

Mémoire d'initiation à la recherche

---

# SIMULATION DE MARCHES ALÉATOIRES SIMPLES SUR $\mathbb{Z}$ CONDITIONNÉES

Elsa AZOULAY, Louise DURAND-JANIN, Jean PICHON

---



Encadré par Julien STOEHR

M1 Mathématiques et Applications  
Majeure Statistiques  
2022 – 2023

# Table des matières

1	Introduction du modèle	2
2	Méthode par tirages successifs	4
3	Méthode d'acceptation-rejet	6
4	Simulation par chaînes de Markov	8
5	Vers les méthodes de couplage	21
6	Conclusion	26
	Références	27

# 1 Introduction du modèle

L'idée de *marche aléatoire* a été introduite par le biostatisticien Karl Pearson en 1905 pour répondre au problème suivant. Un homme part d'un point, parcourt une distance fixée en ligne droite, dans une certaine direction, puis parcourt cette même distance dans une autre direction et ainsi de suite. Quelle est la probabilité qu'il se trouve à une certaine distance de son point de départ après  $n$  étapes ? La marche aléatoire possède la propriété de Markov, de non-dépendance du mouvement présent au passé. Pour cette raison, on l'associe souvent à la marche de l'ivrogne, ne se souvenant pas d'où il vient.

**Définition 1.1** (Marche aléatoire simple sur  $\mathbb{Z}$ ). Soit  $(U_n)_{n \in \mathbb{N}}$  une suite de variables aléatoires indépendantes et identiquement distribuées (i.i.d.) suivant la loi de Rademacher ou de façon équivalente suivant la loi uniforme sur  $\{-1, 1\}$ , notée  $\mathcal{U}(\{-1, 1\})$ . On appelle *marche aléatoire simple sur  $\mathbb{Z}$*  le processus markovien  $(X_n)_{n \in \mathbb{N}}$  issu de  $X_0 = 0$  défini pour tout  $n \in \mathbb{N}$  par la récurrence aléatoire :

$$X_{n+1} = X_n + U_{n+1}. \quad (1)$$

Pour  $N \in \mathbb{N}$  fixé, on appellera **trajectoire** une réalisation  $\mathbf{x} = (x_0, \dots, x_N)$  d'un tel processus  $(X_0, \dots, X_N)$ , et nous noterons  $E$  l'ensemble des trajectoires :

$$E = \{\mathbf{x} = (x_0, x_1, \dots, x_N) \in \mathbb{Z}^{N+1} \mid x_0 = 0 \text{ et } \forall i \in \llbracket 0, N-1 \rrbracket, |x_{i+1} - x_i| = 1\}.$$

Les marches aléatoires simples possèdent plusieurs propriétés intéressantes, notamment la récurrence.

**Proposition 1.1.** *La marche aléatoire simple sur  $\mathbb{Z}$  est récurrente en tout point.*

*Démonstration.* La marche aléatoire simple est irréductible. En effet, deux éléments  $x$  et  $y$  de  $\mathbb{Z}$  peuvent être reliés par un chemin de taille  $n \geq |x - y|$ , avec une probabilité égale à  $2^{-n}$ , donc strictement positive.

Ainsi, tous ses états sont soit transients soit récurrents. Pour montrer que tous ses états sont récurrents, il suffit donc de montrer la récurrence en 0. On veut montrer que la marche aléatoire simple revient une infinité de fois en 0. Pour cela,

introduisons la variable aléatoire  $N_0$  telle que

$$N_0 = \sum_{k \geq 0} \mathbb{1}_{\{X_k=0\}}.$$

Alors, 0 est un état récurrent si, et seulement si,  $\mathbb{E}[N_0] = +\infty$ . On a :

$$\mathbb{E}[N_0] = \sum_{k \geq 0} \mathbb{P}(X_k = 0).$$

Pour  $k \in \mathbb{N}$  :

$$\mathbb{P}(X_k = 0) = \mathbb{P}\left(\sum_{i=1}^k \{X_i - X_{i-1}\} = 0\right) = \begin{cases} 0 & \text{si } k \text{ est impair.} \\ \frac{1}{2^k} \binom{k}{k/2} & \text{si } k \text{ est pair.} \end{cases}$$

Par la formule de Stirling :

$$\frac{1}{2^k} \binom{k}{k/2} \underset{k \rightarrow +\infty}{\sim} \frac{\sqrt{2}}{\sqrt{\pi k}}.$$

Donc,  $\sum_{k \geq 0} \mathbb{P}(X_k = 0)$  diverge. Ainsi,  $\mathbb{E}[N_0] = +\infty$  et 0 est bien un état récurrent de la marche aléatoire simple.  $\square$

La question d'intérêt de ce mémoire porte alors sur la simulation d'un cas particulier de trajectoire.

**Définition 1.2** (Marche aléatoire simple sur  $\mathbb{Z}$  conditionnée). *Une marche aléatoire simple  $(X_n)_{n \in \mathbb{N}}$  sur  $\mathbb{Z}$  est dite conditionnée à revenir à un état  $s \in \mathbb{Z}$  au temps  $N \in \mathbb{N}$  lorsque  $X_N = s$ .*

Dans la suite, on s'intéressera aux marches aléatoires simples conditionnées à revenir en 0 à un temps  $N \in \mathbb{N}$  fixé, qu'on appellera ponts. On note  $F$  l'ensemble des ponts :

$$F = \{\mathbf{x} = (x_0, x_1, \dots, x_N) \in E \mid x_N = 0\}.$$

Une marche ne peut revenir en 0 que si elle compte autant d'incrément 1 que -1. En plus de caractériser un pont, ce critère implique également que  $N$  est nécessairement pair. L'objectif est alors de simuler des ponts suivant la mesure uniforme sur  $F$ , notée  $\mu_F$ . S'il est facile d'obtenir une trajectoire de longueur  $N$ , i.e. tirer uniformément dans  $E$ , en utilisant la récurrence aléatoire (1), la simulation d'un

pont repose sur des mécanismes différents.

Dans ce mémoire nous comparons algorithmiquement et théoriquement différentes méthodes de simulation de ponts. Nous commençons avec la partie 2 qui aborde une méthode par tirages successifs, basée sur la caractérisation des incréments d'un pont et qui simule de manière exacte suivant  $\mu_F$ . Dans la section 3, nous regardons une autre méthode de simulation exacte : la méthode d'acceptation-rejet qui nécessite simplement de savoir simuler une trajectoire de longueur  $N$ . Puis, nous nous intéressons dans la section 4 à des méthodes de simulation par chaînes de Markov. En utilisant une dynamique dite de « *corner flips* », on met en place un processus Markovien de mesure stationnaire  $\mu_F$ . En utilisant en partie 5 des méthodes dites de couplage, il est alors possible de simuler exactement selon  $\mu_F$ .

## 2 Méthode par tirages successifs

Par caractérisation d'un pont, une première méthode serait de tirer uniformément les  $N$  incréments de notre pont successivement, sans remise, dans l'ensemble  $I$  de cardinal  $N$  suivant :

$$I = \{\underbrace{1, \dots, 1}_{N/2}, \underbrace{-1, \dots, -1}_{N/2}\}.$$

**Proposition 2.1.** *Un tel tirage est réalisé suivant la loi cible  $\mu_F$ .*

*Démonstration.* Montrons que deux ponts qui diffèrent d'un coin ont la même probabilité d'être simulés par cette méthode. On remarque que tout pont admet au moins un coin. Pour tout  $\mathbf{x} = (x_0, \dots, x_N) \in F$ , on notera  $\tilde{\mathbf{x}}$  son vecteur d'incréments :

$$\tilde{\mathbf{x}} = (i_1 := x_1 - x_0, \dots, i_N := x_N - x_{N-1}),$$

où, on le rappelle, pour tout  $l \in \llbracket 1, N \rrbracket$ ,

$$i_l \in \{-1, 1\}, \quad \text{et} \quad \sum_{l=1}^N i_l = 0.$$

Soit  $\mathbf{x}, \mathbf{y} \in F$  qui diffèrent en un seul coin à l'indice noté  $k$ . On note  $\tilde{\mathbf{x}}$  et  $\tilde{\mathbf{y}}$  leurs

vecteurs d'incrments respectifs. On a donc

$$\tilde{\mathbf{x}} = (i_1, \dots, i_{k-1}, 1, -1, i_{k+2}, \dots, i_N),$$

$$\tilde{\mathbf{y}} = (i_1, \dots, i_{k-1}, -1, 1, i_{k+2}, \dots, i_N).$$

On notera  $(T_i)_{i \in \llbracket 1, N \rrbracket}$  une suite de variables alatoires reprsentant les rultats des tirages de notre exprience. Calculons les probabilités d'obtention de  $\tilde{\mathbf{x}}$  et de  $\tilde{\mathbf{y}}$ . Pour tout  $k \in \llbracket 1; N-1 \rrbracket$ , on note  $A_k$  les évènements suivants :

$$A_k = \{(T_1, \dots, T_k) = (i_1, \dots, i_k)\}.$$

On note

$$P_b = \mathbb{P}(T_1 = i_1) \prod_{l=2}^{k-1} \mathbb{P}(T_l = i_l | A_{l-1}),$$

$$P_a = \prod_{l=k+2}^N \mathbb{P}(T_l = i_l | A_{l-1}).$$

On calcule alors les probabilités d'obtention de nos deux ponts :

$$\mathbb{P}((T_1, \dots, T_N) = \tilde{\mathbf{x}}) = P_b \mathbb{P}(T_k = 1 | A_{k-1}) \mathbb{P}(T_{k+1} = -1 | A_k) P_a,$$

$$\mathbb{P}((T_1, \dots, T_N) = \tilde{\mathbf{y}}) = P_b \mathbb{P}(T_k = -1 | A_{k-1}) \mathbb{P}(T_{k+1} = 1 | A_k) P_a.$$

On remarque que ces deux tirages sont équiprobables si, et seulement si,

$$\mathbb{P}(T_k = 1 | A_{k-1}) \mathbb{P}(T_{k+1} = -1 | A_k) = \mathbb{P}(T_k = -1 | A_{k-1}) \mathbb{P}(T_{k+1} = 1 | A_k). \quad (2)$$

On note  $N_1$  la variable alatoire comptant le nombre d'incrments 1 obtenus entre le premier tirage et le  $(k-1)$ -ème, c'est-à-dire

$$N_1 = \sum_{i=1}^{k-1} \mathbb{1}_{\{T_i=1\}}.$$

Comme on tire uniformément sans remise, on peut calculer ces probabilités :

$$\mathbb{P}(T_k = 1 | A_{k-1}) = \frac{\frac{N}{2} - N_1}{N - (k-1)},$$

$$\begin{aligned}\mathbb{P}(T_{k+1} = -1|A_k) &= \frac{\frac{N}{2} - \{k - (1 + N_1)\}}{N - (k - 2)}, \\ \mathbb{P}(T_k = -1|A_{k-1}) &= \frac{\frac{N}{2} - \{k - (1 + N_1)\}}{N - (k - 1)}, \\ \mathbb{P}(T_{k+1} = 1|A_k) &= \frac{\frac{N}{2} - N_1}{N - (k - 2)}.\end{aligned}$$

L'égalité (2) étant donc vraie, tous les ponts qui diffèrent d'un coin sont équiprobables. Comme tout pont admet au moins un voisin qui diffère d'un coin, par généralisation de l'équiprobabilité des voisins, on obtient que tous les ponts sont équiprobables, donc chacun de probabilité  $\frac{1}{|F|}$ .  $\square$

**Données :** Taille  $N$

**Résultat :** Simulation d'un pont de taille  $N$  selon  $\mu_F$

$I \leftarrow$  vecteur de longueur  $N$  contenant moitié de 1, moitié de -1 ;

$\mathbf{x} \leftarrow$  vecteur des  $N$  tirages uniformes successifs et sans remise dans  $I$  ;

**Algorithme 1 :** Tirages successifs pour simuler suivant  $\mu_F$

**Remarque.** En langage R, des tirages uniformes successifs et sans remise peuvent être réalisés avec la fonction prédéfinie « *sample* ». C'est donc celle que nous avons utilisé.

Cette première méthode permet de simuler de manière exacte et très efficace avec la fonction *sample*. Nous allons maintenant voir des méthodes faisant appel à des notions mathématiques plus poussées.

### 3 Méthode d'acceptation-rejet

La méthode d'acceptation-rejet permet de simuler suivant la mesure uniforme sur  $F$  via une loi de probabilité qui domine  $\mu_F$  et selon laquelle il est facile de simuler. Ici nous pouvons utiliser le fait que les trajectoires sont faciles à simuler car il s'agit d'un simple échantillon *i.i.d.* de loi  $\mathcal{U}(\{-1, 1\})$  et de taille  $N$ . Par ailleurs, comme  $F \subset E$ , on a, pour tout  $\mathbf{x} \in F$ ,

$$\mu_F(\mathbf{x}) = \frac{1}{|F|} \mathbb{1}_{\{\mathbf{x} \in F\}} \leq \frac{1}{|F|} \mathbb{1}_{\{\mathbf{x} \in E\}} \leq \frac{|E|}{|F|} \frac{1}{|E|} \mathbb{1}_{\{\mathbf{x} \in E\}}.$$

Nous pouvons donc utiliser l'algorithme du rejet, tiré du polycopié [Sto23], pour générer un pont avec la mesure uniforme  $\mu_E$  sur  $E$  comme densité instrumentale.

Soient  $(U_n)_{n \geq 1}$  une suite de variables *i.i.d.* de loi  $\mathcal{U}([0, 1])$  et  $(\mathbf{Y}_n)_{n \geq 1} = (Y_1, \dots, Y_n)_{n \geq 1}$  une suite de variables *i.i.d.* de loi de densité  $\mu_E$  telles que ces deux suites soient indépendantes. Alors, on définit le temps d'arrêt  $T$  par

$$T := \inf\{n \geq 1 : U_n \leq \alpha(\mathbf{Y}_n)\}, \text{ où } \alpha(\mathbf{Y}_n) := \frac{|F|\mu_F(\mathbf{Y}_n)}{|E|\mu_E(\mathbf{Y}_n)} = \mathbb{1}_{\{\mathbf{Y}_n \in F\}},$$

$\mathbf{Y}_T$  suit la loi de densité  $\mu_F$ .

L'algorithme consiste donc à simuler une trajectoire jusqu'à obtenir un pont. En utilisant le fait qu'un pont est caractérisé par une suite d'incrémentes sommant à zéro, l'algorithme du rejet est mis en place comme suit :

**Résultat :** Simulation d'un pont de taille  $N$  selon  $\mu_F$   
 $\mathbf{x} \leftarrow N$  réalisations de variables *i.i.d.* suivant  $\mathcal{U}(\{-1; 1\})$ ;  
**tant que**  $\sum_{i=1}^N x_i \neq 0$  **faire**  
     $\mathbf{x} \leftarrow N$  réalisations de variables *i.i.d.* suivant  $\mathcal{U}(\{-1; 1\})$ ;  
**fin**

**Algorithme 2 :** Acceptation-rejet pour simuler suivant  $\mu_F$

L'efficacité de la méthode peut-être quantifiée par le temps d'attente moyen d'acceptation d'une trajectoire, qui dépend de  $N$ . On a :

$$|E| = 2^N \quad \text{et} \quad |F| = \binom{N}{N/2}.$$

Une marche aléatoire simple étant récurrente en 0, la trajectoire reviendra une infinité de fois en ce point. Mais la probabilité d'acceptation correspond à la probabilité de retour en 0 de la trajectoire à un temps  $N$  fixé, et vaut :

$$\frac{|F|}{|E|} = \binom{N}{N/2} \frac{1}{2^N} \underset{N \rightarrow +\infty}{\sim} \frac{\sqrt{2}}{\sqrt{\pi N}}.$$

La probabilité d'acceptation décroît donc à une vitesse de l'ordre de  $\mathcal{O}(\sqrt{N})$ . Par exemple, pour simuler des ponts de taille  $N = 50$ , la probabilité qu'un élément de  $E$  soit également dans  $F$  est à peine supérieure à 10%, comme on peut l'observer



sur la Figure 1.

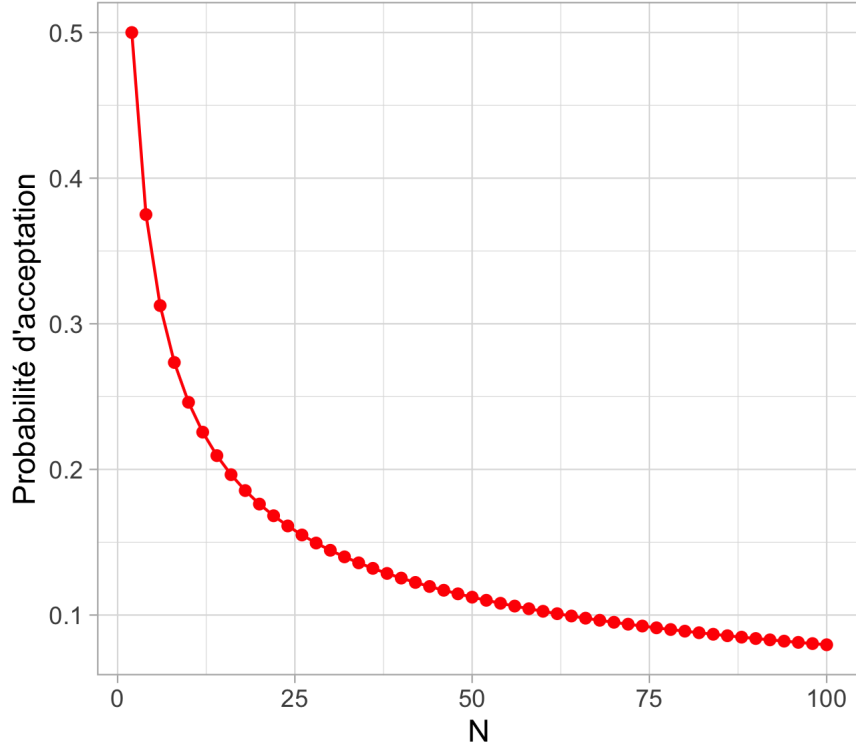


FIGURE 1 – Évolution de la probabilité d’acceptation de la méthode d’acceptation-rejet en fonction de la longueur  $N$  du pont.

La méthode d’acceptation constitue donc une méthode de simulation exacte de notre loi cible  $\mu_F$ . Mais du fait de sa complexité en  $\mathcal{O}(\sqrt{N})$ , pour des longueurs de chaînes grandes, elle entraîne des coûts de simulation élevés. Dans la suite nous allons regarder s’il est possible d’être plus efficace avec des méthodes de type MCMC.

## 4 Simulation par chaînes de Markov

Les algorithmes de type MCMC sont particulièrement intéressants pour générer des échantillons d’une distribution qu’il serait difficile d’échantillonner directement. Pour notre problème, ils consistent à générer une chaîne de Markov à espace d’états dans un sous ensemble de  $E$  et qui admet  $\mu_F$  pour mesure stationnaire. Afin d’en souligner l’intérêt, commençons par un exemple naïf.

**Données :** Taille de pont  $N$   
**Résultat :** Simulation d'un pont de taille  $N$   
 $\mathbf{x} \leftarrow$  vecteur de  $N$  réalisations de  $\mathcal{U}(\{-1; 1\})$ ;  
 $n_1 \leftarrow$  nombre d'occurrences de 1 dans  $\mathbf{x}$ ;  
**si**  $n_1 > \frac{N}{2}$  **alors**  
    | On change les  $(n_1 - \frac{N}{2})$  premiers indices de  $\mathbf{x}$  valant 1 en -1;  
**sinon**  
    | On change les  $(\frac{N}{2} - n_1)$  premiers indices de  $\mathbf{x}$  valant -1 en 1;  
**fin**

**Algorithme 3 :** Première approche

Considérons un générateur « par modification » : après simulation d'une trajectoire, on la modifie de telle sorte à ce qu'elle devienne un pont. L'idée est de choisir uniformément un élément de  $E$  et à forcer le passage dans  $F$ . Cela permet d'éviter de procéder comme dans la méthode d'acceptation-rejet : tirer un grand nombre d'éléments de  $E$  jusqu'à obtenir un élément de  $F$ . On choisit ici de modifier arbitrairement les premiers incréments de la trajectoire pour avoir une quantité égale d'incrément positifs et négatifs, ce qui caractérise un élément de  $F$ , cf. Algorithme 3. La Figure 2 représente la distribution empirique de 10000 ponts de longueur  $N = 6$  obtenus à l'aide de l'Algorithme 3.

On y relève un défaut d'uniformité qu'on peut confirmer par un test d'adéquation du  $\chi^2$ . L'hypothèse nulle d'un tel test est  $H_0$  : « cet échantillon est distribué suivant  $\mu_F$  ». Comme attendu, la p-valeur étant inférieure à  $3.10^{-16}$ , on rejette  $H_0$ . Cette approche biaisée ne génère pas uniformément des ponts car les modifications envisagées ne mettent pas autant de poids sur chaque pont.

**Remarque.** *Pour pouvoir tracer le barplot de la Figure 2, il faut être en mesure de distinguer et d'identifier chaque pont. Pour cela, le vecteur de coordonnées est transformé en un vecteur d'incréments, puis les -1 sont remplacés par des 0, et le nouveau vecteur est écrit en base 2. Le nombre binaire ainsi obtenu est unique à chaque pont.*

Plutôt que de faire une seule modification de trajectoire pour forcer l'appartenance à  $F$ , on pourrait plutôt partir d'un pont quelconque et lui appliquer des modifications successives, qui prennent la forme d'un processus Markovien. Pour pouvoir simuler suivant  $\mu_F$ , ce processus Markovien doit admettre  $\mu_F$  comme mesure stationnaire.

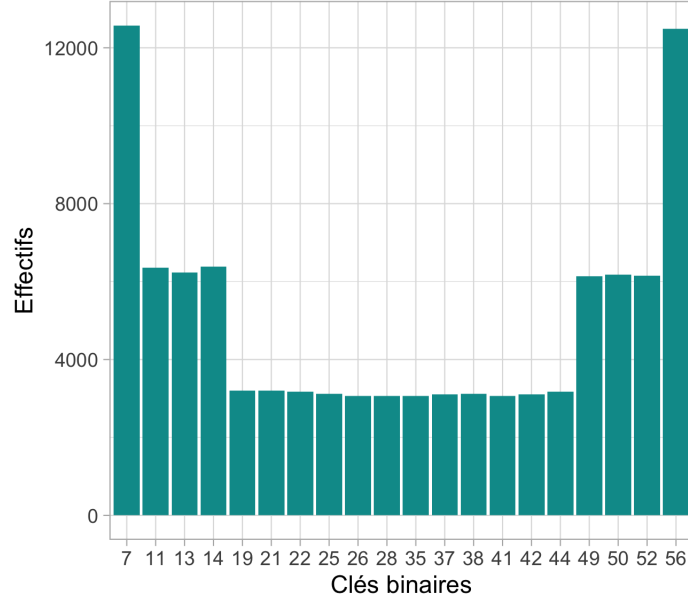


FIGURE 2 – Distribution empirique de 100000 ponts de longueur 6 obtenus avec l’Algorithme 2.

#### 4.1 Dynamique « *corner flips* »

Pour construire un tel processus, nous allons considérer la dynamique dite de « *corner flips* », qui repose sur les deux opérateurs déterministes suivants : pour tout  $\mathbf{x} \in F$ , et  $i \in \llbracket 1, N - 1 \rrbracket$ ,

$$T_d : (\mathbf{x}, i) \rightarrow \begin{cases} (x_0, \dots, x_{i-1}, x_i - 2, x_{i+1}, \dots, x_N) & \text{si } x_{i-1} = x_{i+1} = x_i - 1, \\ \mathbf{x} & \text{sinon,} \end{cases}$$

et

$$T_u : (\mathbf{x}, i) \rightarrow \begin{cases} (x_0, \dots, x_{i-1}, x_i + 2, x_{i+1}, \dots, x_N) & \text{si } x_{i-1} = x_{i+1} = x_i + 1, \\ \mathbf{x} & \text{sinon.} \end{cases}$$

Lorsque c’est possible,  $T_d$  retourne un coin du haut vers le bas et  $T_u$  un coin du bas vers le haut. On peut introduire un processus markovien d’espace d’états  $F$  basé sur une succession de « *corner flips* », qui admet pour mesure invariante notre mesure cible  $\mu_F$ .

Soit  $(U_n)_{n \in \mathbb{N}}$  une suite de variables *i.i.d.* uniformes sur  $\llbracket 1, N - 1 \rrbracket$ , et  $(B_n)_{n \in \mathbb{N}}$

suite de variables *i.i.d.* suivant  $Ber(1/2)$ , toutes deux indépendantes. Pour toute initialisation  $\mathbf{X}_0 \in F$ ,

$$\mathbf{X}_{n+1} = \sum_{k=1}^{N-1} \mathbb{1}_{\{U_{n+1}=k\}} [B_{n+1}T_d(\mathbf{X}_n, k) + \{1 - B_{n+1}\}T_u(\mathbf{X}_n, k)]. \quad (3)$$

Cette dynamique revient, à chaque étape, à choisir uniformément un point de  $\mathbf{X}_n$  et si c'est un coin, de le retourner avec probabilité  $1/2$ . Le noyau de transition associé à cette chaîne de Markov est donné ci-dessous.

En notant  $C(\mathbf{x})$  le nombre de coins d'un pont  $\mathbf{x}$ , on a pour tout  $(\mathbf{x}, \mathbf{y}) \in F^2$ ,

$$Q_{1/2}(\mathbf{x}, \mathbf{y}) := \mathbb{P}(\mathbf{X}_{n+1} = \mathbf{y} | \mathbf{X}_n = \mathbf{x}) \\ = \begin{cases} 0 & \text{si } \mathbf{y} \neq \mathbf{x} \text{ et } \forall i \in \llbracket 1, N-1 \rrbracket, \mathbf{y} \neq T_d(\mathbf{x}, i) \text{ et } \mathbf{y} \neq T_u(\mathbf{x}, i), \\ 1 - \frac{C(\mathbf{x})}{2(N-1)} & \text{si } \mathbf{x} = \mathbf{y}, \\ \frac{1}{2(N-1)} & \text{si } \mathbf{y} \neq \mathbf{x}, \exists i \in \llbracket 1, N-1 \rrbracket \mid \mathbf{y} = T_d(\mathbf{x}, i) \text{ ou } \mathbf{y} = T_u(\mathbf{x}, i). \end{cases}$$

$Q_{1/2}$  est bien stochastique. On note

$$H_{\mathbf{y}} := \{\mathbf{x} \in F \text{ tel que } \exists i \in \llbracket 1, N-1 \rrbracket, \mathbf{x} = T_u(\mathbf{y}, i) \text{ ou } \mathbf{x} = T_d(\mathbf{y}, i)\}.$$

Comme le cardinal de  $H_{\mathbf{y}}$  vaut le nombre de coins de  $\mathbf{y}$ , on a

$$\begin{aligned} \sum_{\mathbf{x} \in F} Q_{1/2}(\mathbf{x}, \mathbf{y}) &= \sum_{\mathbf{x} \in F} \left( \left\{ 1 - \frac{C(\mathbf{x})}{2(N-1)} \right\} \mathbb{1}_{\{\mathbf{x}=\mathbf{y}\}} + \frac{1}{2(N-1)} \mathbb{1}_{\{H_{\mathbf{y}}\}} \right) \\ &= \left( 1 - \frac{C(\mathbf{y})}{2(N-1)} \right) + \frac{1}{2(N-1)} |H_{\mathbf{y}}| \\ &= 1 - \frac{C(\mathbf{y})}{2(N-1)} + \frac{C(\mathbf{y})}{2(N-1)} = 1. \end{aligned}$$

De plus,  $Q_{1/2}(\mathbf{x}, \mathbf{y}) \in [0, 1]$  pour tout  $(\mathbf{x}, \mathbf{y}) \in F^2$ . Donc  $Q_{1/2}$  est bien un noyau de transition.

**Lemme 4.1.** *La matrice de transition  $Q_{1/2}$  est symétrique.*

*Démonstration.* On remarque les équivalences suivantes : pour tout  $i \in \llbracket 1, N-1 \rrbracket$ ,

$$\mathbf{x} \neq T_u(\mathbf{y}, i) \quad \Leftrightarrow \quad \mathbf{y} \neq T_d(\mathbf{x}, i) \quad \text{et} \quad \mathbf{x} = T_u(\mathbf{y}, i) \quad \Leftrightarrow \quad \mathbf{y} = T_d(\mathbf{x}, i).$$

Elles nous montrent que les différents cas de la matrice  $Q_{1/2}$  sont symétriques. Nécessairement cela implique :

$$\forall(\mathbf{x}, \mathbf{y}) \in F^2, \quad Q_{1/2}(\mathbf{x}, \mathbf{y}) = Q_{1/2}(\mathbf{y}, \mathbf{x}),$$

c'est-à-dire que  $Q_{1/2}$  est symétrique. □

Afin de démontrer qu'un tel processus admet pour mesure stationnaire notre mesure de probabilité cible, nous utiliserons ce résultat sur les chaînes de Markov.

**Lemme 4.2.** *Toute chaîne  $(X_n)_{n \in \mathbb{N}}$  irréductible sur un espace d'états  $A$  dénombrable admet une unique mesure de probabilité invariante.*

*Démonstration.* Soit  $(X_n)_{n \in \mathbb{N}}$  une chaîne irréductible sur  $A$  dénombrable, de matrice de transition  $P$ . Commençons par prouver l'existence d'une telle mesure. Soit  $\nu$  une mesure de probabilité sur  $A$ . Soient  $n \in \mathbb{N}$  et  $x \in A$ . On introduit

$$\nu_n(x) = \frac{1}{n} \sum_{k=1}^n \nu P^k(x).$$

Il s'agit d'une suite de mesures de probabilité. En effet, il est d'abord évident que c'est bien une suite de mesures car elles sont positives et qu'elles sont  $\sigma$ -additives par propriété de  $\nu$  et de  $P^n$ . On peut voir par récurrence que ce sont des mesures de probabilité, c'est-à-dire qu'elles intègrent à 1.

**(Initialisation)**

$$\nu_1(A) = \sum_{x \in A} \nu_1(x) = \sum_{x \in A} \sum_{y \in A} \nu(y) P(y, x) = \sum_{y \in A} \nu(y) \sum_{x \in A} P(y, x) = 1$$

par stochasticité de  $P$ .

**(Hérédité)**

$$\nu_{n+1}(x) = \frac{n}{n+1} \nu_n(x) + \frac{1}{n+1} \nu P^{n+1}(x).$$

Or  $P^{n+1}$  n'est pas forcément stochastique, mais  $P$  l'est. Donc

$$\sum_{x \in A} \nu P^{n+1}(x) = \sum_{x \in A} \sum_{y \in A} \nu(y) \sum_{z \in A} P^n(y, x) P(z, x) = \sum_{x \in A} \nu P(z, x) = 1$$

par hypothèse de récurrence et par stochasticité de  $P$ .  $(\nu_n)$  est donc bien une suite de probabilités sur  $[0, 1]^{Card(A)}$  qui est un ensemble fermé borné dans  $A$  de dimension finie, donc compact. Il existe donc une extraction  $\phi$  telle que  $(\nu_{\phi(n)})$  converge, dont on note  $\mu'_F$  la limite. Montrons que  $\mu'_F$  est bien une probabilité et qu'elle est invariante pour  $P$ .

$$\sum_{y \in A} \mu'_F(y) = \sum_{y \in A} \lim_{n \rightarrow +\infty} \nu_{\phi(n)}(y) = \lim_{n \rightarrow +\infty} \sum_{y \in A} \nu_{\phi(n)}(y) = 1.$$

Or pour tout  $n \in \mathbb{N}^*$ , et pour tout  $y \in A$ ,

$$\begin{aligned} \nu_n P(y) &= \frac{1}{n} \sum_{k=1}^n \nu P^{k+1}(y) \\ &= \frac{1}{n} \sum_{k=1}^n \nu P^k(y) - \frac{1}{n} \nu P(y) + \frac{1}{n} \nu P^{n+1}(y) \\ &= \nu_n(y) + \frac{1}{n} (\nu P^{n+1}(y) - \nu P(y)). \end{aligned}$$

Donc

$$|\nu_n P(y) - \nu_n(y)| = \frac{1}{n} |\nu P^{n+1}(y) - \nu P(y)| \leq \frac{1}{n} (|\nu P^{n+1}(y)| + |\nu P(y)|) \leq \frac{2}{n} \xrightarrow{n \rightarrow +\infty} 0.$$

C'est vrai en particulier pour la sous-suite d'extraction  $\phi$ . Donc

$$\lim_{n \rightarrow +\infty} \nu_{\phi(n)} P(y) - \nu_{\phi(n)}(y) = 0 \text{ donc } \mu'_F P(y) - \mu'_F(y) = 0.$$

$\mu'_F$  étant bien invariante pour  $P$ , et  $P$  étant la matrice de transition de  $(X_n)$ , on a montré l'existence d'une probabilité invariante pour tout chaîne irréductible définie sur un espace d'états  $A$  dénombrable.

Pour l'unicité de la mesure, on utilise un résultat de l'article [Bak02], qui stipule que, comme  $(X_n)$  est irréductible sur  $A$  dénombrable, elle est aussi récurrente. Comme elle est irréductible et récurrente, toutes ses mesures stationnaires sont égales à une constante près, ce qui donne l'unicité de la probabilité stationnaire.  $\square$

Ce résultat nous sera utile pour démontrer le résultat central de la section, énoncé ci-dessous.

**Proposition 4.1.** *Le processus  $(X_n)_{n \in \mathbb{N}}$  admet pour unique mesure invariante  $\mu_F$ .*

*Démonstration.*  $F$  étant fini, pour montrer que  $(X_n)$  admet une unique mesure stationnaire, d'après le lemme précédent, il suffit de montrer que la chaîne est irréductible.

On souhaite prouver que pour tout pont, il existe une suite de *corner flips* qui permet d'atteindre l'enveloppe supérieure, avec probabilité non nulle. En effet, il suffit pour cela de retourner le coin vers le bas le plus à gauche et de répéter l'opération jusqu'à ce qu'il n'y ait plus de coins vers le bas. On aura alors atteint l'enveloppe supérieure. A chaque étape, la probabilité de tomber uniformément sur ce coin et de le retourner étant non nulle, on aura bien atteint l'enveloppe supérieure avec probabilité non nulle.

Ainsi, en partant de tout pont  $\mathbf{x}, \mathbf{y} \in F$ , la probabilité d'obtenir l'enveloppe supérieure est non nulle. On en déduit qu'il existe un chemin qui part du pont  $\mathbf{x}$  et arrive au pont  $\mathbf{y}$  avec probabilité non nulle. Cela conclut la preuve de l'irréductibilité de la chaîne  $(X_n)$ , qui implique l'existence d'une unique probabilité stationnaire pour  $(X_n)$ .

On montre ensuite que  $\mu_F$  est bien  $Q_{1/2}$ -invariante. On a que notre mesure cible  $\mu_F$  est  $Q_{1/2}$ -réversible. Pour tout  $\mathbf{x}, \mathbf{y} \in F$ ,  $\mu_F$  étant constante sur  $F$  et par symétrie de  $Q_{1/2}$ ,

$$\sum_{\mathbf{x} \in F} \mu_F(\mathbf{x}) Q_{1/2}(\mathbf{x}, \mathbf{y}) = \sum_{\mathbf{x} \in F} \mu_F(\mathbf{y}) Q_{1/2}(\mathbf{y}, \mathbf{x}).$$

On montre maintenant que la  $Q_{1/2}$ -réversibilité de  $\mu_F$  implique sa  $Q_{1/2}$ -stationnarité. Pour tout  $\mathbf{y} \in F$ ,

$$\mu_F Q_{1/2}(\mathbf{y}) = \sum_{\mathbf{x} \in F} \mu_F(\mathbf{x}) Q_{1/2}(\mathbf{x}, \mathbf{y}) = \sum_{\mathbf{x} \in F} \mu_F(\mathbf{y}) Q_{1/2}(\mathbf{y}, \mathbf{x}) = \mu_F(\mathbf{y}) \sum_{\mathbf{x} \in F} Q_{1/2}(\mathbf{y}, \mathbf{x}) = \mu_F(\mathbf{y})$$

car  $\mu_F$  est  $Q_{1/2}$ -réversible et  $Q_{1/2}$  est stochastique. Donc  $Q_{1/2}$  admet bien  $\mu_F$  comme mesure stationnaire, ce qui conclut la preuve.

$(X_n)$  admet alors bien  $\mu_F$  comme unique mesure stationnaire. □

Il est important de préciser que nous obtenons ici seulement un résultat asymptotique et qu'il ne s'agit pas d'une méthode de simulation exacte. Nous avons représenté ci-dessous, les répartitions de sortie de 10 000 ponts de taille  $N = 6$

pour différents nombres d'itérations de la récurrence.

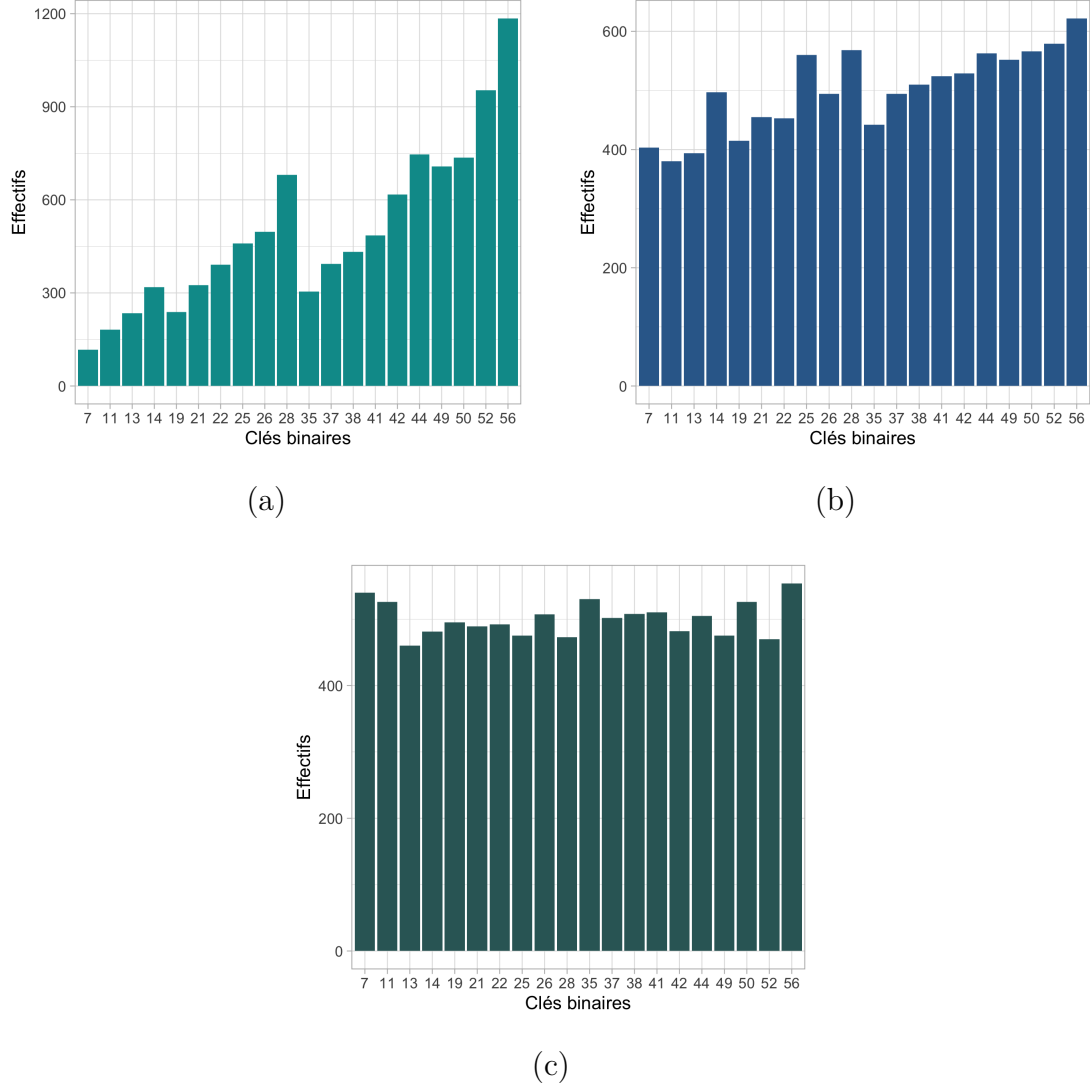


FIGURE 3 – Distribution empirique de 10000 ponts de longueurs  $N = 6$  obtenus après 50 (a), 100 (b) et 200 (c) itérations de notre dynamique *corner flips*.

La distribution semble se rapprocher de la répartition d'une loi uniforme lorsque le nombre d'itérations augmente. Estimons le nombre d'itérations nécessaires pour considérer l'échantillon comme étant simulé suivant  $\mu_F$ . Pour cela, observons l'évolution des p-valeurs obtenues pour les tests d'uniformité du  $\chi^2$  en fonction du nombre d'itérations de la récurrence aléatoire.



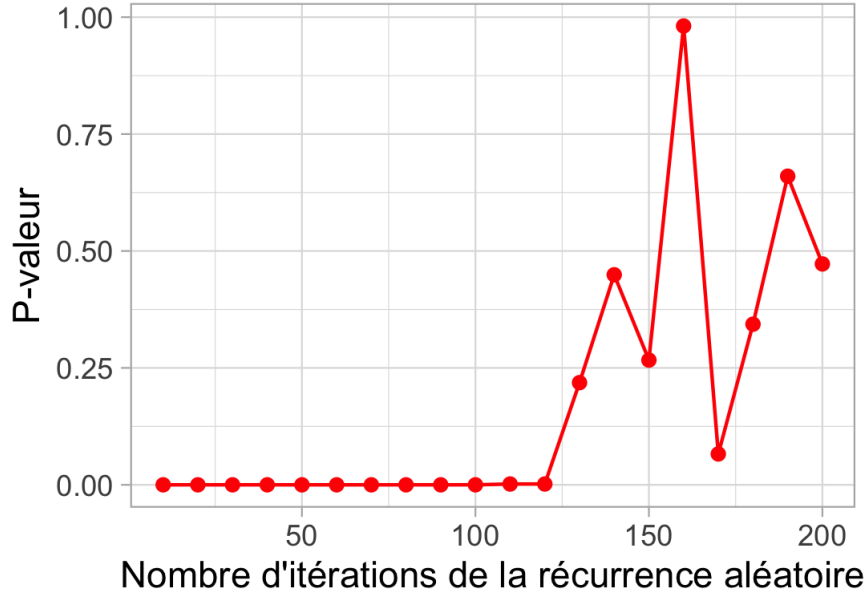


FIGURE 4 – Évolution des p-valeurs obtenues pour les tests d'uniformité du  $\chi^2$  en fonction du nombre d'itérations de la récurrence aléatoire appliquée à 10 000 chaînes de taille  $N = 6$ .

Pour  $N = 6$ , à partir de 130 itérations de la récurrence (3), les p-valeurs du test restent supérieures à 5%. On peut donc considérer que cette dynamique simule selon  $\mu_F$  à partir d'un certain nombre d'itérations de la récurrence. Cependant, elle ne permet pas de simuler suivant  $\mu_F$  de manière exacte.

## 4.2 Généralisation à une dynamique non-symétrique

Une question pertinente serait alors de savoir si  $\mu_F$  est toujours une mesure invariante lorsque la dynamique « *corner flips* » favorise un pivot dans un sens plutôt que dans l'autre. Cela correspond à la récurrence aléatoire (3) dans laquelle la suite de variables aléatoires  $(B_n)$  suit une Bernoulli de paramètre  $p \neq \frac{1}{2}$ . Exprimons la matrice de transition de cette nouvelle chaîne. Pour tout  $\mathbf{x}, \mathbf{y} \in F$  tels que  $\mathbf{x} \neq \mathbf{y}$ , on a

$$Q_p(\mathbf{x}, \mathbf{y}) = \begin{cases} 0 & \text{si } \forall i \in \llbracket 1, N-1 \rrbracket, \mathbf{y} \neq T_d(\mathbf{x}, i) \text{ et } \mathbf{y} \neq T_u(\mathbf{x}, i) \\ \frac{1-p}{N-1} & \text{si } \exists i \in \llbracket 1, N-1 \rrbracket \text{ tel que } \mathbf{y} = T_u(\mathbf{x}, i) \\ \frac{p}{N-1} & \text{si } \exists i \in \llbracket 1, N-1 \rrbracket \text{ tel que } \mathbf{y} = T_d(\mathbf{x}, i) \end{cases}$$

et pour tout  $\mathbf{x} \in F$ , on a

$$Q_p(\mathbf{x}, \mathbf{x}) = 1 - \frac{C^d(\mathbf{x}) + C^u(\mathbf{x})}{N-1} + p \frac{C^d(\mathbf{x})}{N-1} + (1-p) \frac{C^u(\mathbf{x})}{N-1},$$

en notant  $C^d(\mathbf{x})$  et  $C^u(\mathbf{x})$  respectivement le nombre de coins vers le bas et vers le haut du pont  $\mathbf{x}$ .

Ce noyau de transition n'est pas symétrique, contrairement à  $Q_{1/2}$ . Le lemme 4.2 de la section précédente ne s'applique plus et on ne peut donc pas affirmer que la dynamique ainsi définie fournit une chaîne de Markov de mesure stationnaire notre loi cible  $\mu_F$ . En revanche, on peut créer un nouveau noyau de transition pour lequel la chaîne associée admet pour mesure invariante  $\mu_F$ , qu'importe la valeur de  $p$ . Cela est possible grâce au théorème suivant, et ce à partir d'une chaîne qui vérifie des hypothèses plus faibles que dans la section précédente. Ce théorème provient de l'article [CM21].

**Théorème 4.1** (Metropolis-Hastings). *Soit  $Q : A \times A \rightarrow [0, 1]$  un noyau de transition irréductible sur  $A$ , vérifiant pour tout  $\mathbf{x}, \mathbf{y} \in A$ ,*

$$Q(\mathbf{x}, \mathbf{y}) = 0 \text{ si et seulement si } Q(\mathbf{y}, \mathbf{x}) = 0.$$

*Alors  $P$  défini ci-dessous est un noyau de transition sur  $A$ , irréductible et de loi invariante réversible  $\mu_P$ . Pour tout  $\mathbf{x}, \mathbf{y} \in A$ ,*

$$P(\mathbf{x}, \mathbf{y}) = \begin{cases} Q(\mathbf{x}, \mathbf{y}) \rho(\mathbf{x}, \mathbf{y}) & \text{si } \mathbf{x} \neq \mathbf{y} \\ 1 - \sum_{\mathbf{z} \neq \mathbf{x}} P(\mathbf{x}, \mathbf{z}) & \text{si } \mathbf{x} = \mathbf{y} \end{cases}$$

où  $\rho$  est la fonction

$$\rho(\mathbf{x}, \mathbf{y}) = \min \left( 1, \frac{\mu_P(\mathbf{y}) Q(\mathbf{y}, \mathbf{x})}{\mu_P(\mathbf{x}) Q(\mathbf{x}, \mathbf{y})} \right) \mathbb{1}_{\{Q(\mathbf{x}, \mathbf{y}) > 0\}}.$$

*Démonstration.* On a  $0 \leq P(\mathbf{x}, \mathbf{y}) \leq Q(\mathbf{x}, \mathbf{y}) \leq 1$  pour tous  $\mathbf{x} \neq \mathbf{y}$  dans  $A$  car  $\rho \leq 1$ . De plus,

$$\sum_{\mathbf{y} \in A} P(\mathbf{x}, \mathbf{y}) = P(\mathbf{x}, \mathbf{x}) + \sum_{\mathbf{y} \neq \mathbf{x}} P(\mathbf{x}, \mathbf{y}) = 1 - \sum_{\mathbf{z} \neq \mathbf{x}} P(\mathbf{x}, \mathbf{z}) + \sum_{\mathbf{y} \neq \mathbf{x}} P(\mathbf{x}, \mathbf{y}) = 1.$$

Donc  $P$  est à valeurs dans  $[0, 1]$  et stochastique, donc est un noyau de transition. Comme  $\rho > 0$  et que  $Q$  est irréductible alors  $P$  est irréductible.

$P$  est irréductible sur  $A$  fini, il possède donc une loi invariante d'après le lemme 4.2. Montrons alors que, avec les notations du théorème,  $\mu_P$  est invariante pour  $P$ . La définition de  $\rho$  donne, pour tous  $\mathbf{x}, \mathbf{y} \in A$  tels que  $\mathbf{x} \neq \mathbf{y}$  et  $Q(\mathbf{x}, \mathbf{y}) \neq 0$ ,

$$\begin{aligned}\mu_P(\mathbf{x})P(\mathbf{x}, \mathbf{y}) &= \mu_P(\mathbf{x})Q(\mathbf{x}, \mathbf{y})\rho(\mathbf{x}, \mathbf{y}) \\ &= \mu_P(\mathbf{x})Q(\mathbf{x}, \mathbf{y})\frac{\mu_P(\mathbf{y})Q(\mathbf{y}, \mathbf{x})}{\mu_P(\mathbf{x})Q(\mathbf{x}, \mathbf{y})}\rho(\mathbf{y}, \mathbf{x}) \\ &= \mu_P(\mathbf{y})P(\mathbf{y}, \mathbf{x}).\end{aligned}$$

Ainsi  $\mu_P$  est réversible pour  $P$ , et c'est donc la loi invariante de  $P$ .  $\square$

**Algorithme.** Pour construire une chaîne avec un tel noyau  $P$ , cela se ramène au problème de la simulation de la loi discrète  $P(\mathbf{x}, \cdot)$  pour un  $\mathbf{x}$  quelconque. Pour ce faire, soit  $Y$  une variable aléatoire sur  $A$  de loi  $Q(\mathbf{x}, \cdot)$ , et  $U$  une variable aléatoire de loi uniforme sur  $[0, 1]$ , indépendante de  $Y$ . Soit  $Z$  la variable aléatoire définie par  $Z = Y$  si  $U < \rho(\mathbf{x}, Y)$  et  $Z = \mathbf{x}$  sinon. Alors  $Z \sim P(\mathbf{x}, \cdot)$  car pour tous  $\mathbf{y} \neq \mathbf{x}$ ,

$$\mathbb{P}(Z = \mathbf{y}) = \mathbb{P}(U < \rho(\mathbf{x}, Y), Y = \mathbf{y}) = \rho(\mathbf{x}, \mathbf{y})Q(\mathbf{x}, \mathbf{y}) = P(\mathbf{x}, \mathbf{y}).$$

Autrement dit la proposition d'un candidat  $Y$  de loi  $Q(\mathbf{x}, \cdot)$  est acceptée si, et seulement si  $U < \rho(\mathbf{x}, Y)$ . On dit que  $\rho$  est la fonction d'acceptation-rejet tandis que  $Q$  le noyau de proposition ou d'exploration. Le noyau  $Q$  doit être facile à simuler, ce qui est le cas ici. Cela revient à appliquer l'algorithme suivant :

**Données :** un pont  $\mathbf{x}_0$

**Résultat :** Nouvel état de la chaîne  $x_1$

```

 $u \leftarrow$  réalisation d'une  $\mathcal{U}([0; 1])$ ;
 $\mathbf{y} \leftarrow$  réalisation suivant  $Q(\mathbf{x}_0, \cdot)$ ;
 $\rho \leftarrow \min(1, \frac{\mu_P(\mathbf{y})Q(\mathbf{y}, \mathbf{x}_0)}{\mu_P(\mathbf{x}_0)Q(\mathbf{x}_0, \mathbf{y})})\mathbb{1}_{\{Q(\mathbf{x}_0, \mathbf{y}) > 0\}}$ ;
si  $u < \rho$  alors
    |  $x_1 \leftarrow \mathbf{y}$ ;
sinon
    |  $x_1 \leftarrow \mathbf{x}_0$ ;
fin
```

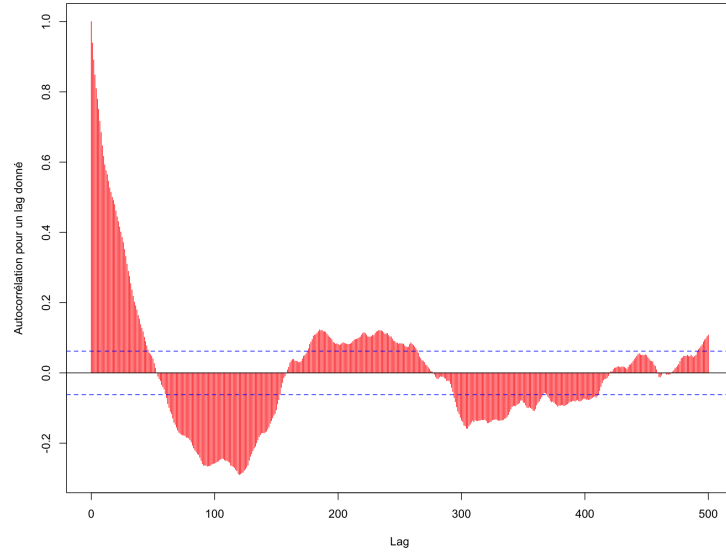
**Algorithme 4 :** Une itération de l'algorithme de Metropolis-Hastings

Le noyau de transition  $Q_p$  de la nouvelle dynamique, bien qu'il n'est pas symétrique, est irréductible et tel que pour tout  $\mathbf{x}, \mathbf{y} \in E$ ,

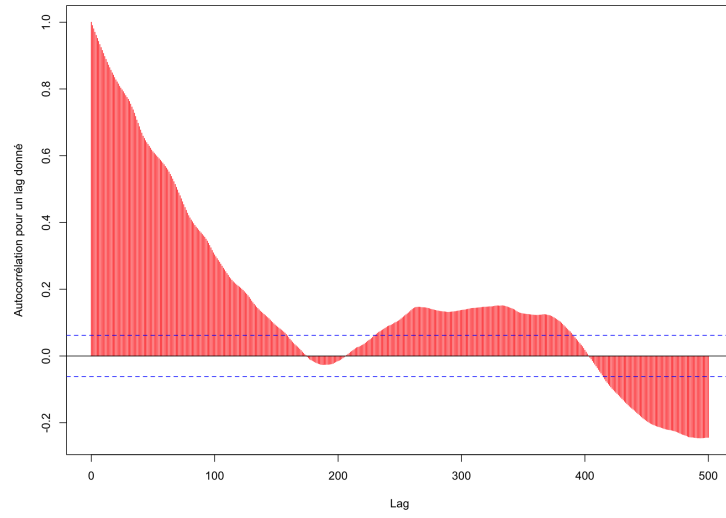
$$Q_p(\mathbf{x}, \mathbf{y}) = 0 \text{ si et seulement si } Q_p(\mathbf{y}, \mathbf{x}) = 0.$$

On peut donc lui appliquer le théorème de Metropolis-Hastings afin de créer une chaîne ayant un nouveau noyau de transition  $P_p = Q_p \rho$ , qui elle admet bien pour mesure invariante la mesure cible  $\mu_F$ . Bien qu'il s'agisse d'un résultat purement stationnaire, comme dans la Figure 4, on peut déterminer, pour un  $N$  donné, le nombre d'itérations à partir duquel on considère que cet algorithme simule suivant  $\mu_F$ .

Dans le cas où la Bernoulli est de paramètre  $p = \frac{1}{2}$ , Metropolis-Hastings ne modifie pas le noyau  $Q$ , car la fonction  $\rho \equiv 1$  par symétrie de  $Q$ . Dans ce cas,  $Q = P$  et donc l'algorithme Metropolis-Hastings ne rejette aucun candidat. La chaîne devrait alors se déplacer d'état en état plus rapidement qu'une chaîne définie avec une Bernoulli de paramètre  $p \neq 1/2$ . Une étude d'autocorrélation des états d'une chaîne à chaque itération permet de rendre compte de l'indépendance entre ses états à différentes distances. En effet, l'état à l'itération  $t$  peut être plus ou moins corrélé à celui de l'état à l'itération  $t + k$ . Les graphes d'évolution de l'autocorrélation en fonction d'un lag  $k$  pour deux chaînes définies avec des Bernoulli de paramètres différents, traduisent le déplacement de ces chaînes.



(a)



(b)

FIGURE 5 – Évolution de l'autocorrélation en fonction du décalage sur la chaîne générée par Metropolis-Hastings dont la Bernoulli est de paramètre  $p = 1/2$  (a),  $p = 1/6$  (b).

L'autocorrélation de la chaîne converge vers 0 plus rapidement pour  $p = 1/2$  que pour

$p = 1/6$ . Cela traduit le fait que la chaîne définie avec  $p = 1/2$  se déplace plus rapidement que l'autre, ce qui permet de converger vers l'état stationnaire plus rapidement. La dynamique définie avec  $p = 1/2$  sera préférée pour les méthodes à suivre.

## 5 Vers les méthodes de couplage

Dans la section précédente, nous avons construit différents processus markoviens ayant pour mesure stationnaire  $\mu_F$ . En utilisant cette dynamique, nous allons nous pencher sur une méthode de simulation exacte : la méthode dite de *coupling from the past*, qui permet de déterminer un temps aléatoire, au bout duquel la chaîne construite est un tirage exact suivant  $\mu_F$ . Avant de la présenter plus en détails, introduisons la notion de couplage, ou *coupling* en anglais. Pour cette section, nous nous baserons sur l'article [Vig06].

L'idée du couplage est simple : elle consiste à faire évoluer simultanément deux ponts à l'aide d'une même dynamique *corner flips*, c'est-à-dire basées sur le même aléa. Introduisons quelques notations. Soient :  $(U_r)_{r \in \mathbb{Z}}$  une suite de variables *i.i.d.* uniformes sur  $\llbracket 1, N-1 \rrbracket$ , et  $(B_r)_{r \in \mathbb{Z}}$  suite de variables *i.i.d.* suivant  $Ber(\frac{1}{2})$ , toutes deux indépendantes. Pour tout  $r \in \mathbb{Z}$ , et pour tout  $\mathbf{x} \in F$ , on note  $h_r$  la fonction

$$h_r(\mathbf{x}) := h(\mathbf{x}, U_{r+1}, B_{r+1}) = \sum_{k=1}^{N-1} \mathbb{1}_{\{U_{r+1}=k\}} [B_{r+1} T_d(\mathbf{x}, k) + (1 - B_{r+1}) T_u(\mathbf{x}, k)].$$

On notera également pour tout  $t_1 < t_2$  entiers,  $H_{t_1}^{t_2}$  la fonction

$$\begin{aligned} H_{t_1}^{t_2}(\mathbf{x}) &= (h_{t_2-1} \circ h_{t_2-2} \circ \dots \circ h_{t_1})(\mathbf{x}) \\ &= h_{t_2-1}(h_{t_2-2}(\dots h_{t_1}(\mathbf{x}) \dots)). \end{aligned}$$

Donc  $h_r$  décrit une étape de notre dynamique, et  $H_{t_1}^{t_2}$  décrit l'évolution de notre dynamique du temps  $t_1$  au temps  $t_2$ . On peut maintenant introduire proprement la notions de couplage.

**Définition 5.1.** *Un processus de couplage d'initialisation  $(\mathbf{x}, \mathbf{y}) \in F^2$  est un processus où l'on applique la même dynamique  $H_0^{C_{\mathbf{x}, \mathbf{y}}}$  aux deux ponts  $\mathbf{x}$  et  $\mathbf{y}$  simultanément, où  $C_{\mathbf{x}, \mathbf{y}}$  est la variable aléatoire appelée temps de coalescence du processus de couplage initialisé en  $(\mathbf{x}, \mathbf{y})$  définie par :*

$$C_{\mathbf{x}, \mathbf{y}} := \inf\{c \in \mathbb{R}_+^*, H_0^c(\mathbf{x}) = H_0^c(\mathbf{y})\}.$$

On admettra que pour toute initialisation  $(\mathbf{x}, \mathbf{y}) \in F^2$ , un tel processus admet un temps de coalescence fini, c'est-à-dire qu'il existe  $c \in \mathbb{R}_+^*$  tel que

$$H_0^c(\mathbf{x}) = H_0^c(\mathbf{y}).$$

Pour plus de détails sur la construction d'un tel processus markovien sur  $F^2$ , on peut se référer à [Sal22]. Nous pouvons nous poser la question suivante : le processus de couplage génère-t-il des simulations suivant  $\mu_F$  ? Pour cela, regardons la répartition des ponts obtenus à coalescence.

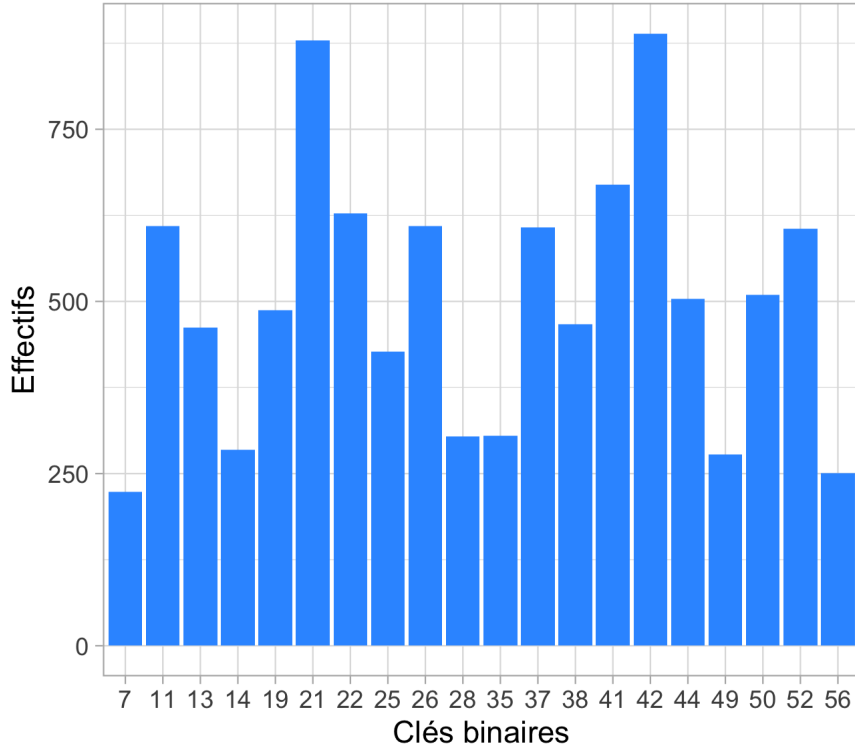


FIGURE 6 – Distribution de 10000 ponts de longueur  $N = 6$  générés par couplage.

Nous constatons encore une fois un défaut d'uniformité. Nous avons pu confirmer cela avec un test du  $\chi^2$  d'adéquation à  $\mu_F$ , de p-valeur de l'ordre de  $10^{-16}$ . Envisageons un autre procédé, appelé *coupling from the past*.

L'idée du *coupling from the past* est de partir d'un instant passé noté  $-K$ , et de faire évoluer simultanément deux ponts à l'aide de la même dynamique *corner flips*. Cependant contrairement au *coupling*, on ne s'arrêtera pas à la coalescence des chaînes mais au temps 0. Il faudra donc choisir  $-K$  de sorte à ce que tout processus coalesce

avant 0. En pratique, cela revient à continuer le processus même après coalescence, et ce pendant un temps aléatoire fini. Notons que, après coalescence, les deux ponts évolueront de la même manière et resteront confondus. Pour justifier le recours à cette méthode, il nous faut montrer que les ponts obtenus en temps 0 via ce processus seront des tirages selon  $\mu_F$ .

**Théorème 5.1** (Coupling from the past). *On note  $K := \inf\{K \in \mathbb{N}, |H_{-K}^0(F)| = 1\}$ , c'est à dire le plus petit horizon passé tel que tous les ponts ont coalescé avant 0. Alors  $H_{-K}^0(F)$  a la même loi que  $\mu_F$ .*

*Démonstration.* D'après le lemme 3.1, on sait qu'une telle chaîne admet une unique mesure stationnaire,  $\mu_F$ . On souhaite d'abord montrer que cette mesure stationnaire est également atteinte dans le passé. Comme les couples  $(U_r, B_r)_{r \in ([0, t])}$  et  $(U_r, B_r)_{r \in ([-t, 0])}$  sont *i.i.d.* et l'intervalle d'évolution de la dynamique de couplage est le même, on a pour tout  $t > 0$  et tout  $x, y \in \Omega$ ,

$$\mathbb{P}(H_0^t(\mathbf{x}) = \mathbf{y}) = Q^{t-0}(\mathbf{x}, \mathbf{y}) = Q^{0-(-t)}(\mathbf{x}, \mathbf{y}) = \mathbb{P}(H_{-t}^0(\mathbf{x}) = \mathbf{y}).$$

donc  $H_0^t(\mathbf{x})$  et  $H_{-t}^0(\mathbf{x})$  sont de lois identiques. De plus, on a pour tout  $\mathbf{x}, \mathbf{y} \in F$ ,

$$\lim_{t \rightarrow \infty} \mathbb{P}(H_{-t}^0(\mathbf{x}) = \mathbf{y}) = \lim_{t \rightarrow \infty} \mathbb{P}(H_0^t(\mathbf{x}) = \mathbf{y}) = \mu_F(\mathbf{y}).$$

La stationnarité est donc aussi atteinte dans le passé. Montrons désormais qu'il suffit d'échantillonner en partant de  $-K$  pour tout de même avoir une échantillon suivant  $\mu_F$  la mesure stationnaire. Pour tout  $t_1 < t$ , on remarque que

$$H_{-t}^0 = H_{-t_1}^0 \circ H_{-t}^{-t_1}.$$

Donc, comme  $H_{-K}^0$  est une fonction constante par définition de  $K$  à  $(U_r, B_r)_{r \in ([-K, 0])}$  fixés, pour tout  $t > K$  et tout  $x \in F$ ,

$$H_{-t}^0(\mathbf{x}) = H_{-K}^0(\mathbf{x}) \circ H_{-t}^{-K}(\mathbf{x}) = H_{-K}^0(\mathbf{x}).$$

C'est donc pourquoi échantillonner suivant  $H_{-\infty}^0(\mathbf{x})$  revient à échantillonner suivant  $H_{-K}^0(\mathbf{x})$ , ce qui prouve ce théorème.  $\square$

Cependant pour mettre ce théorème en pratique, il faut pouvoir estimer  $-K$ . Pour cela, il faudrait lancer le processus de couplage pour toute initialisation  $(x, y) \in F^2$  pour estimer le temps de coalescence moyen  $K$ . Pour cela, il est nécessaire de pouvoir construire et énumérer l'ensemble des ponts de  $F$  ce qui, comme mentionné dans l'introduction,



n'est pas le cas pour  $N$  grand. Toutefois, il est possible de contourner ce problème en s'appuyant sur un résultat, appelé *sandwiching*, tiré du livre [Hä02], qui repose sur une certaine monotonie de notre dynamique. D'abord, on construit une relation d'ordre partielle, plutôt naturelle, sur  $F$ .

**Définition 5.2.** Soit  $\mathbf{x}, \mathbf{y} \in F$ .

$$\mathbf{x} \leq \mathbf{y} \text{ lorsque } \forall i \in \llbracket 1, N-1 \rrbracket, x_i \leq y_i.$$

Montrons maintenant la monotonie de notre dynamique *corner flips*, en s'appuyant sur cette relation d'ordre.

**Théorème 5.2** (Monotonie). Soit  $(\mathbf{x}, \mathbf{y}) \in F^2$  tel que  $\mathbf{x} \leq \mathbf{y}$ . Alors  $h_r(\mathbf{x}) \leq h_r(\mathbf{y})$ .

*Démonstration.* Si  $\mathbf{x} = \mathbf{y}$  alors  $h_r(\mathbf{x}) = h_r(\mathbf{y})$  par composition. Sinon, il existe  $i \in \llbracket 1; N \rrbracket$  tel que  $x_i < y_i$ . Notons  $\Delta_i := y_i - x_i > 0$ . Si  $\Delta_i > 2$ ,  $\Delta_i \geq 4$ . Selon différents cas que nous éviterons d'énumérer par souci de clarté, on sait que pour tout  $\mathbf{x} \in F$ ,  $h_r(\mathbf{x})$  peut prendre les valeurs  $\mathbf{x}, T_u(\mathbf{x}, u)$ , ou  $T_d(\mathbf{x}, u)$ . Ici, dans le pire des cas, on a quand même  $T_u(\mathbf{x}, i) \leq T_d(\mathbf{y}, i)$ . Si  $\Delta_i = 2$ , le seul cas critique est le cas où  $\mathbf{x}$  a un coin vers le bas en  $i$  et  $\mathbf{y}$  un coin vers le haut en  $i$ , c'est-à-dire  $x_{i-1} = x_{i+1} = y_{i-1} = y_{i+1}$ , et où  $u_{r+1} = i$ . Par définition de  $h_r$ , si  $b_{r+1} = 0$ , on a

$$h_r(\mathbf{x}) = T_u(\mathbf{x}, i) \text{ et } h_r(\mathbf{y}) = \mathbf{y}.$$

On a donc  $h_r(\mathbf{x})_i = y_i$ , donc l'ordre est conservé. De même, si  $b_{r+1} = 1$ , on a

$$h_r(\mathbf{x}) = \mathbf{x} \text{ et } h_r(\mathbf{y}) = T_d(\mathbf{y}, i).$$

Alors l'ordre est aussi conservé car  $h_r(\mathbf{y})_i = x_i$ . □

Donc la fonction de récurrence conserve l'ordre de deux chaînes (si il y en a un). Nous pouvons enfin énoncer notre résultat permettant de mettre en pratique la méthode du *coupling from the past*. Notons  $\mathbf{x}_0$  l'enveloppe supérieure de  $F$  et  $\mathbf{y}_0$  son enveloppe inférieure.

**Proposition 5.1** (Sandwiching). Tous processus de *coupling from the past* initialisés en  $(\mathbf{x}, \mathbf{y})$  coalescent avant tous ceux initialisés en  $(\mathbf{x}_0, \mathbf{y}_0)$ . C'est-à-dire que pour tout  $\mathbf{x}, \mathbf{y} \in F$ ,

$$C_{\mathbf{x}, \mathbf{y}} \leq C_{\mathbf{x}_0, \mathbf{y}_0}.$$

*Démonstration.* Par monotonie, pour tout  $\mathbf{x} \leq x_0$ , tout  $\mathbf{y} \geq y_0$  et tout  $t \geq -K$ ,

$$H_{-K}^0(\mathbf{x}) \leq H_{-K}^0(\mathbf{x}_0) \text{ et } H_{-K}^0(\mathbf{y}) \geq H_{-K}^0(\mathbf{y}_0).$$

De plus, par définition de  $C_{x_0, y_0}$ , on a

$$H_{-K}^{C_{x_0, y_0}}(x_0) = H_{-K}^{C_{x_0, y_0}}(y_0).$$

Donc nécessairement, on aura coalescence du processus initialisé en  $(\mathbf{x}, \mathbf{y})$  avant l'instant  $C_{x_0, y_0}$ , d'où le résultat.  $\square$

Le résultat de *sandwiching* permet d'améliorer l'efficacité de l'algorithme, en continuant de simuler exactement suivant la loi cible. En effet, il nous permet d'estimer  $K$  en observant seulement la moyenne des temps de coalescence pour des processus initialisés en  $(\mathbf{x}_0, \mathbf{y}_0)$ , ce qui est possible car les enveloppes sont connues. Intéressons nous à l'évolution de l'estimation de  $K$  en fonction de  $N$ .

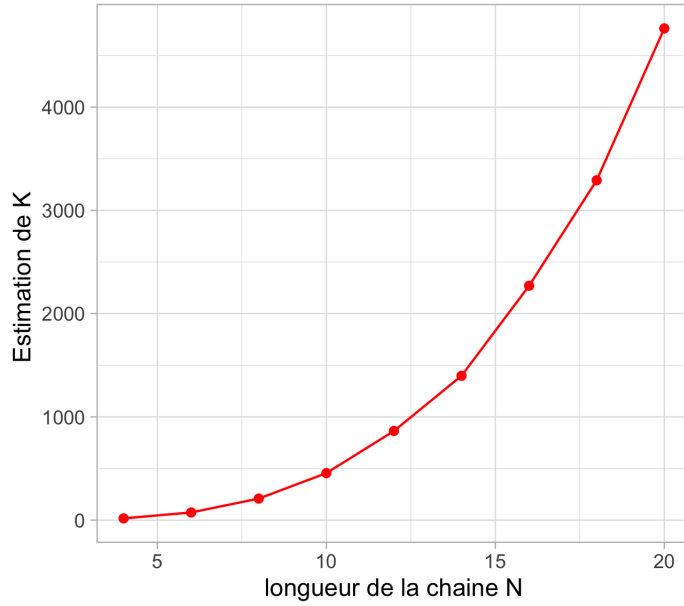


FIGURE 7 – Estimation de  $K$  avec un échantillon de 10000 ponts, en fonction de la longueur  $N$  des initialisations.

Plus la longueur de nos ponts  $N$  est grande, plus il faut remonter loin dans le passé afin d'atteindre la coalescence des enveloppes avant 0. Cependant, en partant de l'estimation de  $-K$ , il est possible que certaines chaînes n'aient pas encore coalescé à l'instant 0. Dans ce cas, il nous faudra partir de  $-K' := -K - \delta$ , où  $\delta$  est un pas à choisir. Prendre

$\delta$  « trop grand » augmenterait les temps de calcul mais assure d'atteindre  $\mu_F$  en 0. En prenant  $\delta$  « trop petit », il risque de ne pas y avoir coalescence des enveloppes avant 0 et il faudra alors augmenter le pas tant que cette condition n'est pas remplie.

L'avantage de la méthode de *coupling from the past*, est qu'elle permet de simuler exactement suivant  $\mu_F$ , tout comme la méthode de tirages successifs de la section 2, et celle d'acceptation-rejet construite dans la section 3. Nous pouvons finalement comparer ces méthodes.

## 6 Conclusion

La Figure 7 permet de rendre compte de la croissance exponentielle, en fonction de  $N$ , du nombre d'itérations nécessaires à la simulation d'un pont suivant  $\mu_F$  avec la méthode de *coupling from the past*. Tandis que la complexité de la méthode d'acceptation-rejet est d'ordre  $\mathcal{O}(\sqrt{N})$ . Cependant, le nombre d'itérations de ces méthodes ne suffisent pas à les comparer : d'autres aspects sont également à prendre en compte. Le nombre de variables aléatoires simulées dans un algorithme peut être très coûteux. Par exemple, la méthode du rejet demande de simuler  $N$  uniformes sur  $\{-1, 1\}$ , tandis que le *coupling from the past* nécessite de simuler une Bernoulli et une uniforme sur  $\llbracket 2, N - 1 \rrbracket$ . Le langage de programmation utilisé et l'utilisation de ses spécificités permet de réduire considérablement le coût de stockage, d'affectation et de calcul.

## Références

- [Bak02] Dominique Bakry. Chaînes de markov dénombrables. Modèles stochastiques, 2002.
- [CM21] Djalil Chafai and Florent Malrieu. Recueil de modèles aléatoires. HAL, 2021.
- [GPM15] Alban Goupil-Papa and Corentin Morandea. La marche de sinaï. Une marche aléatoire en milieu aléatoire, 2015.
- [Hä02] Olle Häggström. Finite markov chain and algorithmic applications. Cambridge University Press, 2002.
- [Sal22] Justin Salez. Coalescence times. Mixing times of Markov chains, 2022.
- [Sto23] Julien Stoeur. Simulation de variables aléatoires. Méthodes de Monte Carlo, 2023.
- [Vig06] Eric Vigoda. Markov chain monte carlo methods. Georgia Tech, 2006.