ . On montre que cet estimateur de l'intégrale est sans biais, c'est-à-dire que l'espérance de  $I_N(f)$  est égale à la véritable valeur  $I(f)$ . Si, de plus, la variance a posteriori de  $f(x)$  est finie :

$$\mathbb{E}_{p(x)}[f(x)] - I(f)^2 = \sigma_f^2 < +\infty \quad (28)$$

alors la variance de l'estimateur est égale à :

$$\text{var}(I_N(f)) = \frac{\sigma_f^2}{N} \quad (29)$$

La loi forte des grands nombres, nous permet alors d'affirmer que  $I_N(f)$  converge presque sûrement (avec probabilité un) vers  $I(f)$  lorsque  $N$  tend vers l'infini :

$$P(\lim_{N \rightarrow \infty} I_N(f) = I(f)) = 1 \quad (30)$$

En appliquant le théorème de la limite centrale, on montre que  $\sqrt{N}[I_N(f) - I(f)]$  converge en loi vers  $\mathcal{N}(0, \sigma_f^2)$  lorsque  $N$  tend vers l'infini.

**Remarque 7.1.** *Contrairement aux méthodes d'approximation déterministes, le taux de convergence est indépendant de la dimension de la fonction à intégrer.*

Le problème consiste donc à obtenir les particules à partir d'une densité donnée. Il est possible d'utiliser une des méthodes décrite précédemment.

### 7.3 Développer votre savoir faire

1. Utiliser l'échantillonnage de Monte Carlo pour déterminer la moyenne d'une loi normale  $\mathcal{N}(5, 2)$ . On utilisera 100 particules.
2. Recommencer 10000 fois la question précédente et enregistrer chaque résultat dans un vecteur  $V_M$ . Tracer le résultat. Que remarquez-vous ?
3. Recommencer les questions précédentes avec 1000 puis 10000 particules. Que remarquez-vous ? Comment expliquez-vous ce phénomène ?