

# Compte rendu stage Simulation parfaite

Antonin LAURENT

8 juillet 2019

## Table des matières

<b>1</b>	<b>Chapitre 1 : Introduction</b>	<b>3</b>
1.1	Définitions et théorèmes . . . . .	3
1.2	Modèles étudiés . . . . .	6
1.2.1	Champs aléatoires de Markov . . . . .	6
1.2.2	Permutations . . . . .	8
1.3	Chaînes de Markov et simulation approchée . . . . .	9
1.4	Création de chaînes de Markov . . . . .	10
1.4.1	Échantillonnage de Gibbs . . . . .	11
1.4.2	Metropolis-Hastings . . . . .	17
1.4.3	Variables aléatoires auxiliaires / Auxiliary random variables . . . . .	22
1.4.4	Shift Chains . . . . .	23
<b>2</b>	<b>Chapitre 2 : Procédures de rejet</b>	<b>24</b>
2.1	Théorème et exemple simple . . . . .	24
2.2	Méthode de rejet pour variables aléatoires à densité . . . . .	25
2.2.1	Exemple : D'une Cauchy vers une Normale . . . . .	27
2.3	Union d'ensembles . . . . .	30
2.4	Simulation à l'aide des inégalités de Markov et Chernoff . . . . .	31
2.4.1	L'inégalité de Markov en tant que procédure de rejet . . . . .	31
2.4.2	Les inégalités de Chernoff en tant que procédure de rejet . . . . .	32
2.5	Défaut des méthodes de rejet . . . . .	38
<b>3</b>	<b>Chapitre 3 : Coupling From The Past</b>	<b>41</b>
3.1	CFTP : approche trajectorielle (Baccelli / Brémaud) . . . . .	41
3.2	CFTP : approche par blocs (Huber) . . . . .	48
3.2.1	Variation de la taille des blocs . . . . .	51

# 1 Chapitre 1 : Introduction

## 1.1 Définitions et théorèmes

### Définition 1. Temps d'arrêt

On dit que  $T$  est un temps d'arrêt pour une suite  $X_1, X_2, \dots$  si, connaissant les valeurs de  $X_1, X_2, \dots, X_n$  on peut déterminer si  $T \leq n$ .

### Définition 2. Fonction calculable

Une fonction est dite calculable (computable en anglais) si il existe un algorithme capable de retourner le résultat de la fonction.

### Définition 3. Algorithme probabiliste

Soient  $\mathcal{I}$  un ensemble d'indices tel que pour tout  $I \in \mathcal{I}$ , il existe une distribution  $\pi_I$  sur un espace d'états  $\Omega_I$ . On se donne une suite de variables aléatoires  $X_1, X_2, \dots$  où les  $X_I \in S_I$ .

Un algorithme probabiliste est une famille de temps d'arrêts  $\{T_I\}_{I \in \mathcal{I}}$  et de fonctions calculables  $\{f_{I,t}\}_{I \in \mathcal{I}, t \in \{1,2,\dots\}}$ . La sortie de l'algorithme est  $f_{I,T_I}(X_1, \dots, X_{T_I})$ .

### Définition 4. Algorithme de simulation parfaite

Un algorithme de simulation parfaite est un algorithme probabiliste dont la sortie est une variable aléatoire qui provient d'une distribution cible. Les algorithmes de simulation parfaite sont une sous-classe des algorithmes de simulation exacte, algorithmes qui tirent d'une distribution ciblée. Cependant les algorithmes dont le temps d'arrêt  $T_I$  est déterministe (algorithmes tournant selon un nombre fini de choix aléatoires) sont généralement considérés comme algorithmes de simulation exacte mais pas comme algorithmes de simulation parfaite.

**Théorème 1** (Théorème fondamental de la simulation parfaite). *On suppose que pour  $U_1, U_2, \dots$  iid tel que  $U_i \sim \text{Unif}([0,1])$ , il existe des fonctions calculables  $b, g$  et  $f$  telles que la fonction  $b$  aie pour image  $\{0,1\}$  et  $\mathbb{P}(b(U) = 1) > 0$ .*

*Pour une variable aléatoire  $X$  qui vérifie :*

$$X \sim b(U)g(U) + (1 - b(U))f(X, U), \quad (1)$$

*soit  $T = \inf\{t : b(U_t) = 1\}$ . On a alors que :*

$$Y = f(\dots f(f(g(U_T), U_{T-1}), U_{T-2}), \dots, U_1)$$

*a la même distribution que  $X$  et on a  $\mathbb{E}[T] = \frac{1}{\mathbb{P}(b(U)=1)}$ .*

*Démonstration.* Soient  $X_0, X_1, \dots$  des tirages indépendants chacun distribués selon  $X$ , et  $U_1, U_2, \dots$  iid tel que  $U_i \sim \text{Unif}([0,1])$ . Pour  $X_t$ , fixons  $X_{t,t} = X_t$  et récursivement, on pose :

$$X_{t,i} = b(U_{i+1})g(U_{i+1}) + (1 - b(U_{i+1}))f(X_{t,i+1}, U_{i+1}),$$

pour  $i \in \{0, \dots, t-1\}$ .

On a alors d'après la relation (1) :  $X_{t,0} \sim X$ . On montre ce résultat pour les premiers indices  $t$ , les suivants sont prouvés de la même manière.

**t=0**

$$X_{0,0} = X_0$$

comme on l'a posé précédemment, d'où

$$X_{0,0} \sim X.$$

**t=1**

$$X_{1,0} = b(U_1)g(U_1) + (1 - b(U_1))f(X_{1,1}, U_1)$$

Or,

$$X_{1,1} = X_1 \sim X.$$

En remplaçant dans l'équation précédente, on obtient :

$$X_{1,0} = b(U_1)g(U_1) + (1 - b(U_1))f(X_1, U_1)$$

D'après la relation (1), on obtient alors  $X_{1,0} = X_1 \sim X$ .

**t=2**

$$\begin{aligned} X_{2,0} &= b(U_1)g(U_1) + (1 - b(U_1))f(X_{2,1}, U_1) \\ X_{2,1} &= b(U_2)g(U_2) + (1 - b(U_2))f(X_{2,2}, U_2) \\ &= b(U_2)g(U_2) + (1 - b(U_2))f(X_2, U_2) \end{aligned}$$

À nouveau, par la relation (1), on a :  $X_{2,1} = X_2$  On remplace donc par cette valeur dans  $X_{2,0}$  et on obtient :

$$X_{2,0} = b(U_1)g(U_1) + (1 - b(U_1))f(X_2, U_1)$$

D'après la relation (1), on obtient alors  $X_{2,0} = X_2 \sim X$ .

On a donc  $X_{0,0}, X_{1,0}, X_{2,0}, \dots$  de même distribution que  $X$  mais pas forcément indépendants. On considère à présent la variable  $Y$  énoncée dans le théorème. On a alors la relation suivante :  $X_{t,0} = Y$  si  $t \geq T$ . On illustre ce résultat avec comme exemple  $T = 2$  et  $t = 2, 3$  :

On a donc :

$$Y = f(g(U_2), U_1)$$

Montrons que  $X_{2,0} = X_{3,0} = Y$ . Étant donné que  $T = 2$ , on a  $b(U_1) = 0$  et  $b(U_2) = 1$ , d'où :

$$\begin{aligned} X_{2,0} &= f(X_{2,1}, U_1) \\ X_{2,1} &= g(U_2) \end{aligned}$$

on remplace dans la première équation et on obtient

$$X_{2,0} = f(g(U_2), U_1)$$

D'où le résultat pour  $t=2$ . Voyons pour  $t=3$ .

De même, puisque  $T = 2$ , on a  $b(U_1) = 0$  et  $b(U_2) = 1$ , d'où :

$$\begin{aligned} X_{3,0} &= f(X_{3,1}, U_1) \\ X_{3,1} &= g(U_2) \end{aligned}$$

on remplace dans la première équation et on obtient

$$X_{3,0} = f(g(U_2), U_1)$$

D'où le résultat.

Ensuite, puisque les  $U_i$  sont indépendants, on a que :  $\mathbb{P}(T > t) = (1 - \mathbb{P}(b(U) = 1))^t$  et puisque, par hypothèse,  $\mathbb{P}(b(U) = 1) > 0$ , on a la relation

qui tend vers 0 pour  $t$  tendant vers l'infini. Il ne reste plus qu'à montrer  $Y \sim X$ . Pour tout  $t$ , pour tout ensemble  $C$  :

$$\mathbb{P}(Y \in C) = \mathbb{P}(Y \in C, t \geq T) + \mathbb{P}(Y \in C, t < T)$$

puisque  $X_{t,0} = Y$  si  $t \geq T$ , on a

$$\begin{aligned} &= \mathbb{P}(X_{t,0} \in C, t \geq T) + \mathbb{P}(Y \in C, t < T) \\ &= \mathbb{P}(X_{t,0} \in C) - \mathbb{P}(X_{t,0} \in C, t < T) + \mathbb{P}(Y \in C, t < T) \end{aligned}$$

puisque  $X_{t,0} \sim X$ , on a

$$= \mathbb{P}(X \in C) - \mathbb{P}(X_{t,0} \in C, t < T) + \mathbb{P}(Y \in C, t < T)$$

Les deux derniers termes sont bornés par  $\mathbb{P}(T > t) = (1 - \mathbb{P}(b(U) = 1))^t$ , et puisque l'équation est vraie pour n'importe quel  $t$ , on obtient :  $\mathbb{P}(Y \in C) = \mathbb{P}(X \in C)$  pour tout ensemble  $C$ , donc  $Y \sim X$ . Le fait que  $\mathbb{E}[T] = \frac{1}{\mathbb{P}(b(U)=1)}$  provient du fait que  $T$  suit une loi géométrique de paramètre  $\mathbb{P}(b(U) = 1)$

□

**Définition 5.** *Algorithme arretable*

Dans le théorème fondamental de la simulation parfaite, si  $X$  et  $T$  sont indépendants, on dit que l'algorithme est arretable, sinon il est non-arretable.

**Définition 6.** *Algorithme de simulation parfaite à lecture unique*

Un algorithme de simulation parfaite qui utilise  $X \sim b(U)g(U) + (1 - b(U))f(X, U)$  est un algorithme à lecture unique si  $f(X, u) = f(X, u')$  pour tout  $u, u'$ . Sinon, c'est un algorithme à lecture double.

En général, un algorithme arretable est préférable à un algorithme non-arretable, et un algorithme à lecture unique est préférable à un algorithme à lecture double.

## 1.2 Modèles étudiés

### 1.2.1 Champs aléatoires de Markov

On considère un graphe  $G = (V, E)$ .  $V$  est l'ensemble des nœuds et  $E$  est l'ensemble des arêtes. On notera par la suite  $\Delta$  le degré du graphe (égal au degré du nœud ayant un nombre de voisins maximal). On notera par  $\Omega$  l'ensemble des labels des nœuds.

**Définition 7.** *Ensemble séparant*

On dit qu'un sous-ensemble de nœuds  $S$  sépare les nœuds  $i$  et  $j$  si tout chemin du graphe menant  $i$  à  $j$  passe par  $S$ .

**Définition 8.** *Champ aléatoire de Markov*

Pour un graphe  $G = (V, E)$  et un ensemble de labels  $\Omega$ , on dit que la distribution de  $X$  sur  $\Omega$  est un champ aléatoire de Markov si, pour tous les nœuds  $i$  et  $j$ , pour tous les ensembles  $S$  séparant  $i$  de  $j$ , on a :  $[X(i)|X(j), X(S)] \sim [X(i)|X(S)]$ . Un état  $x \in \Omega$  est appelé une configuration.

**Définition 9.** *Clique*

Une clique est un sous-ensemble de nœuds du graphe tel que chaque paire de nœuds soit connectée par une arête.

**Théorème 2** (Théorème de Hammersley-Clifford). *Pour un graphe fini  $G = (V, E)$ , la distribution  $\pi$  est un champ aléatoire de Markov si elle a pour densité  $f_X$  et qu'il existe des fonctions  $\phi_C$  pour toute clique  $C$  telles que  $f_X$  puisse s'écrire :*

$$f_X(x) = \frac{1}{Z} \prod_{C \in \text{cliques}(G)} \phi_C(x)$$

**Définition 10.** *Auto-modèle*

On dit qu'un champ aléatoire de Markov est un auto-modèle, si il existe des fonctions  $f_i$  et  $g_i$  telles que la densité de  $X \sim \pi$  peut-être écrite sous la forme :

$$f_X(x) = \frac{1}{Z} \left[ \prod_{i \in V} f_i(X(i)) \right] \left[ \prod_{\{i,j\} \in E} g_{\{i,j\}}(X(i), X(j)) \right]$$

Dans la suite, on définit des exemples bien connus d'auto-modèles que l'on étudiera par la suite.

**Exemple 1 : Modèle d'Ising**

Le modèle d'Ising est un auto-modèle de paramètres  $\mu$  et  $\beta$ , où  $\Omega = \{-1, 1\}^V$ , et  $f(c) = \exp(\mu c)$ ,  $g(c_1, c_2) = \exp(\beta \mathbf{1}(c_1 = c_2))$ . Dans la littérature, on appelle  $\mu$  magnétisation et  $\beta$  la température inverse. Lorsque  $\beta > 0$ , on dit que le modèle est ferromagnétique. Lorsque  $\mu = 0$ , on dit que le modèle est sans champ magnétique.

### Exemple 2 : Modèle Hard-core gas

Le modèle hard-core gas est un auto-modèle défini sur  $\{0, 1\}^V$  et de paramètre  $\lambda > 0$  où  $f(c) = \lambda^c$  et  $g(c_1, c_2) = 1 - c_1 c_2$ . Lorsqu'un nœud  $\nu$  a pour label 1, on dit que le nœud est occupé, sinon, il est inoccupé.

### Exemple 3 : Modèle de Strauss

Le modèle de Strauss est un auto-modèle sur  $\{0, 1\}^V$  de paramètres  $\lambda > 0$  et  $\gamma \in [0, 1]$  où  $f(c) = \lambda^c$  et  $g(c_1, c_2) = 1 + (\gamma - 1)c_1 c_2$ . Lorsqu'un nœud  $\nu$  a pour label 1, on dit que le nœud est occupé, sinon, il est inoccupé.

#### 1.2.2 Permutations

Un problème classique de distribution sur les permutations est lié à la recherche du permanent d'une matrice non-négative.

#### Exemple

Pour une matrice non-négative  $w(i, j)$ , notons :

$$w(\sigma) = \prod_{i=1}^n w(i, \sigma(i))$$

Tant qu'il existe au moins une permutation  $\sigma$  telle que  $w(i, \sigma(i)) > 0$  pour tout  $i$ , cela donne une densité non normalisée sur l'ensemble des permutations. La constante de normalisation pour cette densité est alors appelée le permanent de la matrice  $w(i, j)$ . Si aucun  $\sigma$  ne vérifie cette relation, le permanent est 0.

Une autre distribution importante que l'on va considérer est la distribution uniforme sur les permutations où certains objets doivent avoir une position plus faible que les autres.

#### Définition 11. Relation d'ordre partiel

Soit un ensemble  $P$ . Une relation d'ordre partiel sur  $P$  est une relation binaire  $\preceq$  telle que pour tout  $a, b, c \in P$ , la relation est :

1. (Réflexive)  $a \preceq a$
2. (Antisymétrique) Si  $a \preceq b$  et  $b \preceq a$ , alors  $a = b$
3. (Transitive) Si  $a \preceq b$  et  $b \preceq c$ , alors  $a \preceq c$

Un ensemble disposant d'une relation d'ordre partiel est parfois appelé poset d'après l'anglais partially ordered set.

#### Définition 12. Extension linéaire d'un ordre partiel



Une extension linéaire d'un ordre partiel sur  $1, \dots, n$  est une permutation pour laquelle si  $i$  et  $j$  sont tels que  $\sigma(i) \prec \sigma(j)$ , alors  $i < j$ .

### 1.3 Chaînes de Markov et simulation approchée

Jusqu'au développement des algorithmes de simulation parfaite/exacte, la manière principale d'obtenir une réalisation d'une distribution ciblée était une méthode d'approche. Plusieurs algorithmes et méthodes ont été mis en place et l'ensemble de ces méthodes porte le nom de Chaîne de Markov Monte Carlo (Markov Chain Monte Carlo, MCMC dans la littérature). On ne rappellera pas ici les principales définitions pour les chaînes de Markov mais d'autres seront nécessaires pour la suite. On s'intéressera notamment aux chaînes de Harris et au théorème ergodique associé.

#### Définition 13. Chaîne de Harris

Une chaîne de Markov  $\{X_t\}$  sur un espace d'état  $\Omega$  est une chaîne de Harris si il existe des ensembles mesurables  $A, B \in \Omega$  et  $\epsilon > 0$  pour  $x \in A$  et  $y \in B$ , et une mesure de probabilité  $\rho$  où  $\rho(B) = 1$  tels que l'on ait :

1. Pour  $T_A = \inf \{t \geq 0 : X_t \in A\}$ ,  $(\forall z \in \Omega)(\mathbb{P}(T_A < \infty | X_0 = z) > 0)$ .
2. Si  $x \in A$  et  $C \subseteq B$ , alors  $\mathbb{P}(X_1 \in C | X_0 = x) \geq \epsilon \rho(C)$

#### Définition 14. Chaîne récurrente

Soit  $R = \inf \{n > 0 : X_n \in A\}$ . On dit qu'une chaîne de Harris est une chaîne récurrente si pour tout  $x \in A$ ,  $\mathbb{P}(R < \infty | X_0 = x) = 1$ . Une chaîne qui n'est pas récurrente est dite transiente.

#### Définition 15. Chaîne apériodique

Une chaîne de Harris récurrente est apériodique si pour tout  $x \in \Omega$ , il existe  $n$  tel que pour tout  $n' \geq n$ ,  $\mathbb{P}(X_{n'} \in A | X_0 = x) > 0$

**Théorème 3** (Théorème ergodique pour les chaînes de Harris). *Soit  $X_n$  une chaîne de Harris récurrente et apériodique de distribution stationnaire  $\pi$ . Si  $\mathbb{P}(R < \infty | X_0 = x) = 1$  pour tout  $x$ , alors, pour  $t \rightarrow \infty$ , pour tout ensemble mesurable  $C$  et pour tout  $x$  :*

$$|\mathbb{P}(X_t \in C | X_0 = x) - \pi(C)| \rightarrow 0$$

Ce théorème est le cœur des méthodes MCMC, puisqu'il "suffit" de construire une chaîne de Harris ayant pour distribution stationnaire la distribution souhaitée et de la faire tourner pendant un nombre infini de pas. Cependant, n'ayant pas un temps infini, les utilisateurs font tourner leurs algorithmes durant un grand nombre de pas et espèrent arriver dans la distribution stationnaire.

Nous pourrions cependant déterminer à quel point la chaîne est proche de la loi stationnaire à l'aide du concept de couplage (voir chapitre 3 pour une définition et application étendue).

**Définition 16.** *Couplage*

Supposons que  $\{X_t\} \sim \nu_X$  et  $\{Y_t\} \sim \nu_Y$ . Un couplage de  $\{X_t\}$  et  $\{Y_t\}$  est un processus bivarié  $\{(X'_t, Y'_t)\}$  tel que  $\{X'_t\} \sim \nu_X$  et  $\{Y'_t\} \sim \nu_Y$ .

**Théorème 4** (Lemme de Couplage). *Soit  $Y_0 \sim \pi$  et  $X_0 = x_0$  tels que les deux variables évoluent de manière couplée. Alors, pour tout mesurable  $C$  :*

$$|\mathbb{P}(X_t \in C | X_0 = x) - \pi(C)| \leq \mathbb{P}(X_t \neq Y_t).$$

## 1.4 Création de chaînes de Markov

Afin d'utiliser les méthodes MCMC, il faut créer des chaînes de Harris qui convergent vers la distribution  $\pi$  ciblée. Il est généralement mieux de créer des chaînes réversibles plutôt que des chaînes simplement stationnaires.

Nous utiliserons la notation suivante pour la définition de réversibilité :  $\pi(dx) = f(x)dx$ . Et donc, pour tout mesurable  $A$ ,  $\pi(A) = \int_{x \in A} \pi(dx) = \int_{x \in A} f(x)dx$ .

Nous utiliserons l'équation de balance détaillée pour en déduire la réversibilité :

**Définition 17.** *Équation de balance détaillée*

Une distribution  $\pi$  est réversible selon une chaîne de Markov  $\{X_t\}$  en particulier si :  $\pi(dx)\mathbb{P}(X_{t+1} \in dy | X_t = x) = \pi(dy)\mathbb{P}(X_{t+1} \in dx | X_t = y)$ .

**Lemme 1.** *Si  $\pi$  est réversible, alors  $\pi$  est stationnaire.*

*Démonstration.* Soit  $\Omega$  l'espace d'état de la chaîne de Markov  $\{X_t\}$  considérée et  $\pi$  réversible pour cette chaîne. Pour  $X_t \sim \pi$ , et  $C$  un ensemble mesurable, alors on a :

$$\begin{aligned}
\mathbb{P}(X_{t+1} \in C) &= \mathbb{E}[\mathbf{1}(X_{t+1} \in C)] \\
&= \mathbb{E}[\mathbb{E}[\mathbf{1}(X_{t+1} \in C) | X_t]] \\
&= \int_{x \in \Omega} \mathbb{E}[\mathbf{1}(X_{t+1} \in C) | X_t = x] \pi(dx) \\
&= \int_{x \in \Omega} \mathbb{P}(X_{t+1} \in C | X_t = x) \pi(dx) \\
&= \int_{x \in \Omega} \int_{y \in C} \mathbb{P}(X_{t+1} \in dy | X_t = x) \pi(dx) \\
&= \int_{y \in C} \int_{x \in \Omega} \mathbb{P}(X_{t+1} \in dx | X_t = y) \pi(dy) \\
&= \int_{y \in C} \mathbb{P}(X_{t+1} \in \Omega | X_t = y) \pi(dy) \\
&= \int_{y \in C} \pi(dy) \\
&= \pi(C)
\end{aligned}$$

D'où la stationnarité de  $\pi$ .

□

Plusieurs types de chaînes réversibles existent telles que l'échantillonnage de Gibbs (Gibbs sampler), Metropolis-Hastings, etc. Ces chaînes sont présentées dans la suite.

#### 1.4.1 Échantillonnage de Gibbs

On présente ici un échantillonneur de Gibbs, qui agit sur un espace d'états de la forme  $C^V$ . On appelle  $\nu \in V$  une dimension du problème. Pour  $X_t = x$ , l'échantillonneur choisit une dimension  $\nu$  uniformément sur  $V$ . Soit  $L(x, \nu)$  l'ensemble des états qui sont exactement les nœuds de la configuration  $x$  sauf en  $\nu$ , écrit de la manière suivante :  $L(x, \nu) = \{y : (\forall w \in V \setminus \{\nu\})(y(w) = x(w))\}$ . Pour  $X_t = x$ , l'état suivant  $X_{t+1}$  est choisi selon  $\pi$  à conditionné à être dans  $L(x, \nu)$ .

On présente comme exemple le modèle d'Ising précédemment vu. Une dimension est alors un nœud de la configuration. On choisit donc un nœud uniformément dans  $V$ , et on considère les états qui sont exactement  $x$  en tous les autres nœuds autres que  $\nu$ . Pour le modèle d'Ising, la valeur en  $\nu$  de  $x$  est 1 ou  $-1$ . On note ces configurations  $x_{\nu \rightarrow 1}$  et  $x_{\nu \rightarrow -1}$ . On choisit alors l'état suivant entre  $x_{\nu \rightarrow 1}$  et  $x_{\nu \rightarrow -1}$ , où le choix est fait proportionnellement à  $\pi$ . On a donc :

$$\mathbb{P}(X_{t+1} = x_{\nu \rightarrow 1}) = \frac{\pi(\{x_{\nu \rightarrow 1}\})}{\pi(\{x_{\nu \rightarrow 1}\}) + \pi(\{x_{\nu \rightarrow -1}\})}$$

Or, pour le modèle d'Ising,

$$\pi(\{x\}) = \frac{1}{Z} \prod_{i \in V} \exp(\mu X(i)) \prod_{\{i,j\} \in E} \exp(\beta \mathbf{1}(x(i) = x(j))).$$

Après simplification, on obtient :

$$\begin{aligned} \mathbb{P}(X_{t+1} = x_{\nu \rightarrow 1}) &= \frac{\exp(\mu) \prod_{\{\nu,j\} \in E} \exp(\beta \mathbf{1}(x(j) = 1))}{\exp(\mu) \prod_{\{\nu,j\} \in E} \exp(\beta \mathbf{1}(x(j) = 1)) + \exp(-\mu) \prod_{\{\nu,j\} \in E} \exp(\beta \mathbf{1}(x(j) = -1))} \\ &= \frac{\exp(\beta n_1 + \mu)}{\exp(\beta n_1 + \mu) + \exp(\beta n_{-1} - \mu)} \end{aligned}$$

Où  $n_c$  est le nombre de voisins de  $\nu$  de label  $c$ . Par exemple, si  $\nu$  est adjacent à trois nœuds de label 1 et un nœud de label  $-1$ , alors la probabilité que  $\nu$  se voit labelliser 1 est  $\exp(3\beta + \mu) / (\exp(3\beta + \mu) + \exp(\beta - \mu))$ . Mettre ensuite en place cette méthode algorithmiquement est très simple.

Vérifions la réversibilité de la chaîne.

*Démonstration.* Les cas où la chaîne ne change pas d'état après un pas, (ie  $X_t = x_{\nu \rightarrow 1}$  et  $X_{t+1} = x_{\nu \rightarrow 1}$  ou  $X_t = x_{\nu \rightarrow -1}$  et  $X_{t+1} = x_{\nu \rightarrow -1}$ ) nous donnent immédiatement l'équation de balance détaillée, il ne reste alors qu'à traiter deux cas.

— Premier cas :  $X_t = x_{\nu \rightarrow 1}$  et  $X_{t+1} = x_{\nu \rightarrow -1}$ .

On a alors :

$$\pi(\{x_{\nu \rightarrow 1}\}) \mathbb{P}(X_{t+1} = x_{\nu \rightarrow -1} | X_t = x_{\nu \rightarrow 1}) = \pi(\{x_{\nu \rightarrow 1}\}) \frac{\exp(-\mu + \beta n_{-1})}{\exp(-\mu + \beta n_{-1}) + \exp(\mu + \beta n_1)}$$

D'autre part :

$$\pi(\{x_{\nu \rightarrow -1}\}) \mathbb{P}(X_{t+1} = x_{\nu \rightarrow 1} | X_t = x_{\nu \rightarrow -1}) = \pi(\{x_{\nu \rightarrow -1}\}) \frac{\exp(\mu + \beta n_1)}{\exp(-\mu + \beta n_{-1}) + \exp(\mu + \beta n_1)}$$

On aura donc l'égalité entre ces termes si et seulement si on a :

$$\pi(\{x_{\nu \rightarrow 1}\}) \exp(-\mu + \beta n_{-1}) = \pi(\{x_{\nu \rightarrow -1}\}) \exp(\mu + \beta n_1)$$

Les simplifications suivantes s'opèrent dans les termes en  $\pi$  :

- simplification des  $\frac{1}{Z}$
- simplification des  $\prod_{i \in V} \exp(\mu X(i))$  sauf au nœud  $\nu$
- simplification des  $\prod_{\{i,j\} \in E} \exp(\beta \mathbf{1}(x(i) = x(j)))$  sauf aux arêtes ayant  $\nu$  comme extrémité

On obtient alors :

$$\exp(\mu + \beta n_1) \exp(-\mu + \beta n_{-1}) = \exp(-\mu + \beta n_{-1}) \exp(\mu + \beta n_1)$$

On a donc bien vérifié l'égalité et donc l'équation de balance détaillée.

Le deuxième cas :  $X_t = x_{\nu \rightarrow -1}$  et  $X_{t+1} = x_{\nu \rightarrow 1}$  se vérifie de la même manière.

Ayant vérifié tous les cas possibles, on obtient donc l'équation de balance détaillée qui conduit bien à la réversibilité de la chaîne.

□

## Programmation sous R

```
1  ## X = graphe (rectangulaire ou carré), u = lancer uniforme sur
   [0,1], v = noeud de X ##
2  IsingGibbsUpdate <- function(X,u,v,beta,mu){ ## Permet de faire un
   pas de la chaîne de Markov associée selon le Gibbs Sampler##
3
4  ##On entoure la matrice de valeurs aberrantes pour la manipuler
   plus facilement##
5  ##On change les indices du point considéré pour se conformer à la
   nouvelle matrice##
6
7  i = v[1] + 1
8  j = v[2] + 1
9
10 X = rbind(rep(0,dim(X)[2]),X)
11 X = rbind(X,rep(0,dim(X)[2]))
12 X = cbind(rep(0,dim(X)[1]),X)
13 X = cbind(X,rep(0,dim(X)[1]))
14
15 ##Les indicatrices permettent de vérifier quelles valeurs ont les
   voisins du point considéré ##
16
17 if(u< exp(mu + beta*((X[i+1,j]==1)*1 + (X[i-1,j]==1)*1 +(X[i,j
   +1]==1)*1 + (X[i,j-1]==1)*1))/((exp(mu + beta*((X[i+1,j]==1)*1
   + (X[i-1,j]==1)*1 +(X[i,j+1]==1)*1 + (X[i,j-1]==1)*1))) + exp
   (-mu + beta*((X[i+1,j]==-1)*1 + (X[i-1,j]==-1)*1 +(X[i,j
   +1]==-1)*1 + (X[i,j-1]==-1)*1))))){
18   X[i,j] = 1
19 }
20 else{
21   X[i,j] = -1
22 }
23
24 ##On supprime les valeurs insérées##
25 X=X[,-1]
26 X=X[,-dim(X)[2]]
27 X=X[-1,]
28 X=X[-dim(X)[1],]
29
30 X
31 }
```

```

1  IsingGibbssteps <- function(X,beta,mu,n){ ##effectue n pas de la
      chaîne de Markov pour le sampler de Gibbs, pour un graphe
      rectangulaire X##
2
3  for (i in 1:n){
4
5      ##choix uniforme du noeud considéré##
6      k = sample(1:dim(X)[1],1)
7      l = sample(1:dim(X)[2],1)
8      ##lancer uniforme sur [0,1]##
9      u = runif(1)
10     ##mise à jour du point##
11     X=IsingGibbsUpdate(X,u,c(k,l),beta,mu)
12
13 }
14
15 X
16 }

```

```

1  IsingGibbsconditionnalstop <- function(X,beta,epsilon){ ##effectue
    un nombre aléatoire de pas de la chaîne de Markov pour le
    sampler de Gibbs, pour un graphe carré X##
2
3  ##On crée le vecteur Y qui contiendra les ratios de 1 pour chaque
    graphe##
4  Y=c(mean(X==1))
5  bol = TRUE
6  i=1
7
8  while (bol){
9
10     ##Choix uniforme du noeud considéré##
11     k = sample(1:dim(X)[1],1)
12     l = sample(1:dim(X)[2],1)
13
14     ##lancer uniforme sur [0,1]##
15     u = runif(1)
16
17     ##Mise à jour du noeud considéré##
18     X=IsingGibbsUpdate(X,u,c(k,l),beta)
19
20     ##Mise à jour du vecteur des ratios##
21     Y = c(Y,mean(X==1))
22
23     ##Si la ligne est décommentée, on obtient une représentation de
        la configuration X tout les (dim(X)[1]**2)/epsilon pas##
24     ##if ((i%((dim(X)[1]**2)/epsilon))==0){image(X)}
25
26     ##Si on a effectué un nombre de pas suffisant, et que le nombre de
        1 n'a pas varié énormément durant un certain nombre d'
        itérations, on s'arrête##
27     if ((i>=((log(dim(X)[1]**2)*dim(X)[1]**2)/epsilon)) && ((mean(Y[
        i+1-floor((dim(X)[1]**(1/2))/epsilon):i])-Y[i])<epsilon)){
28         break()
29     }
30     i=i+1
31 }
32
33 return(list(X,i))
34
35 }

```



### 1.4.2 Metropolis-Hastings

Pour la méthode de Metropolis-Hastings, pour chaque configuration  $x$  de la chaîne, on a une densité  $q_x$ . On propose ensuite un état suivant  $y$ , selon la densité  $q_x$ . L'état suivant sera donc  $x$  ou  $y$ . La probabilité avec laquelle est choisie la configuration  $y$  est donnée de telle sorte que la réversibilité s'applique. On la donne ci-dessous. Notez que, pour que cela fonctionne, il est nécessaire que si  $q_x(y) > 0$ , alors  $q_y(x) > 0$  aussi. On a alors le changement vers  $y$  avec probabilité :

$$\min \left\{ 1, \frac{f_\pi(y)q_y(x)}{f_\pi(x)q_x(y)} \right\}$$

On notera aussi que  $f_\pi$  peut-être normalisée ou non.

En exemple, on considère à nouveau le modèle d'Ising. A chaque pas, on choisit un nœud uniformément dans  $V$  puis un label candidat pour ce nœud est choisi uniformément dans  $\{-1, 1\}$ . On a alors  $q_x(y) = q_y(x) = 1/2$ . On calcule ensuite, pour  $n_c$  le nombre de voisins de  $\nu$  de label  $c$  (où  $c$  est le label proposé) et  $n_{x(\nu)}$  le nombre de voisins de  $\nu$  de même label que  $\nu$  :

$$\frac{f_\pi(y)}{f_\pi(x)} = \frac{\exp(n_c\beta + c\mu)}{\exp(n_{x(\nu)}\beta + x(\nu)\mu)}$$

L'algorithme est alors très simple à mettre en place. D'abord, vérifions la réversibilité de la chaîne.

*Démonstration.* Les cas où la chaîne ne change pas d'état après un pas, (ie  $X_t = x_{\nu \rightarrow 1}$  et  $X_{t+1} = x_{\nu \rightarrow 1}$  ou  $X_t = x_{\nu \rightarrow -1}$  et  $X_{t+1} = x_{\nu \rightarrow -1}$ ) nous donnent immédiatement l'équation de balance détaillée, il ne reste alors qu'à traiter deux cas.

— Premier cas :  $X_t = x_{\nu \rightarrow 1}$  et  $X_{t+1} = x_{\nu \rightarrow 1}$ .

On a alors :

$$\pi(\{x_{\nu \rightarrow 1}\})\mathbb{P}(X_{t+1} = x_{\nu \rightarrow 1} | X_t = x_{\nu \rightarrow 1}) = \pi(\{x_{\nu \rightarrow 1}\})\min \left\{ 1, \frac{\exp(-\mu + \beta n_{-1})}{\exp(\mu + \beta n_1)} \right\}$$

D'autre part :

$$\pi(\{x_{\nu \rightarrow -1}\})\mathbb{P}(X_{t+1} = x_{\nu \rightarrow 1} | X_t = x_{\nu \rightarrow -1}) = \pi(\{x_{\nu \rightarrow -1}\})\min \left\{ 1, \frac{\exp(\mu + \beta n_1)}{\exp(-\mu + \beta n_{-1})} \right\}$$

On aura donc l'égalité entre ces termes si et seulement si on a :

$$\pi(\{x_{\nu \rightarrow 1}\})\min \left\{ 1, \frac{\exp(-\mu + \beta n_{-1})}{\exp(\mu + \beta n_1)} \right\} = \pi(\{x_{\nu \rightarrow -1}\})\min \left\{ 1, \frac{\exp(\mu + \beta n_1)}{\exp(-\mu + \beta n_{-1})} \right\}$$

Les simplifications suivantes s'opèrent dans les termes en  $\pi$  :

- simplification des  $\frac{1}{Z}$
- simplification des  $\prod_{i \in V} \exp(\mu X(i))$  sauf au nœud  $\nu$
- simplification des  $\prod_{\{i,j\} \in E} \exp(\beta \mathbf{1}(x(i) = x(j)))$  sauf aux arêtes ayant  $\nu$  comme extrémité

On obtient alors :

$$\exp(\mu + \beta n_1) \min \left\{ 1, \frac{\exp(-\mu + \beta n_{-1})}{\exp(\mu + \beta n_1)} \right\} = \exp(-\mu + \beta n_{-1}) \min \left\{ 1, \frac{\exp(\mu + \beta n_1)}{\exp(-\mu + \beta n_{-1})} \right\}$$

Par propriété du min, on obtient alors l'égalité entre les termes, qui conduit à l'équation de balance détaillée.

Le deuxième cas :  $X_t = x_{\nu \rightarrow -1}$  et  $X_{t+1} = x_{\nu \rightarrow 1}$  se vérifie de la même manière.

Ayant vérifié tous les cas possibles, on obtient donc l'équation de balance détaillée qui conduit bien à la réversibilité de la chaîne.

□

## Programmation sous R

```
1  ## X = graphe (rectangulaire ou carré), u = lancer uniforme sur
   [0,1], v = noeud de X ##
2
3  MetropolisHastingsUpdate <- function(X,u,v,beta){ ## Permet de faire
   un pas de la chaîne de Markov associée selon Metropolis-
   Hastings
4
5  ##On entoure la matrice de valeurs aberrantes pour la manipuler
   plus facilement##
6  ##On change les indices du point considéré pour se conformer à la
   nouvelle matrice##
7
8  i = v[1] + 1
9  j = v[2] + 1
10
11 X = rbind(rep(0,dim(X)[2]),X)
12 X = rbind(X,rep(0,dim(X)[2]))
13 X = cbind(rep(0,dim(X)[1]),X)
14 X = cbind(X,rep(0,dim(X)[1]))
15
16 ##On choisit une couleur candidate pour le noeud v##
17
18 c=sample(c(-1,1),1,prob=c(1/2,1/2))
19
20 ##Les indicatrices permettent de vérifier quelles valeurs ont les
   voisins du point considéré ##
21
22 if(u< exp(beta*((X[i+1,j]==c)*1 + (X[i-1,j]==c)*1 +(X[i,j+1]==c)*1
   + (X[i,j-1]==c)*1))/exp(beta*((X[i+1,j]==X[i,j])*1 + (X[i-1,j]
   )==X[i,j])*1 +(X[i,j+1]==X[i,j])*1 + (X[i,j-1]==X[i,j])*1))) {
23   X[i,j] = c
24 }
25
26 ##On supprime les valeurs insérées##
27 X=X[,-1]
28 X=X[,-dim(X)[2]]
29 X=X[-1,]
30 X=X[-dim(X)[1],]
31
32 X
33 }
```

```

1  MetropolisHastingssteps <- function(X,beta,n){ ##effectue n pas de
    la chaîne de Markov pour Metropolis-Hastings, pour un graphe
    rectangulaire X##
2
3  for (i in 1:n){
4
5    ##choix uniforme du noeud##
6    k = sample(1:dim(X)[1],1)
7    l = sample(1:dim(X)[2],1)
8
9    ##lancer uniforme sur [0,1]##
10   u = runif(1)
11
12   ##mise à jour de la configuration X##
13   X=MetropolisHastingsUpdate(X,u,c(k,l),beta)
14 }
15
16 X
17
18 }

```

```

1
2 MetropolisHastingsconditionnalstop <- function(X,beta,epsilon){ ##
   effectue un nombre aléatoire de pas de la chaîne de Markov pour
   Metropolis-Hastings, pour un graphe carré X##
3
4 ##On créé le vecteur Y qui contiendra les ratios de 1 pour chaque
   graphe##
5 Y=c(mean(X==1))
6 bol = TRUE
7 i=1
8
9
10 while (bol){
11
12   ##Choix uniforme du noeud considéré##
13   k = sample(1:dim(X)[1],1)
14   l = sample(1:dim(X)[2],1)
15
16   ##lancer uniforme sur [0,1]##
17   u = runif(1)
18
19   ##Mise à jour du noeud considéré##
20   X=MetropolisHastingsUpdate(X,u,c(k,l),beta)
21
22   ##Mise à jour du vecteur des ratios##
23   Y = c(Y,mean(X==1))
24
25   ##Si la ligne est décommentée, on obtient une représentation de la
   configuration X tout les (dim(X)[1]**2)/epsilon pas##
26   ##if ((i%%((dim(X)[1]**2)/epsilon)==0)){image(X)}##
27
28   ##Si on a effectué un nombre de pas suffisant, et que le nombre de
   1 n'a pas varié énormément durant un certain nombre d'
   itérations, on s'arrête##
29   if ((i>=((log(dim(X)[1]**2)*dim(X)[1]**2)/epsilon)) && ((mean(Y[
   i+1-floor((dim(X)[1]**(1/2)))/epsilon:i])-Y[i])<epsilon)){
30     break()
31   }
32   i=i+1
33 }
34
35 return(list(X,i))
36
37 }

```

## Discussion du critère d'arrêt dans "IsingGibbsconditionnalstop"

### et "MetropolisHastingsconditionnalstop"

Le premier critère pour l'arrêt du programme est si le nombre de pas  $i$  vérifie :  $i \geq ((\log(n^2) * n^2) / \epsilon)$ , où  $n^2$  est la taille du graphe carré considéré et  $\epsilon$  est la "précision" que l'utilisateur souhaite.

Le terme en  $n^2 * \log(n^2)$  provient du problème du collectionneur. En effet, on souhaite, pour atteindre un certain équilibre, avoir donné la chance à chacun des nœuds d'avoir subi un changement. Cela revient à avoir sélectionné chacun des nœuds au moins une fois, ce qui revient au problème du collectionneur, où en moyenne le temps d'obtention de tous les éléments de la collection se fait en  $n * \log(n)$  si  $n$  est la taille de la collection complète. Le terme en  $\epsilon$  permet de s'assurer d'avoir obtenu la visite complète de tous les nœuds, mais servira plus amplement dans l'autre condition d'arrêt présentée ci-après.

Le second critère est si le vecteur des ratios de 1 vérifie :

$$|Y[i] - \frac{1}{\lfloor \frac{n}{\epsilon} \rfloor} \sum_{k=i+1-\lfloor \frac{n}{\epsilon} \rfloor}^i Y[k]| < \epsilon$$

C'est-à-dire, si le ratio de 1 n'a pas évolué de plus de  $\epsilon$  en moyenne depuis le temps  $i + 1 - \lfloor \frac{n}{\epsilon} \rfloor$ . On parle donc de "précision"  $\epsilon$  puisque plus cette valeur est petite, plus le temps requis pour que peu de changements s'effectuent en moyenne est grand.

Le terme en  $n$  (où  $n^2$  est la taille de la matrice) est ajouté pour obtenir une dépendance à la dimension dans la précision.

### **1.4.3 Variables aléatoires auxiliaires / Auxiliary random variables**

Dans la plupart des applications du MCMC, la densité ciblée  $X$  a une structure multiplicative. Pour ces densités, il est possible d'ajouter un vecteur  $Y$  de variables aléatoires supplémentaires telles que  $(X, Y)$  a une distribution jointe plus simple. Par exemple, on considère à nouveau le modèle d'Ising avec  $\mu = 0$ . Pour chaque arête  $\{i, j\}$ , on crée une variable aléatoire auxiliaire  $Y(\{i, j\})$  telle que sa distribution conditionnée sur  $X$  soit uniforme sur  $[0, 1]$  si  $X(i) \neq X(j)$  et uniforme sur  $[0, \exp(\beta)]$  si on a l'égalité.

La densité jointe est alors uniforme sur :

$$X \in \{0, 1\}^V \text{ et } Y \in \{[0, \infty) : (\forall \{i, j\})(y\{i, j\}) \leq \min(\exp(\beta), 1))\}$$

La chaîne de Markov est alors la suivante : pour  $X$  donné, il suffit de choisir une nouvelle valeur de  $Y$  sachant  $X$ , puis choisir un nouveau  $X$  conditionné sur  $Y$ .

De par la construction de la chaîne, tirer  $Y$  sachant  $X$  est direct. Cependant,

tirer  $X$  sachant  $Y$  est une autre histoire : prenons pour exemple le modèle d'Ising avec  $\beta > 0$ . On a alors  $\exp(\beta) > 1$ . Lorsque  $y(\{i, j\}) \in [0, 1]$ , alors il est possible que  $x(i) = x(j)$  ou que  $x(i) \neq x(j)$ , mais lorsque  $y(\{i, j\}) > 1$ , alors on a forcément que  $x(i) = x(j)$ . Les arêtes avec  $y(\{i, j\}) > 1$  séparent le graphe en groupes de composants connectés tels que chacun des composants doivent avoir le même label.

Il suffit donc de séparer le graphe en groupes de composants connectés en se servant des arêtes  $y(\{i, j\}) > 1$ , puis il faut choisir un label uniformément sur  $\{-1, 1\}$  pour les composantes de ces groupes.

#### 1.4.4 Shift Chains

Un autre type de chaîne utile pour les modèles répulsifs est le shift. Pour illustrer le modèle, on considère le modèle hard-core où chaque nœud du graphe est soit occupé ( $x(\nu) = 1$ ) soit inoccupé ( $x(\nu) = 0$ ). Chaque nœud occupé contribue d'un facteur  $\lambda$  à la densité.

Dès lors qu'un des voisins de  $\nu$  est occupé, la chaîne ne peut changer la valeur de  $\nu$ .

Le shift autorise un échange de label entre  $\nu$  et son voisin occupé avec une certaine probabilité  $p_{shift}$ . Lorsque deux voisins de  $\nu$  ou plus sont occupés, aucun changement n'est possible et  $\nu$  reste inoccupé.

## 2 Chapitre 2 : Procédures de rejet

### 2.1 Théorème et exemple simple

**Théorème 5.** Soit  $\nu$  une mesure finie sur un ensemble  $B$  et soit  $A$  un sous-ensemble de  $B$  tel que  $\nu(A) > 0$ . Soit  $X_1, X_2, \dots$  des variables aléatoires iid tirées selon  $\nu$  sur l'ensemble  $B$ . Soit enfin  $T = \inf \{t : X_t \in A\}$ . Alors :

$$X_T \sim [X_1 | X_1 \in A]$$

C'est à dire, si on tire successivement des valeurs dans  $B$  selon  $\nu$  et que l'on rend la première valeur dans  $A$ , alors le résultat provient de la mesure  $\nu$  conditionnée à être dans  $A$ .

*Démonstration.* Cela suit du théorème fondamental de la simulation parfaite en ayant  $U_1, U_2, \dots$  des tirages dans  $B$  selon  $\nu$ ,  $b(U) = \mathbf{1}(U \in A)$ ,  $g(U) = U$  et  $f(X, U) = X$ .  $\square$

Le nombre moyen de pas nécessaire pour obtenir un tirage dans  $A$  dépend de la taille de  $A$  par rapport à la taille de  $B$ .

**Théorème 6.** La probabilité qu'un tirage  $X$  dans  $B$  selon  $\nu$  soit dans  $A$  est  $\nu(A)/\nu(B)$ . Le nombre moyen de tirages nécessaires est  $\nu(B)/\nu(A)$ .

*Démonstration.* La première moitié du théorème découle simplement de la définition de mesure de probabilité. La seconde provient du fait que  $T$  soit une loi géométrique de paramètre  $\nu(A)/\nu(B)$ , et donc sa moyenne est  $\nu(B)/\nu(A)$ .  $\square$

#### Exemple : tirage uniforme sur le cercle unité

On choisit  $B = [-1, 1] \times [-1, 1]$  et  $A = \{(x, y) \in \mathbb{R}^2 | x^2 + y^2 \leq 1\}$ .

L'aire de  $B$  est 4 tandis que l'aire du cercle unité est  $\pi$ . Le nombre moyen de lancers est alors  $2 * (4/\pi) = 2 * 1.273$ , où le facteur 2 provient du nombre d'uniformes nécessaires pour obtenir un tirage dans  $B$ .

On propose une mise en place sous R en fin de chapitre.

On déduit de cette méthode un moyen de générer une réalisation de Cauchy standard :

**Théorème 7.** Si  $(X, Y)$  est un tirage aléatoire sur le cercle unité de  $\mathbb{R}^2$ , alors  $X/Y$  est un tirage aléatoire selon la loi de Cauchy standard.

*Démonstration.* Soit  $\phi$  une fonction borélienne bornée sur  $\mathbb{R}$ . On a, pour  $(X, Y)$  variable aléatoire uniforme sur le disque unité :

$$\mathbb{E} \left[ \phi \left( \frac{Y}{X} \right) \right] = \frac{1}{\pi} \int_{\mathbb{R}^2} \phi \left( \frac{y}{x} \right) \mathbf{1}_{\{x^2 + y^2 \leq 1\}} dx dy$$



On passe en coordonnées polaires :  $y = r \sin(\theta)$ ,  $x = r \cos(\theta)$ . On a alors  $dx dy = r dr d\theta$  et on passe d'une intégrale sur  $\mathbb{R}^2$  vers une intégrale sur  $\mathbb{R}^+ \times [0, 2\pi]$ , d'où :

$$\begin{aligned}\mathbb{E} \left[ \phi \left( \frac{Y}{X} \right) \right] &= \frac{1}{\pi} \int_{\mathbb{R}^+ \times [0, 2\pi]} \phi(\tan(\theta)) \mathbf{1}_{\{r \leq 1\}} r dr d\theta \\ &= \frac{1}{2\pi} \int_0^{2\pi} \phi(\tan(\theta)) d\theta \\ &= 2 * \frac{1}{2\pi} \int_{-\pi/2}^{\pi/2} \phi(\tan(\theta)) d\theta\end{aligned}$$

On pose ensuite le changement de variables  $u = \tan(\theta)$  pour enfin obtenir :

$$\mathbb{E} \left[ \phi \left( \frac{Y}{X} \right) \right] = \int_{\mathbb{R}^+} \phi(u) \frac{1}{\pi(1+u^2)} du$$

Ce qui donne bien la densité d'une loi de Cauchy standard. On obtient aussi le résultat pour  $X/Y$  puisque l'inverse de la loi de Cauchy standard reste la loi de Cauchy standard.

□

## 2.2 Méthode de rejet pour variables aléatoires à densité

La méthode de rejet de base n'inclut pas les variables aléatoires à densité. Cependant, celles-ci sont traitées aisément à l'aide des variables aléatoires auxiliaires. Pour cela, on considère le théorème suivant, provenant de la théorie de la mesure.

**Théorème 8** (Théorème fondamental de la simulation de Monte Carlo). *Soit  $X$  variable aléatoire de densité (possiblement non-normalisée)  $f_X$  par rapport à une mesure  $\nu$  sur  $\Omega$ . Si on a  $[Y|X] \sim \text{Unif}([0, f_X])$ , alors  $(X, Y)$  est un lancer provenant de la mesure produit  $\nu \times \text{Unif}$  sur l'ensemble  $\{(x, y) : x \in \Omega, 0 \leq y \leq f_x\}$ .*

Essentiellement, ce théorème nous dit que pour obtenir un lancer selon une certaine densité, il suffit de tirer selon une loi uniforme sur un espace plus grand. De plus, la densité de  $X$  n'est pas forcément normalisée, ce qui sera essentiel dans les applications qui suivent.

Pour comprendre comment cela fonctionne, nous allons traiter un exemple en toute généralité, puis un exemple concret.

Soit  $\mu$  une mesure sur un ensemble  $\Omega$ . On suppose que, pour une densité  $g$ , il est possible de générer un lancer provenant de la densité produit  $\mu \times \text{Unif}$  sur  $\Omega_g = \{(x, y) : x \in \Omega, 0 \leq y \leq g\}$ .

Une fois que ce lancer est obtenu,  $X$  provient alors de la densité  $g$ . Mais

supposons ici que le but est d'obtenir un lancer selon une densité  $f$ .  
Notons d'abord que, si  $(X, Y)$  est de mesure  $\mu \times \text{Unif}$  sur  $\Omega_g$ , alors  $(X, 2Y)$  est de mesure  $\mu \times \text{Unif}$  sur  $\Omega_{2g} = \{(x, y) : x \in \Omega, 0 \leq y \leq 2g\}$ . On peut même généraliser : pour toute constante  $c > 0$ ,  $(X, cY)$  est de mesure  $\mu \times \text{Unif}$  sur  $\Omega_{cg}$ .

Supposons alors que  $c \geq \sup_{x \in \Omega} f(x)/g(x)$ . Alors :

$$\Omega_f \subseteq \Omega_{cg}$$

On peut alors utiliser une méthode de rejet pour obtenir une réalisation provenant de la densité  $f$ . Il suffit de générer  $X$  selon la densité  $g$ , puis de tirer uniformément  $Y$  sur  $[0, c * g(X)]$ . Si  $Y$  est aussi dans  $[0, f(X)]$  (de sorte que  $(X, Y) \in \Omega_f$ ) alors on accepte  $(X, y)$ , ce qui signifie que  $X$  est un lancer selon la densité  $f$ .

Pour tester si le lancer uniforme de  $[0, c * g(X)]$  est dans  $[0, f(X)]$ , il n'est pas nécessaire de vérifier si l'uniforme tombe effectivement dedans : on peut simplement tirer une variable aléatoire  $C$  suivant une loi de Bernoulli de paramètre  $f(X)/[c * g(X)]$ . Si  $C = 1$ , on considère que  $X$  est un lancer provenant de la densité  $f$ .

La probabilité d'acceptation varie inversement avec  $c$ , le meilleur choix possible pour  $c$  est alors  $\sup_{x \in \Omega} f(x)/g(x)$ . La valeur de  $c$  n'est pas toujours évidente à calculer. Dans ce cas, une majoration suffit : l'algorithme fera simplement plus de lancers en moyenne.

Nous allons maintenant calculer la probabilité d'acceptation d'un lancer.

**Lemme 2.** Soit  $Z_f = \int_{x \in \Omega} f(x) \nu(dx)$  et  $Z_g = \int_{x \in \Omega} g(x) \nu(dx)$ , les constantes de normalisation pour les densités (possiblement) non-normalisées  $f$  et  $g$ . Alors, pour la méthode par rejet présentée précédemment, la probabilité d'acceptation est  $Z_f/[c * Z_g]$  et le nombre moyen de lancers est  $c * Z_g/Z_f$ .

*Démonstration.* Pour  $C$ , la loi de Bernoulli présentée précédemment, le problème est de calculer  $\mathbb{P}(C = 1)$ .

$$\begin{aligned} \mathbb{P}(C = 1) &= \mathbb{E}[\mathbb{E}[\mathbf{1}(C = 1)|X]] \\ &= \int_{x \in \Omega} \mathbb{P}(C = 1|X = x) \mathbb{P}(X \in dx) \\ &= \int_{x \in \Omega} f(x)/[cg(x)] (g(x)/Z_g) \nu(dx) \\ &= [cZ_g]^{-1} \int_{x \in \Omega} f(x) \nu(dx) \\ &= Z_f/[c * Z_g] \end{aligned}$$

□

### 2.2.1 Exemple : D'une Cauchy vers une Normale

Un exemple simple est la situation suivante : on suppose qu'un utilisateur peut générer une réalisation d'une loi de Cauchy standard de densité  $g(x) = [\pi(1 + x^2)]^{-1}$ . Cet utilisateur souhaite générer une réalisation d'une loi Normale standard de densité  $f(x) = \frac{1}{\sqrt{2\pi}} \exp(-x^2/2)$ . D'abord, la normalisation n'est pas nécessaire, donc soit  $g_1(x) = (1 + x^2)^{-1}$  et  $f_1(x) = \exp(-x^2/2)$ . On a que  $f_1(x)/g_1(x) \leq 2, \forall x$ . On implémente alors sous R l'algorithme :

## Mise en place de la méthode de rejet sous R : d'une Cauchy vers une Normale

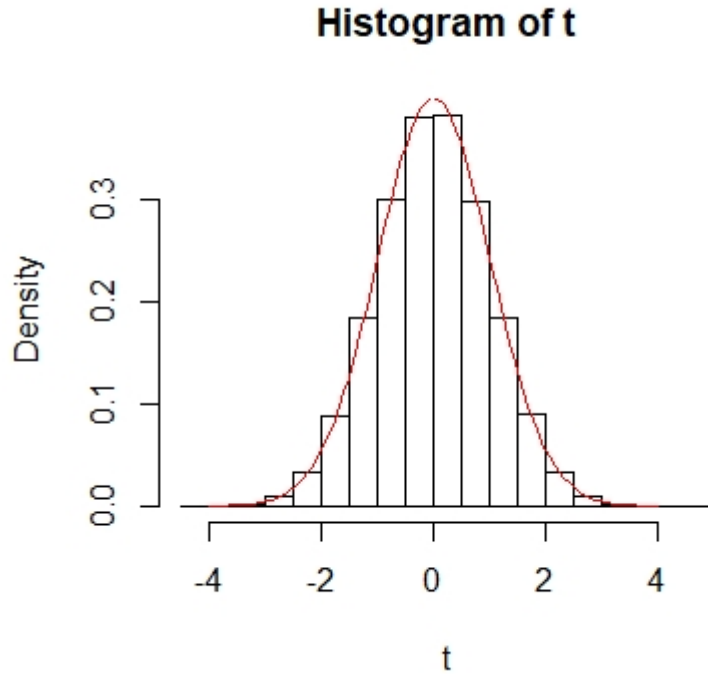
```
1  normalfromcauchy <- function(){## Fonction permettant de générer une
   réalisation de la loi normale standard à partir de la loi de
   Cauchy standard par méthode de rejet ##
2
3  ##penser à changer le return pour obtenir une réalisation (return(x)
   ) ou le temps mis à l'obtenir (return(n))##
4
5  ##On initialise les valeurs ##
6  C=0
7  x=0
8  n=0
9
10  ##Tant que la bernouilli n'est pas 1##
11  while(C!=1){
12
13    ##On génère une Cauchy standard##
14    x = rcauchy(1)
15
16    ##On génère une Bernouilli du paramètre indiqué##
17    C = rbinom(1,1,((sqrt(exp(1))/2)*(1+x^2)*exp(-(1/2)*x^2)))
18
19    ##mise à jour du nombre de lancers##
20    n=n+1
21  }
22  return(x)
23 }
24
25
26
27  ##On calcule le nombre moyen de lancers nécessaires pour atteindre
   la sortie de l'algorithme précédent##
28
29  ##On initialise le nombre moyen##
30  s=0
31
32  ##On lance 10^6 fois l'algorithme qui retourne le nombre de lancers
   effectué ##
33  for (i in 1:10^6) {
34    s=s+normalfromcauchy()
35  }
36
37  ##On moyenne sur les 10^6 lancers
38  s = s/10^6
39
40
41
```

```

42  ##On créé un histogramme illustrant  $10^5$  lancers de l'algorithme
    rendant une réalisation de la normale standard##
43
44  ##On initialise le tableau qui contiendra les  $10^5$  lancers
45  t = matrix(0,1, $10^5$ )
46
47  ##On remplit le tableau en lançant  $10^5$  fois l'algorithme##
48  for (i in 1: $10^5$ ) { #changer le return pour obtenir des réalisations
49    t[i]=normalfromcauchy()
50  }
51
52
53  ##On produit l'histogramme du tableau t et on rajoute la courbe de
    la densité de la loi normale standard##
54  hist(t,prob=T)
55
56  curve(dnorm(x, 0, 1), col="red",xlim=c(-4,4),add=T)

```

L'histogramme rendu après un lancer du programme est le suivant :



On obtient bien l'allure de la densité d'une loi normale standard.

On récupère aussi le nombre moyen de lancers requis afin d'obtenir une réalisation : 2.50588. On peut retrouver ce résultat de manière théorique en calculant la probabilité d'acceptation :

$$\begin{aligned}\mathbb{P}(C = 1) &= \int_{x \in \mathbb{R}} \mathbb{P}(X \in dx, C = 1) \\ &= \int_{x \in \mathbb{R}} g(x) \frac{1}{2} (1 + x^2) \exp(-x^2/2) dx \\ &= (2\pi)^{-1/2}\end{aligned}$$

Le nombre moyen de lancers est donc  $\sqrt{2\pi} = 2.506\dots$

Il est possible d'accélérer l'algorithme : on a en fait  $\sup_x (1+x^2) \exp(-x^2/2) = 2/\sqrt{e}$ . En remplaçant dans l'algorithme (mais aussi en calculant exactement l'intégrale), le nombre moyen de lancers devient  $\sqrt{2\pi/e} = 1.520\dots$

### 2.3 Union d'ensembles

Une autre manière d'utiliser les méthodes de rejet est de prendre en compte les multiples manières d'obtenir une réalisation. Par exemple, on considère le problème de génération selon une mesure  $\nu$  sur  $S_1 \cup \dots \cup S_n$ , où

les ensembles  $S_i$  sont de mesure finie  $\nu(S_i)$ .

On considère l'algorithme suivant : on tire  $I$  dans  $\{1, 2, \dots, n\}$  de telle sorte que  $\mathbb{P}(I = i) \propto \nu(S_i)$ . Puis, conditionnellement sur  $I$ , on tire  $X$  dans  $S_I$  selon la mesure  $\nu$ .

Le problème est que  $X$  n'est pas tiré selon  $\nu$  sur  $S_1 \cup \dots \cup S_n$ , car il y a plusieurs manières pour lesquelles  $X$  aurait pu être choisi. Par exemple, si  $X \in S_1$  et  $X \in S_2$  mais  $X \notin S_3 \dots S_n$ , alors  $I$  aurait pu être 1 ou 2 lors du choix de  $X$ .

**Lemme 3.** *Pour la procédure précédente,  $X$  est un tirage selon  $\nu$  sur  $S_1 \cup \dots \cup S_n$  avec densité  $f(x) = \#\{i : x \in S_i\} / \nu(S_1 \cup \dots \cup S_n)$*

*Démonstration.* Soit  $x \in S_1 \cup \dots \cup S_n$  et  $C = \nu(S_1 \cup \dots \cup S_n)$  Alors :

$$\begin{aligned} \mathbb{P}(X \in dx) &= \sum_{i=1}^n \mathbb{P}(X \in dx, I = i) \\ &= \sum_{i=1}^n \mathbb{P}(I = i) \mathbb{P}(X \in dx | I = i) \\ &= \sum_{i=1}^n C^{-1} \nu(S_i) \mathbf{1}(x \in S_i) \nu(dx) \nu(S_i)^{-1} \\ &= C^{-1} \#\{i : x \in S_i\} \nu(dx) \end{aligned}$$

□

Par exemple, supposons que l'on veuille tirer uniformément sur les 3 cercles unités respectivement centrés en  $(0, 0)$ ,  $(1/2, 0)$  et  $(1/2, 1/2)$ .

On tire alors  $I \sim \text{Unif}(\{1, 2, 3\})$ . Si  $I = 1$ , on tire  $X$  sur le cercle unité centré en  $(0, 0)$ . Pour  $I = 2$  ou  $I = 3$ , on tire  $X$  sur le cercle centré en  $(1/2, 0)$  et  $(1/2, 1/2)$  respectivement. Après, avoir tiré  $I$  puis  $X$  conditionnellement à  $I$ , on accepte  $X$  avec probabilité égale à l'inverse du nombre de cercles où se situe  $X$ .

## 2.4 Simulation à l'aide des inégalités de Markov et Chernoff

### 2.4.1 L'inégalité de Markov en tant que procédure de rejet

**Lemme 4** (Inégalité de Markov). *On considère une variable aléatoire  $X$  non-négative avec probabilité 1 et d'espérance finie  $\mathbb{E}[X]$ . Alors, pour tout  $a > 0$  :*

$$\mathbb{P}(X \geq a) \leq \frac{\mathbb{E}[X]}{a}$$

Du point de vue simulation, l'inégalité de Markov nous permet de tirer depuis  $[X|X \geq a]$ . C'est à dire, si  $X$  a pour densité  $f_X(x)$  alors le but est de tirer depuis la densité non-normalisée  $f_X(x)\mathbf{1}(x \geq a)$ . Pour pouvoir appliquer une méthode de rejet, on utilise la densité non-normalisée  $xf_X(x)$ . On a alors que  $f_X(x)\mathbf{1}(x \geq a)/[xf_X(x)]$  vaut 0 lorsque  $x < a$ , et  $1/x$  lorsque  $x \geq a$ . Le produit n'est donc jamais supérieur à  $1/a$  ce qui permet d'utiliser la méthode de rejet suivante :

---

**Méthode rejet Inégalité de Markov** Entrées :  $f, a$  Sortie :  $X \sim f$  sachant  $X > a$

---

```

while  $C \neq 1$  do
  Tirer  $X$  selon la densité non-normalisée  $xf(x)$ 
  Tirer  $C \sim \text{Bern}(a\mathbf{1}(X \geq a)/x)$ 
end while

```

---

On remarque que la constante de normalisation pour la densité  $xf(x)$  est simplement :  $\int xf(x)\nu(dx) = \mathbb{E}[X]$ . On a alors :

$$\mathbb{P}(C = 1) = \int_{x \geq 0} \frac{xf_X(x)}{\mathbb{E}[X]} \frac{a\mathbf{1}(X \geq a)}{x} = \int_{x \geq a} \frac{af_X(x)}{\mathbb{E}[X]} = \frac{a\mathbb{P}(X \geq a)}{\mathbb{E}[X]}$$

Puisque la chance d'accepter est au plus 1, cette fraction est aussi au plus 1, ce qui montre l'inégalité de Markov.

#### 2.4.2 Les inégalités de Chernoff en tant que procédure de rejet

Les inégalités de Chernoff donnent des limites supérieures sur la probabilité qu'une somme de variables aléatoires est supérieure (ou inférieure) à une certaine valeur. Soit  $S_n = X_1 + \dots + X_n$ , où  $X_1, \dots, X_n$  sont iid ; alors le but est de générer un tirage selon  $S_n$  tel que  $S_n \geq a$  ou  $S_n \leq a$ . L'inégalité de Markov peut aussi être appliquée à une somme de variables aléatoires, mais la borne donnée n'est pas aussi précise que celle obtenue avec les inégalités de Chernoff.

**Lemme 5** (Inégalités de Chernoff). *Soit une variable aléatoire  $X$  telle que  $\mathbb{E}[e^{tX}]$  soit finie pour  $t \in [a, b]$ , où  $a$  est négatif et  $b$  positif. Alors pour un certain  $c \in \mathbb{R}$ , et en notant par  $\text{mgf}_X(t)$  la fonction génératrice des moments pour la variable aléatoire  $X$ ,*

$$\begin{aligned} \mathbb{P}(X \geq c) &\leq \text{mgf}_X(t) / \exp(tc), \quad \forall t \in [0, b] \\ \mathbb{P}(X \leq c) &\leq \text{mgf}_X(t) / \exp(tc), \quad \forall t \in [a, 0] \end{aligned}$$



*Démonstration.* Tout d'abord :  $\mathbb{P}(X \geq c) = \mathbb{P}(tX \geq tc)$  pour tout  $t$  positif. De plus,  $\mathbb{P}(tX \geq tc) = \mathbb{P}(\exp(tX) \geq \exp(tc))$ . Enfin, on applique l'inégalité de Markov pour obtenir  $\mathbb{P}(\exp(tX) \geq \exp(tc)) \leq \mathbb{E}[\exp(tX)] / \exp(tc)$ . Le résultat pour l'autre inégalité est démontré de la même manière.  $\square$

Nous considérons à présent ce qu'il se passe lorsque l'on considère les inégalités de Chernoff pour une somme de variables aléatoires indépendante. On rappelle d'abord un résultat sur les fonctions génératrices des moments.

**Lemme 6.** *Soit  $S_n = X_1 + \dots + X_n$ , où les  $\{X_i\}$  sont iid de fonction génératrice des moments finie. Alors on a :  $\mathbb{E}[\exp(tS_n)] = [\mathbb{E}[\exp(tX_i)]]^n$*

On applique ce lemme aux inégalités de Chernoff pour obtenir :

**Lemme 7.** *Soit  $S_n = X_1 + \dots + X_n$ , où les  $\{X_i\}$  sont iid de fonction génératrice des moments  $mgf_{X_i}(t)$  finie pour  $t \in [a, b]$ , où  $a$  est négatif et  $b$  positif. Alors on a :*

$$\begin{aligned}\mathbb{P}(S_n \geq \alpha n) &\leq \left( \frac{mgf_{X_i}(t)}{\exp(t\alpha)} \right)^n, \quad \forall t \in [0, b] \\ \mathbb{P}(S_n \leq \alpha n) &\leq \left( \frac{mgf_{X_i}(t)}{\exp(t\alpha)} \right)^n, \quad \forall t \in [a, 0]\end{aligned}$$

Supposons que  $X_i \sim f_X$ . Le but est d'utiliser la fonction génératrice des moments pour obtenir un meilleur algorithme de rejet. Pour cela, il doit être possible de générer une réalisation selon la densité  $g_t(x) \propto e^{tx} f_x(x)$ . Lorsque  $t$  est grand, cette densité aura tendance à prendre de plus grandes valeurs. Lorsque  $t$  est grand dans les négatifs,  $g_t$  tendra à prendre des valeurs plus faibles.

Soit  $t > 0$ . Pour  $x \geq a$ , alors  $g_t(x) = e^{tx} f_X(x) \geq e^{ta} f_X(x)$ . Alors une réalisation de  $g_t$  peut être acceptée comme réalisation de  $f_X$  avec probabilité  $e^{ta}/e^{tx}$ . Si la probabilité que  $x$  soit bien plus grand que  $a$  est faible, alors la probabilité d'acceptation sera très proche de 1. Une méthode similaire s'applique lorsque  $t < 0$ . On en déduit alors l'algorithme suivant :

---

**Méthode rejet Inégalités de Chernoff**

---

Entrées :  $f_X, a, t$

Sortie :  $S_n = X_1 + \dots + X_n$  sachant  $S_n \geq a$  (lorsque  $t > 0$ ) ou sachant  $S_n \leq a$  (lorsque  $t < 0$ )

---

Si  $t > 0$ , alors  $A \leftarrow [a, \infty)$ , sinon  $(-\infty, a]$

**while**  $C \neq 1$  **do**

    Tirer  $X_1, \dots, X_n$  iid selon la densité non-normalisée  $e^{tx}f(x)$

$S_n \leftarrow X_1 + \dots + X_n$

    Tirer  $C \sim \text{Bern}(\exp(t(a - S_n))\mathbf{1}(S_n \in A))$

**end while**

---

**Lemme 8.** *Supposons que  $\text{mgf}_{X_i}(t) \exp(-ta/n) < 1$ . Alors l'algorithme précédent génère une réalisation de  $[S_n | S_n > a]$  lorsque  $t > 0$  ou de  $[S_n | S_n < a]$  lorsque  $t < 0$ .*

*Démonstration.* On considère tout vecteur  $(x_1, \dots, x_n)$  tel que  $\sum x_i = s$ . On considère  $X_1, \dots, X_n$  iid de densité  $f_X(x)$  et  $X'_1, \dots, X'_n$  iid de densité  $\exp(tx)f_X(x)$ . Alors la densité de  $S'_n = X'_1 + \dots + X'_n$  est simplement la densité de  $S_n = X_1 + \dots + X_n$  avec un facteur  $\exp(tx_1)\exp(tx_2)\dots\exp(tx_n) = \exp(ts)$ .

On génère dans l'algorithme une variable aléatoire de densité  $g(s) = \exp(ts)f_{S_n}(s)$  et la densité ciblée est  $f(s) = f_{S_n}(s)\mathbf{1}(s \in A)$ . On a alors  $f(s)/g(s)$  qui est soit 0 lorsque  $s \notin A$  soit  $\exp(-ts)$  lorsque  $s \in A$ .

Si  $A = [a, \infty)$ , alors  $t > 0$  et  $\exp(-ts) \leq \exp(-ta)$ . De même, lorsque  $A = (-\infty, a]$ , alors  $t < 0$  et  $\exp(-ts) \leq \exp(-ta)$ . On choisit donc  $c = \exp(-ta)$  et on obtient  $f(s)/[cg(s)] = \exp(ta)\exp(-ts)\mathbf{1}(s \in A)$  de même que dans l'algorithme.  $\square$

## Mise en place de la méthode de rejet pour R :

### Inégalités de Chernoff et queue de loi binomiale

```
1  AR_chernoffs <- function(n,p,a){ ##fonction permettant de générer
    une réalisation de la loi binomiale de paramètres n et p,
    conditionnellement à être supérieure ou égale à a en utilisant
    les inégalités de Chernoff##
2
3
4  ##On initialise la Bernouilli##
5  C=0
6
7  ##On crée la valeur gamma présentée précédemment##
8  gam = (a/n)*(1-p)/(p*(1-(a/n)))
9
10 ##On initialise la Binomiale de paramètres n et p
11 t = 0
12
13 ##On initialise le nombre de lancers##
14 k = 0
15
16 ##Tant que la Bernouilli n'est pas 1##
17 while (C != 1){
18
19     ##On génère n Bernouillis de paramètres a/n##
20     t = rbinom(n,1,a/n)
21
22     ##On effectue la somme pour obtenir la binomiale considérée##
23     x = sum(t)
24
25     ##On génère une réalisation de la Bernouilli du paramètre
        indiqué
26     C = rbinom(1,1,(x>=a)*1*gam**((a-x)))
27
28     ##On met à jour le nombre de lancers effectués##
29     k=k+1
30 }
31
32 ##On rend soit le nombre de lancers, soit la réalisation de la loi
        demandée (ie, k ou x)##
33 return(k)
34
35 }
36
37
38 ##On calcule ici le nombre moyen de lancers nécessaires pour sortir
        de l'algorithme précédent
```

```

39
40 s2=0
41 for (i in 1:10^5) { #changer le return pour obtenir le nombre moyen
    de lancers nécessaires
42     s2=s2 + AR_chernoffs(10,0.1,5)
43 }
44 s2 = s2/10^5

```

```

1  AR_basic <- function(n,p,a){##fonction permettant de générer une
    réalisation de la loi binomiale de paramètres n et p,
    conditionnellement à être supérieure ou égale à a en utilisant
    la méthode de rejet basique##
2
3  ##On initialise le critère d'arrêt##
4  C=0
5
6  ##On initialise la Binomiale de paramètres n et p##
7  t = 0
8
9  ##On initialise le nombre de lancers##
10 k = 0
11
12 ##Tant que le critère d'arrêt n'est pas 1##
13 while (C != 1){
14
15     ##On génère une réalisation de la Binomiale de paramètres n et p
        ##
16     t = rbinom(1,n,p)
17
18     ##On met à jour le critère d'arrêt selon que la binomiale soit
        supérieure à a ou non##
19     C = (t>=a)*1
20
21     ##On met à jour le nombre de lancers##
22     k=k+1
23 }
24
25 ##On rend soit le nombre de lancers, soit la réalisation de la loi
    demandée (ie, k ou t))##
26 return(k)
27
28 }
29
30
31 ##On calcule ici le nombre moyen de lancers nécessaires pour sortir
    de l'algorithme précédent
32 s1=0
33 for (i in 1:10^6) { #changer le return pour obtenir le nombre moyen
    de lancers nécessaires
34     s1=s1+ AR_basic(10,0.1,5)
35 }
36 s1= s1/10^6

```

On utilise l'exemple suivant pour tester nos algorithmes : pour  $n = 10$  et  $p = 0.1$ , pour  $X$  la binomiale de paramètres précisés précédemment, on veut générer une réalisation de  $[X|X \geq 5]$ . La méthode de base nous donne un nombre moyen de lancers de 610.9838 tandis que la méthode à l'aide des inégalités de Chernoff nous donne un nombre moyen de lancers de 3.698136. D'où l'efficacité de cette méthode.

## 2.5 Défaut des méthodes de rejet

Le principal défaut des méthodes de rejet concerne l'approche des problèmes considérés. On prend pour exemple la génération de la variable aléatoire uniforme dans la boule de dimension  $n$ .

La méthode présentée précédemment dans le cas  $n = 2$  nécessite de tirer uniformément sur le carré  $[-1, 1] \times [-1, 1]$ . La probabilité d'acceptation  $p$  est alors l'aire du cercle sur l'aire du carré, ie,  $p = \pi/4$ .

On peut généraliser cette méthode aux dimensions supérieures : on tire uniformément dans l'hypercube unité de dimension  $n$  :  $[-1, 1]^n$ , le but étant d'obtenir un tirage uniforme dans la boule unité de même dimension.

**Théorème 9** (Volume de la boule unité). *Le volume de la boule unité en dimension  $n$  est  $\frac{\pi^{\frac{n}{2}}}{\Gamma(\frac{n}{2}+1)}$ , où  $\Gamma$  désigne la fonction Gamma.*

*Démonstration.* Notons par  $V^{(n)}[1]$  le volume la boule unité de dimension  $n$  et de rayon 1. On a d'abord  $V^{(1)}[1] = 2$ , puis pour tout  $n \geq 1$ , on a par récurrence, en utilisant la relation suivante : pour  $B_d(0, 1)$  la boule unité de dimension  $d$ ,

$$B_d(0, 1) = \left\{ (x_1, \dots, x_{d-1}, h) \mid h \in [-1, 1], (x_1, \dots, x_{d-1}) \in B_{d-1}(0, \sqrt{1-h^2}) \right\}$$

Et en utilisant le théorème de Fubini,

$$V^{(n+1)}[1] = \int_{-1}^1 V^{(n)}[1] \left( \sqrt{1-x^2} \right)^n dx = V^{(n)}[1] * 2 \int_0^1 (1-x^2)^{n/2} dx$$

On effectue ensuite le changement de variables  $u = x^2$  ce qui nous donne  $x = \sqrt{u}$  et  $dx = \frac{du}{2\sqrt{u}}$  pour obtenir :

$$V^{(n+1)}[1] = V^{(n)}[1] * 2 \int_0^1 (1-x^2)^{n/2} dx = V^{(n)}[1] \int_0^1 (1-u)^{n/2} u^{-1/2} du$$

L'intégrale à droite est connue comme la fonction bêta, d'où :

$$V^{(n+1)}(1) = V^{(n)}[1]B\left(\frac{n}{2} + 1, \frac{1}{2}\right)$$

Or on peut exprimer la fonction bêta par rapport à la fonction gamma pour obtenir :

$$V^{(n+1)}(1) = V^{(n)}[1] \frac{\Gamma(\frac{n}{2} + 1)\Gamma(\frac{1}{2})}{\Gamma(\frac{n}{2} + \frac{3}{2})}$$

Enfin, en utilisant le fait que  $\Gamma(\frac{1}{2}) = \sqrt{\pi}$ , et à l'aide d'une simple récurrence, on a enfin que :

$$V^{(n)}[1] = \frac{\pi^{\frac{n}{2}}}{\Gamma(\frac{n}{2} + 1)}$$

□

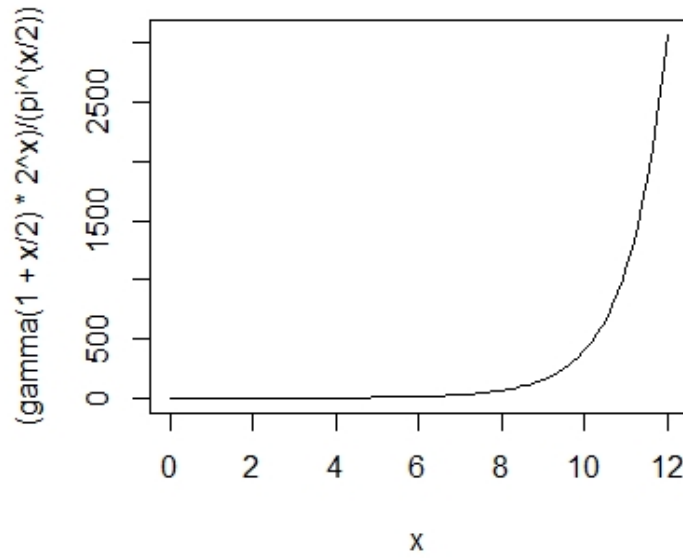
La probabilité d'acceptation  $p$  d'un tirage aléatoire dans l'hypercube unité de dimension  $n$  en tant que tirage dans la boule unité de même dimension est donc le volume de la boule sur le volume de l'hypercube, soit encore :

$$p = \frac{V^{(n)}[1]}{2^n} = \frac{\pi^{\frac{n}{2}}}{2^n \Gamma(\frac{n}{2} + 1)}$$

On en déduit le nombre moyen  $k$  de lancers nécessaires à l'acceptation (par simple étude de loi géométrique) :

$$k(n) = \frac{1}{p} = \frac{2^n \Gamma(\frac{n}{2} + 1)}{\pi^{\frac{n}{2}}}$$

On trace ci-dessous  $k$  en fonction de  $n$  pour  $n \leq 12$  :



On remarque une augmentation très rapide du nombre de lancers nécessaires à l'obtention d'une réalisation selon que la dimension croît. La méthode de rejet n'est donc plus efficace pour des problèmes dont l'approche n'est pas connue, une étude approfondie étant nécessaire.



### 3 Chapitre 3 : Coupling From The Past

#### Définition : Fonction de mise à jour

On dit que  $\phi : \Omega \times [0, 1] \rightarrow \Omega$  est une fonction de mise à jour pour une chaîne de Markov  $\{X_t\}$  si, pour  $U \sim \text{Unif}[0, 1]$ ,  $[X_{t+1}|X_t] \sim \phi(X_t, U)$ .

La fonction  $\phi$  est déterministe : tout l'aléatoire est contenu dans la variable  $U$ .

Toute chaîne qui peut-être simulée sur ordinateur est un exemple de fonction de mise à jour.

Une même chaîne de Markov peut-être représentée par plusieurs fonctions de mise à jour : la fonction de mise à jour n'est pas forcément unique.

À l'aide d'une fonction de mise à jour  $\phi$ , on peut représenter la trajectoire d'une chaîne de Markov  $\{X_t\}$ . En effet, soient  $U_0, U_1, U_2, \dots$  iid  $\sim \text{Unif}([0, 1])$ . Pour un état initial  $x_0$ , on a :  $X_1 = \phi(x_0, U_0)$  puis pour  $i > 1$ ,  $X_i = \phi(X_{i-1}, U_{i-1})$ .

On notera alors la trajectoire jusqu'au temps  $t$  sous la forme :

$$\phi_t(x_0, U) = \phi(\phi(\phi(\dots(\phi(x_0, U_0), U_1), \dots, U_{t-1}))$$

Ensuite, pour n'importe quels états  $x_0$  et  $y_0$  dans  $\Omega$ , on définit pour une fonction de mise à jour  $\phi : X_t = \phi_t(x_0, U)$  et  $Y_t = \phi_t(y_0, U)$  (en utilisant les mêmes valeurs de  $U$ ). On appelle ce procédé un couplage. Notons qu'avec ce couplage, si il existe  $t \geq 0$  tel que  $X_t = Y_t$ , alors on dit que les processus ont fusionné (ou se sont rejoints, etc).

#### Définition : Couplage

Soit  $\mathcal{S}$  un ensemble de processus stochastiques définis sur un même ensemble d'indices  $\mathcal{I}$  et un même espace d'états  $\Omega$ . Si il existe un indice  $i \in \mathcal{I}$  et un état  $x \in \Omega$  tels que pour tout  $S \in \mathcal{S}$ , on aie  $S_i = x$ , alors on dit que les processus stochastiques ont coalescé (ou se sont rejoints, couplés, etc).

#### 3.1 CFTP : approche trajectorielle (Baccelli / Brémaud)

Soit  $P$  une matrice de transition ergodique sur l'espace d'état fini  $E = \{1, \dots, r\}$ , de distribution stationnaire. Comme définie précédemment, à l'aide d'une fonction de mise à jour  $h$ , on peut implémenter la chaîne de Markov de la manière suivante :

$$X_{n+1} = h(X_n, \xi_n)$$

pour une suite de variables aléatoires  $\{\xi_n\}_{n \geq 1}$  iid uniformes sur  $[0, 1]$  et indépendantes de l'état initial.

On considère à présent un tableau de choix aléatoires  $\{\xi_k(i)\}_{k \in \mathbb{Z}, i \in E}$  iid

uniformes sur  $[0, 1]$ . Pour tout  $k \in \mathbb{Z}$  et tout  $i \in E$ , soit  $\{X_n^k(i)\}_{n \geq k}$  définie par récurrence :

$$X_{n+1}^k(i) = h(X_n^k(i), \xi_n(X_n^k(i))), \quad n \geq k,$$

avec pour condition initiale  $X_k^k(i) = i$ , et  $h$  comme définie précédemment. Pour tout  $k$  et  $i$ ,  $\{X_n^k(i)\}_{n \geq k}$  est une chaîne de Markov homogène de matrice de transition  $P$ . Par la structure de récurrence stochastique sous-jacente, les chaînes de la famille définie précédemment sont telles que : pour tout  $k \in \mathbb{Z}$ ,  $X_n^k(i) = X_n^k(j)$  implique que  $X_m^k(i) = X_m^k(j)$ , pour tout  $m \geq n$ . On note par :

$$N^+ = \inf \{n \geq 0 | X_n^0(1) = X_n^0(2) = \dots = X_n^0(r)\}$$

( $= +\infty$  si la condition n'est jamais satisfaite) le temps de coalescence forwards de la chaîne. On notera le temps de coalescence backwards de la chaîne par :

$$N^- = \inf \{n \geq 1 | X_0^{-n}(1) = X_0^{-n}(2) = \dots = X_0^{-n}(r)\}$$

( $= +\infty$  si la condition n'est jamais satisfaite)

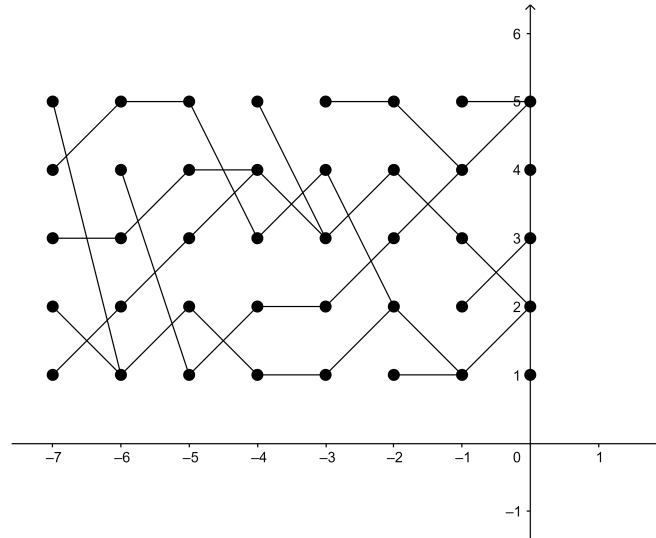


FIGURE 1 – Exemple de temps de coalescence backwards,  $N^- = 7$

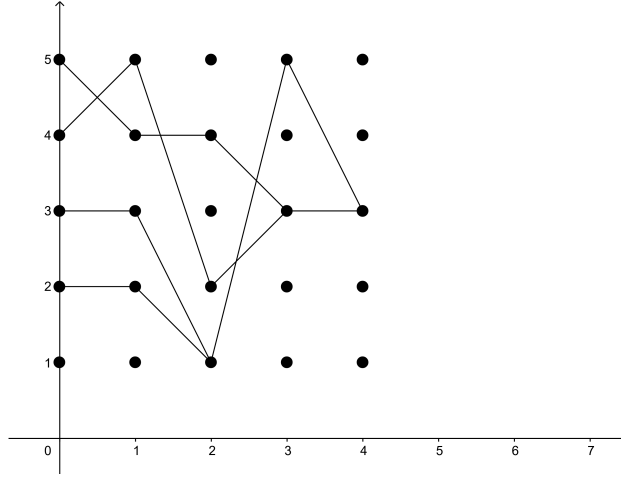


FIGURE 2 – Exemple de temps de coalescence forwards,  $N^+ = 4$

**Théorème 10.** *Le temps de coalescence forwards  $N^+$  est presque sûrement fini.*

*Démonstration.* Il suffit de prouver le résultat dans le cas de  $r$  chaînes de Markov homogènes, complètement indépendantes les unes des autres, de même matrice de transition. Nous n'avons pas l'hypothèse d'indépendance dans la construction des chaînes de Markov donnée précédemment. Cependant, la probabilité de coalescence (probabilité que  $N^+$  soit fini) dans notre situation est bornée inférieurement par la probabilité de coalescence dans le cas complètement indépendant. Pour mieux le comprendre, on construit d'abord le modèle de chaînes indépendantes :

$$\overline{X}_{n+1}(i) = h(\overline{X}_{n+1}(i), \bar{\xi}_{n,i}), \quad n \geq 0,$$

(avec pour condition initiale  $\overline{X}_0(i) = i$ ), qui utilise  $r$  composantes de mise à jour iid  $\{\bar{\xi}_{n,i}\}$ .

La différence entre ce modèle et celui que l'on a introduit réside dans le nombre de mise à jour trop élevé de notre modèle. Afin de construire un ensemble de  $r$  chaînes semblable à celui de notre modèle, il suffit d'utiliser les mêmes mise à jour pour deux chaînes dès lors qu'elles se rencontrent. Il est alors clair que le temps de coalescence forwards du modèle ainsi modifié est plus petit ou égal à celui du modèle complètement indépendant.

Il reste alors à prouver que pour un nombre fini de chaînes de Markov homogènes, ergodiques et indépendantes, elles finiront par se rencontrer. Cela suit du fait que le produit de chaînes ergodiques indépendantes est une chaîne ergodique.

□

**Théorème 11.** *Les variables aléatoires  $N^+$  et  $N^-$  ont la même distribution.*

*Démonstration.* Soit  $k \in \mathbb{N}$ . On considère le modèle modifié obtenu en remplaçant  $\xi_{-k+l}(i)$  par  $\xi_l(i)$ , pour tout  $l$  tel que  $0 \leq l \leq k$ , et pour  $i \in E$ . Notons par  $N'$  le temps de coalescence backwards du modèle modifié. Clairement  $N^-$  et  $N'$  ont la même distribution.

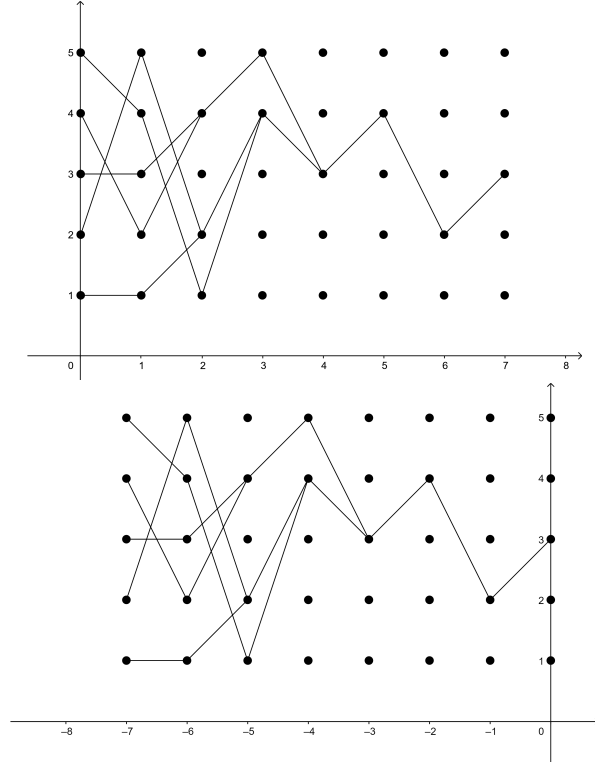


FIGURE 3 –  $N^+ \leq k$  implique  $N' \leq k$

On suppose maintenant que  $N^+ \leq k$ . Alors, dans le modèle modifié, les chaînes commençant au temps  $-k$  depuis les états  $1, \dots, r$  coalescent au temps  $-k + N^+ \leq 0$  (voir figure précédente), et par conséquent  $N' \leq k$ . Donc,  $N^+ \leq k$  implique  $N' \leq k$ , puis :

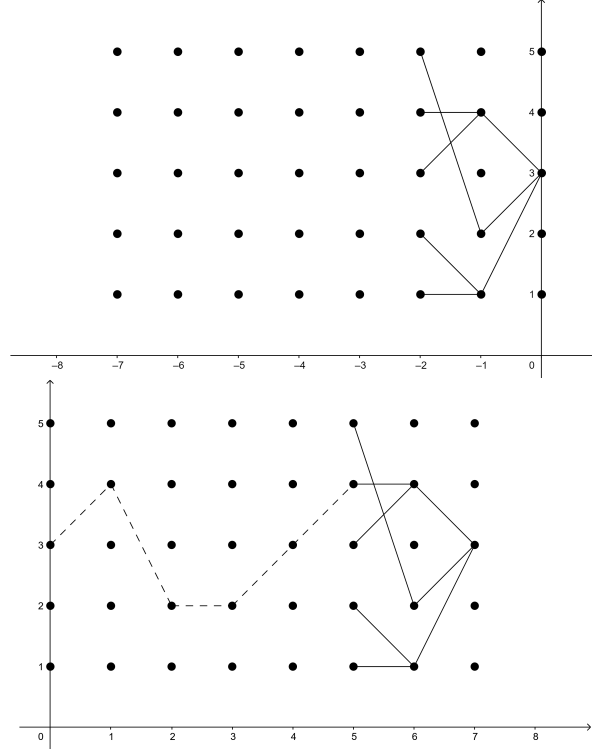
$$\mathbb{P}(N^+ \leq k) \leq \mathbb{P}(N' \leq k) = \mathbb{P}(N^- \leq k)$$

D'autre part, on suppose que  $N' \leq k$ . Alors, dans le modèle original, les chaînes commençant depuis les états  $1, \dots, r$  au temps  $k - N'$  coalesceront au temps  $k$ . On en déduit donc  $N^+ \leq k$  (voir figure suivante). On a alors que  $N' \leq k$  implique  $N^+ \leq k$ , puis :

$$\mathbb{P}(N^- \leq k) = \mathbb{P}(N' \leq k) \leq \mathbb{P}(N^+ \leq k)$$

Grâce aux deux inégalités démontrées précédemment, on en déduit le résultat.

□

FIGURE 4 –  $N' \leq k$  implique  $N^+ \leq k$ 

On définit à présent la variable aléatoire suivante :

$$Z = Z(i) = X_0^{-N^-}(i)$$

(la variable aléatoire  $Z$  est indépendante de  $i \in E$  ; dans la figure 1, on a  $Z = 2$ ).

On a le théorème suivant :

**Théorème 12.** *Le temps de coalescence backwards  $N^-$  est presque sûrement fini. De plus, la variable aléatoire  $Z$  admet  $\pi$  comme distribution.*

*Démonstration.* Puisque, pour tout  $k \in \mathbb{N}$ ,  $\mathbb{P}(N^- \leq k) = \mathbb{P}(N^+ \leq k)$  (d'après le résultat précédent), le fait que  $N^-$  soit fini provient directement du fait que  $N^+$  soit fini (théorème 10).

D'autre part, puisque pour  $n \geq N^-$ , on a  $X_0^{-n}(i) = Z$ ,

$$\mathbb{P}(Z = j) = \lim_{n \rightarrow \infty} \mathbb{P}(X_0^{-n}(i) = j) = \lim_{n \rightarrow \infty} p_{ij}(n) = \pi(j),$$

où la dernière égalité provient du théorème ergodique pour les chaînes de Markov. □

**Exemple : distribution stationnaire et temps de coalescence forwards**

Le théorème précédent montre qu'en utilisant le temps de coalescence backwards, on obtient une réalisation de la loi stationnaire par la méthode CFTP. La question est alors la suivante : pourquoi ne pas utiliser le temps de coalescence forwards ? La réponse à cette question s'obtient en étudiant l'exemple simple suivant :

On considère  $\{X_n\}$  la chaîne de Markov à deux états de matrice de transition  $P = \begin{pmatrix} 0 & 1 \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix}$ .

On remarque que cette chaîne a pour loi stationnaire  $\pi = (\frac{1}{3} \ \frac{2}{3})$ .

On remarque ensuite que, pour  $N^+$  le temps de coalescence forwards, on a  $X_0^{N^+}(1) = X_0^{N^+}(2) = 2$  presque sûrement. En effet, par l'absurde, si les deux chaînes issues des états 1 et 2 s'étaient rejointes au temps  $N^+$  à l'état 1, alors on aurait deux cas possibles :

- Au temps  $N^+ - 1$ , la chaîne issue de l'état 1 se déplace à l'état 1 au temps suivant : une absurdité par la structure de la chaîne ( $P_{1,1} = 0$ )
- Au temps  $N^+ - 1$ , les deux chaînes ont déjà coalescé, d'où la coalescence au temps  $N^+$  : une absurdité par la définition de  $N^+$

Cependant le temps de coalescence forwards n'est pas une cause perdue : certains modèles permettent d'utiliser la méthode CFTP en conjugaison avec celui-ci pour obtenir une réalisation de la loi stationnaire, comme le montre l'exemple suivant :

On considère  $\{X_n\}$  la chaîne de Markov à deux états de matrice de transition  $P = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix}$ .

On remarque facilement que la loi stationnaire est  $\pi = (\frac{1}{2} \ \frac{1}{2})$ , puis que, au temps  $N^+$  :

$$\mathbb{P}(X_0^{N^+}(1) = X_0^{N^+}(2) = 1) = \mathbb{P}(X_0^{N^+}(1) = X_0^{N^+}(2) = 2) = \frac{1}{2}$$

car les deux événements sont équiprobables et que l'on sait que l'on coalesce forcément pour la première fois au temps  $N^+$ .

La méthode présentée précédemment est particulièrement coûteuse lorsque l'espace d'états est grand (puisque'il faut faire évoluer des trajectoires à partir de chaque état). Cependant, si la méthode ne requerrait la coalescence que de 2 chaînes et non plus  $r$ , celle-ci serait bien plus efficace. Propp et

Wilson ont montré comment utiliser cela dans le cas monotone suivant.  
On suppose qu'il existe une relation d'ordre partiel sur l'espace d'état  $E$  que l'on notera par  $\preceq$ .  
On suppose ensuite que la fonction de mise à jour préserve cette relation d'ordre, ie,

$$i \preceq j \Rightarrow h(i, \xi) \preceq h(j, \xi) \quad \forall \xi$$

On suppose ensuite que  $1 \preceq 2 \preceq \dots \preceq r$  et on considère le modèle avec une seule mise à jour commune (ie,  $\xi_n(i) = \xi_n$  pour tout  $n$  et tout  $i$ ). On définit à présent le temps de coalescence backwards monotone :

$$M = \inf \left\{ n \geq 1 \mid X_0^{-n}(1) = X_0^{-n}(r) \right\}$$

(=  $+\infty$  si, pour tout  $n \geq 1$  la condition n'est pas satisfaite). La procédure de samplage correspondant est appelée algorithme monotone de Propp-Wilson.

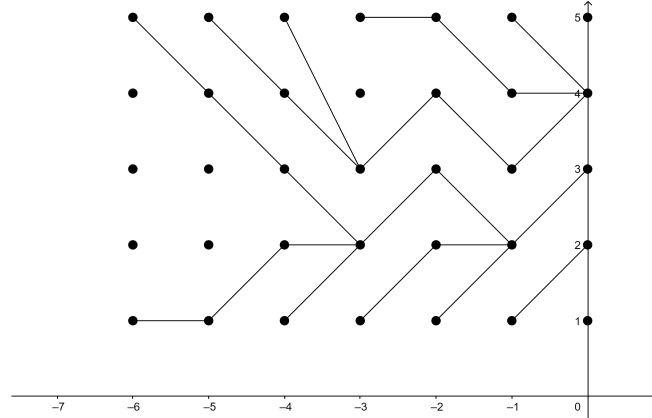


FIGURE 5 – Algorithme monotone de Propp-Wilson ; ici  $M = 6$

**Théorème 13.** *Le temps de coalescence backwards monotone  $M$  est presque sûrement fini. De plus, les variables aléatoires  $X_0^{-M}(1) = X_0^{-M}(r)$  sont de distribution  $\pi$ .*

*Démonstration.* On appelle  $N^+$  le temps de couplage forwards des deux chaînes  $\{X_n^0(1)\}$  et  $\{X_n^0(r)\}$ . La preuve que  $N^+$  est fini est immédiate dans notre cas puisque  $N^+$  est majoré par le premier temps  $n \geq 0$  tel que  $X_n^0(r) = 1$  qui est fini. Ayant cela, la preuve se déroule comme dans celle du théorème 12 grâce à la propriété de funneling de la chaîne, ie, pour tout  $n \in \mathbb{Z}$ , pour tout  $i \in E$ ,  $X_n^k(1) \preceq X_n^k(i) \preceq X_n^k(r)$  (voir notamment la figure précédente).  $\square$

### 3.2 CFTP : approche par blocs (Huber)

On considère un état stationnaire  $X$ . Puis, pour  $t$  fixé, soit  $Y = \phi_t(X, U)$ , où  $\phi$  est la fonction de mise à jour de la chaîne considérée et  $U \in [0, 1]^t$  sont les choix aléatoires effectués. Alors  $Y$  est aussi stationnaire, car c'est la composition de  $t$  états stationnaires.

La sortie de l'algorithme de CFTP sera toujours  $Y$ . On considère ensuite  $W = (U_1, U_2, \dots, U_t)$  uniforme sur  $[0, 1]^t$  puis un ensemble mesurable  $A \subset [0, 1]^t$ . Alors soit  $W \in A$ , soit  $W \notin A$ . D'où :

$$Y = \phi_t(X, W)\mathbf{1}(W \in A) + \phi_t(X, W)\mathbf{1}(W \notin A) \quad (2)$$

Supposons qu'on puisse trouver un ensemble  $A$  tel que lorsque  $W \in A$ ,  $\phi(X, W)$  ne dépende pas de  $X$ . C'est à dire, dès lors que  $W \in A$ , on ait  $\phi(x, w) = \phi(x', w)$  pour tout  $x, x' \in \Omega$ . Une telle chaîne à "oublié" son point de départ lors de son déplacement. Si cela arrive, il n'y a alors pas besoin de connaître la valeur de  $X$  pour obtenir  $Y$  dans la formule précédente.

Pour voir cela, on considère l'exemple suivant :

#### Exemple

Soit  $\Omega = \{0, 1, 2\}$ . Soit une fonction de mise à jour  $\phi$  telle que :

$$\phi(x, U) = x + \mathbf{1}(x < 2, U > \frac{1}{2}) - \mathbf{1}(x > 0, U \leq \frac{1}{2}).$$

Soient  $\{X_t\}$  et  $\{Y_t\}$  deux chaînes de Markov ayant  $\phi$  comme fonction de mise à jour et telles que  $X_0 = 0$  et  $Y_0 = 2$ . On suppose que  $U_0 = 0.64, U_1 = 0.234$  et  $U_2 = 0.1$ . On a donc les trajectoires suivantes :  $(X_0, X_1, X_2, X_3) = (0, 1, 0, 0)$  et  $(Y_0, Y_1, Y_2, Y_3) = (2, 2, 1, 0)$ .

Les deux chaînes ont donc fusionné au temps  $t = 3$ .

Pour cet exemple, on a donc :

$$\phi_3(0, W) = \phi_3(1, W) = \phi_3(2, W) = 0$$

D'où :  $\phi_3(\{0, 1, 2\}, W) = \{0\}$

Supposons que les 3 premiers pas étaient (haut, bas, bas). Cette suite de mouvements correspond à l'ensemble des valeurs appartenant à  $A_1 = (\frac{1}{2}, 1] \times [0, \frac{1}{2}] \times [0, \frac{1}{2}]$ . Si  $W \in A_1$ , alors  $\phi_3(\{0, 1, 2\}, W) = \{0\}$ .

Une autre suite de mouvements valide est (bas, bas, bas) ce qui correspond à  $A_2 = [0, \frac{1}{2}] \times [0, \frac{1}{2}] \times [0, \frac{1}{2}]$ . Bien sûr, si  $A_1$  et  $A_2$  fonctionnent, alors  $A_3 = A_1 \cup A_2$  fonctionne aussi. D'où  $\phi_3(\{0, 1, 2\}, W) = \{0\}$  pour tout  $W \in A_3$ .

Le but ici est de montrer que  $A$  n'a pas besoin d'être exactement l'ensemble de tous les mouvements qui coalesceront en un point. Cependant, plus  $A$  sera grand, plus la probabilité que  $W$  soit dans  $A$  sera grande.



Retour à l'équation (2). Dans le premier terme, la valeur de  $X$  n'est pas importante, alors on peut écrire :  $\phi_t(X, W)\mathbf{1}(W \in A) = \phi_t(x_0, W)\mathbf{1}(W \in A)$ , où  $x_0$  est un élément arbitraire de l'espace d'états. On a alors :

$$Y = \phi_t(x_0, W)\mathbf{1}(W \in A) + \phi_t(X, W)\mathbf{1}(W \notin A)$$

Donc lorsque  $W \in A$ , il n'y a pas besoin de connaître la valeur de  $X$  afin de calculer  $Y$ . Cependant, lorsque  $W \notin A$ , on doit évaluer  $\phi_t(X, W)$  pour obtenir  $Y$ . L'idée principale du CFTP est de trouver  $X \sim \pi$  en appelant récursivement le CFTP (c'est la partie "from the past" de l'algorithme) puis en calculant  $Y$  comme précédemment. Alors par définition de  $Y$ , on générera éventuellement un  $W \in A$  tel qu'on ait pas besoin de connaître  $X$ , et il ne restera plus qu'à dérouler tous les appels récursifs effectués.

Pour comprendre cela en pratique, on altère légèrement notre notation en ajoutant un indice de temps. Soit  $Y_0 = Y$ ,  $W_0 = W$ , et  $Y_{-1} = W$ . Avec cette notation :

$$Y_0 = \phi_t(x_0, W_0)\mathbf{1}(W_0 \in A) + \phi_t(Y_{-1}, W_0)\mathbf{1}(W_0 \notin A)$$

Donc si  $W_0 \in A$ , on est bon, sinon, on doit générer  $Y_{-1}$ . Ce sera fait de la même manière que lorsque l'on a généré  $Y_0$  :

$$Y_{-1} = \phi_t(x_0, W_{-1})\mathbf{1}(W_{-1} \in A) + \phi_t(Y_{-2}, W_{-1})\mathbf{1}(W_{-1} \notin A)$$

où  $W_{-1} \sim \text{Unif}([0, 1])^t$  et est indépendant de  $W_0$ .

En général,

$$Y_{-1} = \phi_t(x_0, W_{-i})\mathbf{1}(W_{-i} \in A) + \phi_t(Y_{-i-1}, W_{-i})\mathbf{1}(W_{-i} \notin A)$$

**Théorème 14.** *Théorème CFTP (Coupling From The Past)*

On suppose que  $\phi$  est une fonction de mise à jour pour une chaîne de Markov définie sur  $\Omega$ , telle que pour  $U = (U_0, U_{-1}, \dots, U_{-t-1}) \sim \text{Unif}([0, 1]^t)$  on ait :

- Pour  $Y \sim \pi$ ,  $\phi(Y, U_0) \sim \pi$
- Il existe un ensemble  $A \subseteq [0, 1]^t$  tel que  $\mathbb{P}(U \in A) > 0$  et  $\phi_t(\Omega, A) = \{x\}$  pour un certain  $x \in \Omega$

Posons alors, pour tout  $x \in \Omega$ ,

$$Y_0 = \phi_t(x_0, U_0)\mathbf{1}(U_0 \in A) + \phi_t(\phi_t(x_0, U_{-1}), U_0)\mathbf{1}(U_{-1} \in A) + \\ \phi_t(\phi_t(\phi_t(x_0, U_{-2}), U_{-1}), U_0)\mathbf{1}(U_{-2} \in A) + \dots$$

Alors  $Y_0 \sim \pi$ .

*Démonstration.* Soit  $x_0 \in \Omega$ . Le résultat est immédiat en utilisant le théorème fondamental de la simulation parfaite, en posant  $g(U) = \phi_t(x_0, U)$ ,  $b(U) = \mathbf{1}(U \in A)$ , et  $f(X, U) = \phi_t(X, U)$ .  $\square$

L'idée clé qu'ont eu Propp et Wilson est qu'il n'est pas nécessaire de connaître tous les termes de la suite pour trouver  $Y_0$ . Il suffit en effet de connaître  $U_0, \dots, U_{-T}$ , où  $U_T \in A$ . Tant que  $\mathbb{P}(U \in A) > 0$ , alors  $T$  suivra une loi géométrique de paramètre  $\mathbb{P}(U \in A)$ . Le pseudo-code suivant accomplit alors cette tâche :

---

**Coupling-from-the-past** Sortie :  $Y \sim \pi$

---

```

1: Tirer  $U \sim \text{Unif}([0, 1]^t)$ 
2: if  $U \in A$  then
3:   Rendre  $\phi_t(x_0, U)$  (où  $x_0$  est un élément arbitraire de  $\Omega$ )
4: else
5:    $X \leftarrow$  Coupling-from-the-past
6:   Rendre  $\phi_t(X, U)$ 
7: end if
```

---

Cet algorithme est un algorithme récursif, autorisé à s'appeler à nouveau en ligne 5. Lorsque cela arrive, les valeurs de  $U$  ne sont pas passées en paramètre : le nouvel appel génère en ligne 1 ses propres valeurs de  $U$  indépendantes des précédentes. C'est une étape importante, car sinon le théorème fondamental de la simulation parfaite ne s'appliquerait pas. On considère à présent le temps d'exécution de l'algorithme mesuré par le nombre d'appels à  $\phi$ .

**Lemme 9.** *Le nombre d'appels à  $\phi$  dans l'algorithme Coupling-from-the-past est en moyenne  $\frac{t}{\mathbb{P}(U \in A)}$ .*

*Démonstration.* Soit  $T = \inf \{\tau \geq 1 \mid U_\tau^t \in A\}$ .  $T$  suit par définition la loi géométrique de paramètre  $\mathbb{P}(U \in A)$

Alors, l'algorithme s'arrête au temps  $T$  après être passé  $T - 1$  fois dans la partie ELSE puis une fois dans la première partie.

Dans la partie ELSE, l'algorithme fait appel  $t$  fois à  $\phi$ , d'où :  $(T - 1) * t$  appels à  $\phi$ .

La première partie fait aussi  $t$  appels à  $\phi$ . En sommant, on obtient le nombre total d'appels :  $T * t$ .

Or, on connaît la loi de  $T$ , donc, en prenant l'espérance, on obtient :

$$\mathbb{E}[T * t] = t * \mathbb{E}[T] = \frac{t}{\mathbb{P}(U \in A)}$$

$\square$

Pour garder le temps d'exécution suffisamment bas, il est important que  $t$  soit suffisamment grand pour que  $\mathbb{P}(U \in A)$  soit raisonnablement élevé. Dès lors que cette condition est achevée,  $t$  devra être le plus petit possible, afin de garder un nombre de mise à jour faible. Cependant, il existe un moyen de contourner ce problème.

### 3.2.1 Variation de la taille des blocs

Lors de la construction du CFTP, une fonction de mise à jour à été composée avec elle-même  $t$  fois. On appelle cet ensemble de  $t$  pas un bloc. On suppose qu'il y ait un seuil  $t'$  tel que  $\mathbb{P}(U \in A) = 0$  si  $t < t'$ . En pratique, un tel seuil est très difficile à calculer : il est donc commun d'utiliser des blocs de taille qui augmente afin d'être sûr que  $t$  dépassera éventuellement ce seuil.

En général, il n'y a aucune raison pour que le même nombre de pas soit utilisé pour chaque bloc.

Le théorème fondamental de la simulation parfaite s'adapte facilement au cas où  $b, g$  et  $f$  changent à chaque étape de la récursion.

**Théorème 15** (Théorème fondamental de la simulation parfaite (seconde forme)). *On suppose que pour  $U_1, U_2, \dots$  iid, on ait des suites de fonctions calculables  $\{b_t\}, \{g_t\}$  et  $\{f_t\}$  telles que chaque  $\{b_t\}$  ait pour image  $\{0, 1\}$  et que  $\prod_{t=1}^{\infty} \mathbb{P}(b_t(U) = 0) = 0$ . Soit  $X$  la variable aléatoire telle que, pour tout  $t$  :*

$$X \sim b_t(U)g_t(U) + (1 - b_t(U))f_t(X, U)$$

*Soit  $T = \inf \{t : b_t(U_t) = 1\}$ . Alors,*

$$Y = f_0(\dots f_{T-2}(f_{T-1}(g_T(U_T), U_T), U_{T-1}), \dots, U_1)$$

*a la même distribution que  $X$  et  $\mathbb{E}[T] = \frac{1}{\mathbb{P}(b_t(U_t)=1)}$*

*Démonstration.* La preuve est presque identique à celle du théorème fondamental de la simulation parfaite présentée dans le chapitre 1.  $\square$