

---

**MESURE DE GIBBS  
ET  
ALGORITHMES MCMC**

---

**Auteurs :**

Maria BASTIANI  
Noémie BERGUES  
Edimah SONGO

**Encadrant :**

Julien Stoehr

## **Remerciements**

Nous tenons à remercier Monsieur Julien Stoehr, notre encadrant, pour sa patience, sa disponibilité et surtout ses précieux conseils qui nous ont permis d'aiguiser notre réflexion et de nous guider dans la rédaction de ce mémoire tout au long de l'année.

# Table des matières

<b>Introduction</b>	<b>4</b>
<b>1 Mesures de Gibbs</b>	<b>5</b>
1.1 Définition . . . . .	5
1.2 Problématiques . . . . .	8
<b>2 Simulation suivant la mesure de Gibbs</b>	<b>10</b>
2.1 Rappels sur les chaînes de Markov . . . . .	11
2.2 Algorithmes MCMC . . . . .	15
2.2.1 Algorithme de Metropolis-Hastings . . . . .	15
2.2.2 Algorithme de Gibbs . . . . .	20
2.2.3 Algorithme du recuit simulé . . . . .	22
2.3 Algorithme de Propp-Wilson . . . . .	25
2.3.1 Coupling from the past . . . . .	25
2.3.2 La méthode du sandwich . . . . .	27
2.4 Comparaison des méthodes . . . . .	30
<b>3 Inférence du paramètre de la mesure de Gibbs</b>	<b>32</b>
3.1 Estimateur du maximum de vraisemblance . . . . .	32
3.2 Estimation de la loi <i>a posteriori</i> . . . . .	34
<b>Conclusion</b>	<b>38</b>
<b>Bibliographie</b>	<b>39</b>
<b>Annexe</b>	<b>40</b>

# Introduction

La mesure de Gibbs est une mesure de probabilité définie sur un ensemble fini de grand cardinal. Elle tient son nom de Josiah Willard Gibbs, père fondateur de la mécanique statistique dont l'objectif est d'expliquer les lois de la thermodynamique à l'aide de modèles statistiques pour des grands ensembles de particules. Cette mesure permet de modéliser des systèmes complexes incluant par exemple une structure de dépendance spatiale. Dans ce cas, la mesure permet de spécifier le comportement global du système à partir de caractéristiques locales. Il est ainsi naturel de retrouver cette mesure en physique statistique. Des exemples classiques sont les modèles d'Ising et de Potts qui décrivent l'évolution à l'échelle macroscopique d'un ensemble de particules à partir des interactions entre les paires de particules. Mais cette mesure se retrouve également dans d'autres domaines d'application comme l'écologie, le traitement d'image ou la chimie.

Bien qu'utiles d'un point de vue modélisation, les mesures de Gibbs présentent néanmoins une spécificité rendant difficile les approches probabilistes et statistiques. En effet, une mesure de Gibbs forme une famille exponentielle dont la fonction de partition est difficile si ce n'est impossible à calculer. Ainsi on ne dispose pas d'une forme analytique pour la densité ou la vraisemblance qui font partie intégrante de ces approches et il est nécessaire de mettre en place des solutions adaptées.

Dans ce mémoire, nous nous intéressons à deux questions fondamentales. La première est celle de la simulation suivant une mesure de Gibbs. Tout d'abord, les méthodes de Monte-Carlo par chaîne de Markov (MCMC) et le résultat fondateur énoncé par Metropolis *et al.* (1953) apportent une réponse à cette problématique. L'idée générale est de construire une chaîne de Markov dont la loi de probabilité invariante est la mesure de Gibbs souhaitée. Ces méthodes ne permettent néanmoins pas de faire de la simulation exacte. Propp et Wilson (1996) ont proposé une solution basée sur des techniques de couplage et permettant de lever cette limitation lorsque le noyau de transition de la chaîne vérifie une propriété de monotonie. La seconde question est celle de l'estimation du paramètre d'une mesure de Gibbs. Que ce soit dans un cadre fréquentiste ou bayésien, la difficulté intrinsèque vient de l'absence de forme explicite pour la vraisemblance du modèle. Cette difficulté peut néanmoins être contournée dès lors que l'on sait simuler suivant la mesure. L'estimateur du maximum de vraisemblance ne peut donc par exemple pas être déterminé directement. Le gradient peut néanmoins être estimé par des méthodes de Monte-Carlo et le problème d'optimisation se résout alors à l'aide de l'algorithme de Robbins et Monro (1951). Dans le paradigme bayésien, les estimateurs sont usuellement construits via la distribution empirique d'un échantillon produit par un algorithme MCMC. Chaque itération inclut en général une étape d'acceptation rejet nécessitant de calculer un rapport de vraisemblance. Cette étape étant impossible pour une mesure de Gibbs, Murray *et al.* (2012) puis Alquier *et al.* (2016) ont proposé de remplacer, là encore, ce rapport par un estimateur de Monte-Carlo.

Le mémoire s'organise en trois chapitres. Le Chapitre 1 introduit la mesure de Gibbs et les problématiques spécifiques nous intéressant. Dans le Chapitre 2, nous présentons les algorithmes MCMC et l'algorithme de couplage de Propp-Wilson permettant de simuler suivant les mesures considérées. Enfin, dans le Chapitre 3 nous présentons comment ces algorithmes de simulation permettent de construire des estimateurs fréquentistes ou bayésiens du paramètre de la mesure.

# 1

## Mesures de Gibbs

### 1.1 Définition

La **mesure de Gibbs** (Chafai et Malrieu (2018)) est une mesure de probabilité définie sur un ensemble  $E$  fini, usuellement de très grand cardinal. C'est un exemple particulier de famille exponentielle régulière de paramètre  $\beta \in \mathbb{R}^d$  et de statistique naturelle  $H : E \rightarrow \mathbb{R}^d$ , que l'on appellera aussi *fonction énergie* ou *hamiltonien*. La mesure de Gibbs se définit comme suit pour  $x \in E$  :

$$\mu_\beta(x) = \frac{1}{Z_\beta} e^{-\langle \beta, H(x) \rangle}, \quad \text{où} \quad Z_\beta = \sum_{y \in E} e^{-\langle \beta, H(y) \rangle} \text{ est la fonction de partition.}$$

Dans le cadre de ce mémoire nous considérerons des variables aléatoires  $X = (X_1, \dots, X_N)$  distribuées suivant une mesure de Gibbs avec un paramètre réel ( $d = 1$ ). Lorsque  $\beta = 0$ , la mesure de Gibbs est la mesure uniforme sur l'ensemble  $E$ . *A contrario*, plus  $\beta$  est grand plus la mesure charge des configurations particulières. On note  $h_-$  le minimum de la fonction  $H$  sur  $E$  et on définit  $M = \{x \in E, H(x) = h_-\}$  l'ensemble des configurations d'énergie minimale. On remarque que pour tout  $x \in E$  :

$$\mu_\beta(x) = \frac{e^{-\beta(H(x)-h_-)}}{\text{card}(M) + \sum_{y \notin M} e^{-\beta(H(y)-h_-)}} \xrightarrow{\beta \rightarrow +\infty} \begin{cases} \text{card}(M)^{-1}, & \text{si } x \in M, \\ 0 & \text{sinon.} \end{cases} \quad (1.1)$$

Donc  $\mu_\beta$  converge vers la mesure uniforme sur  $M$  lorsque  $\beta \rightarrow +\infty$ .

Dans la suite, nous nous intéresserons plus particulièrement à des modèles de dépendance spatiale, c'est-à-dire au cas où un graphe non-orienté induit une topologie sur les composantes de  $X$ . On utilisera le formalisme suivant :

**Définition 1** (Graphe non-orienté). *Un graphe non-orienté  $G$  est un schéma constitué d'un ensemble  $S$  de points appelés sommets (ou noeuds), et d'un ensemble  $A$  de lignes entre sommets appelées arêtes. On note un tel graphe  $G = (S, A)$ . On notera également  $V$  l'ensemble des valeurs pouvant être prises par les sommets du graphe.*

**Définition 2** (Sommets adjacents). *On dit que deux sommets  $i, j \in S$  sont adjacents s'ils sont reliés par une arête. On note  $i \sim j$  la relation d'adjacence entre les sommets  $i$  et  $j$ , et  $v(i) = \{j \in S \mid (i, j) \in A\}$  l'ensemble des voisins ou sommets adjacents de  $i$ .*

**Définition 3** (Clique). *Une clique est un sous-ensemble de sommets du graphe, adjacents deux à deux. On note  $\mathcal{C}$  l'ensemble des cliques d'un graphe  $G$ .*

Pour  $(X_i)_{i \in S}$  la fonction énergie se décomposera comme une somme sur des cliques du graphe :

$$H(x) = \sum_{c \in \mathcal{C}} H_c(x_c).$$

Dans ce mémoire, on considère  $S = \llbracket 1, N \rrbracket$ ,  $N \in \mathbb{N}^*$  et on se concentre sur deux types d'hamiltonien : l'un dont seule l'énergie sur les noeuds est non nulle et l'autre où seule l'énergie sur les arêtes est non nulle. Autrement dit :

$$H(x) = \sum_{i=1}^N H_i(x_i) \quad \text{ou} \quad H(x) = \sum_{i \sim j} H_{ij}(x_i, x_j),$$

où  $\sum_{i \sim j}$  représente la somme sur toutes les arêtes du graphe.

## Modèle du composé chimique

Cette mesure de Gibbs modélise la répartition, au sein d'un composé chimique, de particules chargées positivement ou négativement.

Chaque sommet  $i \in S$  est soit vierge et vaut 0, soit occupé par une particule chimique de type  $+1$  ou  $-1$ . La répulsion entre les composés se traduit par la contrainte que deux particules voisines ne doivent pas être de charges opposées. L'ensemble des configurations est le suivant :

$$E = \{x \in \{-1, 0, 1\}^N \mid \text{tel que si : } i \sim j \text{ alors } x_i x_j \neq -1\}.$$

Le hamiltonien correspond au nombre de particules chargées présentes dans le composé. Pour  $x \in E$  :

$$H(x) = \sum_{i=1}^N |x_i|.$$

**Exemple 1.** Soient  $X = (X_1, X_2, X_3, X_4)$  et  $G = (\llbracket 1, 4 \rrbracket, \{(1, 2), (2, 3), (2, 4), (3, 4)\})$ .

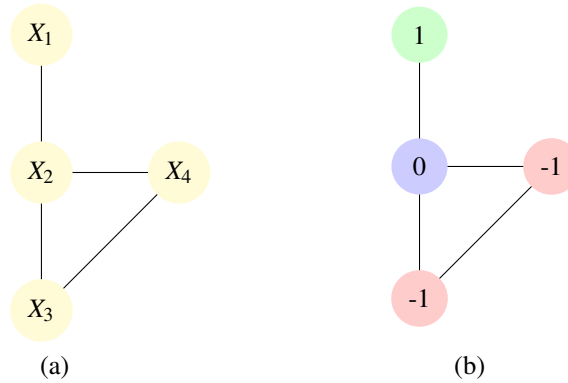


FIGURE 1.1 – (a) Représentation graphique de  $X$ . (b) Exemple de configuration avec  $H(x) = 3$ .

**Remarque 1.** La configuration de  $E$  d'énergie minimale est le graphe dont la valeur de tous les sommets est égale à 0.

## Modèle d'Ising

Le modèle d'Ising est un modèle de physique statistique utilisé pour représenter le comportement ferromagnétique de matériaux.

Les variables aléatoires  $X_i$ ,  $i \in S$  représentent le spin des atomes, i.e., elles sont à valeurs dans  $\{-1, 1\}$ . L'ensemble des configurations est le suivant :

$$E = \{-1, 1\}^N.$$

Le hamiltonien associé à ce modèle est défini pour  $x \in E$  :

$$H(x) = - \sum_{i \sim j} x_i x_j.$$

**Exemple 2.** Soient  $X = (X_1, X_2, X_3, X_4, X_5)$  et  $G = ([1, 5], \{(1, 2), (1, 3), (2, 3), (2, 4), (3, 5), (4, 5)\})$ .

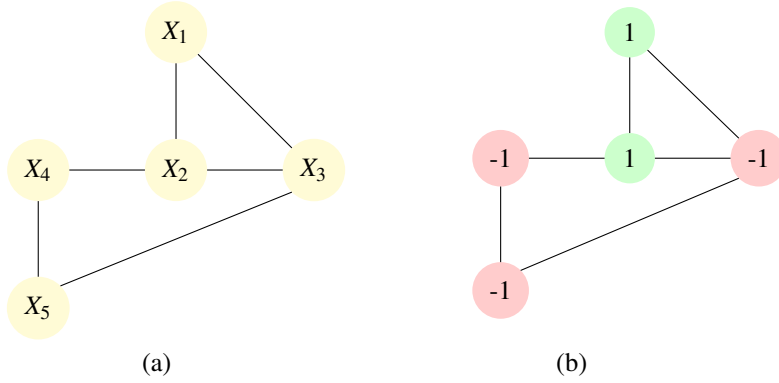


FIGURE 1.2 – (a) Représentation graphique de  $X$ . (b) Exemple de configuration avec  $H(x) = 0$ .

**Remarque 2.** Les configurations de  $E$  d'énergie minimale sont celles pour lesquelles un seul type de spin est présent, i.e., tous les sommets valent 1 ou  $-1$ .

## Modèle de Potts

Le modèle de Potts est une généralisation du modèle d'Ising dans lequel les variables aléatoires  $X_i$ ,  $i \in S$ , sont à valeurs dans  $[1, K]$ ,  $K \in \mathbb{N}^*$ . L'ensemble des configurations est le suivant :

$$E = [1, K]^N.$$

Le hamiltonien est fonction du nombre de paires homogènes dans le graphe. On a pour  $x \in E$  :

$$H(x) = - \sum_{i \sim j} \mathbb{1}_{\{x_i = x_j\}}.$$

**Exemple 3.** Soient  $K = 3$ ,  $X = (X_1, X_2, X_3, X_4, X_5, X_6, X_7)$  et  $G = ([1, 7], \{(1, 2), (2, 4), (2, 5), (3, 4), (3, 6), (3, 7), (5, 6), (6, 7)\})$ .

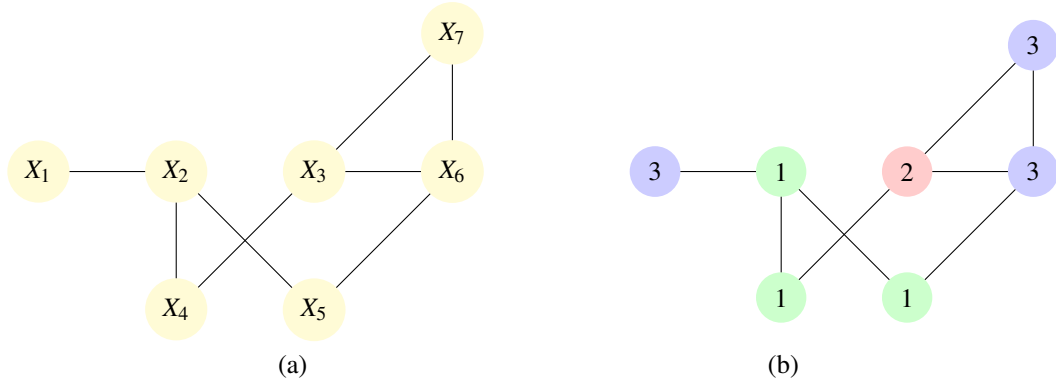


FIGURE 1.3 – (a) Représentation graphique de  $X$ . (b) Exemple de configuration avec  $H(x) = -3$ .

### Remarque 3.

- Il est possible de se ramener au modèle d'Ising en reparamétrant le modèle de Potts pour  $K = 2$ . En effet, considérons  $x$  une réalisation du modèle de Potts avec  $K = 2$  et  $\tilde{x}$  une réalisation du modèle d'Ising, on a alors pour tout site  $(i, j) \in A$ ,  $\tilde{x}_i \tilde{x}_j = 2\mathbb{1}_{\{x_i = x_j\}} - 1$ .
- Les configurations de  $E$  d'énergie minimale sont les graphes dont la valeur de tous les sommets est identique et égale à un entier  $k \in \llbracket 1, K \rrbracket$ .

Dans les applications des Chapitres 2 et 3, on se restreint pour ces modèles au cas où  $G$  est une grille régulière de taille  $h \times w$  au 4 plus proches voisins, que l'on représentera sous la forme d'une matrice de même dimension (cf. Figure 1.4)

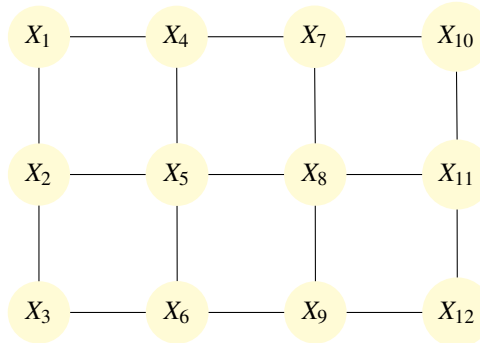


FIGURE 1.4 – Exemple d'une grille régulière de taille  $3 \times 4$

## 1.2 Problématiques

La fonction de partition n'admet pas toujours d'expression explicite. En effet, le calcul de  $Z_\beta$  nécessite d'être capable de sommer sur l'ensemble des éléments de  $E$ . Or, lorsque  $E$  est de grand cardinal, énumérer ces derniers est long, voire impossible.



**Exemple 4.** Dans le cadre du modèle de Potts, on a  $\text{card}(E) = K^N$ . Autrement dit, le cardinal de  $E$  croît exponentiellement en le nombre de sommets dans le graphe. Pour  $K = 5$  et  $N = 25$ , on a :

$$\text{card}(E) = 2,9\text{e}^{17}.$$

Sommer sur l'ensemble de ces configurations représente une difficulté combinatoire et n'est possible que pour de très petits graphes.

Sans accès à cette fonction de partition  $Z_\beta$ , la densité est connue à une constante près. Cela constitue un problème central que ce soit du point de vue de la simulation d'une mesure de Gibbs (cf. Chapitre 2), ou celui de l'inférence du paramètre  $\beta$  dans les paradigmes fréquentiste et bayésien (cf. Chapitre 3).

## 2

# Simulation suivant la mesure de Gibbs

Il existe différentes méthodes pour simuler suivant des densités connues à une constante près. L'une des plus simples à mettre en oeuvre est l'algorithme du rejet qui nécessite uniquement de savoir majorer uniformément la densité cible. Cette méthode est néanmoins coûteuse en temps de calcul. Regardons comment s'applique cette méthode pour la mesure de Gibbs modélisant le composé chimique. On définit l'ensemble  $\Omega = \{-1, 0, 1\}^N$ , on a donc  $E \subset \Omega$ .  $E$  étant de cardinal fini, on a pour tout  $x \in E$  :

$$e^{-\beta H(x)} \leq \max_{x \in E} (e^{-\beta H(x)}) =: m.$$

Ainsi on peut borner  $\mu_\beta$  comme suit :

$$\mu_\beta(x) \leq \frac{m \times \text{card}(\Omega)}{Z_\beta} \times \frac{1}{\text{card}(\Omega)} \mathbb{1}_{\{x \in \Omega\}}.$$

En tirant uniformément dans  $\Omega$ , la probabilité d'acceptation moyenne est

$$\frac{Z_\beta}{\text{card}(\Omega) \times m} = \frac{\sum_{x \in E} \exp\{-\beta H(x)\}}{\text{card}(\Omega) \times m} \leq \frac{\text{card}(E)}{\text{card}(\Omega)}.$$

La probabilité d'acceptation est donc d'autant plus faible que  $\text{card}(E)$  est petit devant  $\text{card}(\Omega)$ . La loi uniforme sur  $\Omega$  peut cependant sembler naïve. On obtient néanmoins un résultat similaire en prenant comme loi de proposition la loi basée sur la densité marginale des sites (cf. Algorithme 1). On note :

$$p = (p_1, p_0, p_1) \quad \text{avec} \quad p_0 = \frac{1}{1 + 2 \exp(-\beta)}, \quad p_1 = \frac{\exp(-\beta)}{1 + 2 \exp(-\beta)}.$$

Pour un site  $x_i$ , avec  $i \in \llbracket 1, N \rrbracket$ , on définit la densité marginale

$$\tilde{g}(x_i | p) = p_1 \cdot \delta_{-1}(x_i) + p_0 \cdot \delta_0(x_i) + p_1 \cdot \delta_1(x_i).$$

En tirant chaque site indépendamment on obtient donc la densité jointe pour  $x \in \Omega$

$$g(x | p) = \tilde{g}(x_1 | p) \times \dots \times \tilde{g}(x_N | p) = \frac{\exp\{-\beta H(x)\}}{\{1 + 2 \exp(-\beta)\}^N}.$$

On a alors :

$$\frac{\mu_\beta(x)}{g(x | p)} = \frac{\exp\{-\beta H(x)\} \mathbb{1}_{\{x \in E\}}}{Z_\beta} \times \frac{\{1 + 2 \exp(-\beta)\}^N}{\exp\{-\beta H(x)\}} = \frac{\{1 + 2 \exp(-\beta)\}^N}{Z_\beta} \mathbb{1}_{\{x \in E\}}.$$

Avec  $(U_i)_{i \in \mathbb{N}^*}$  iid suivant  $\mathcal{U}([0, 1])$  et  $(x^{(i)})_{i \in \mathbb{N}^*}$  iid suivant  $g(\cdot | p)$ , le temps d'arrêt de la méthode du rejet est donc :

$$T = \inf\{n \in \mathbb{N}^* \mid \mathbb{1}_{\{x^{(n)} \in E\}} \geq U_n\} = \inf\{n \in \mathbb{N}^* \mid x^{(n)} \in E\}.$$

On en déduit que l'espérance du temps d'arrêt est égale à la probabilité d'être dans l'ensemble  $E$ . Ainsi l'algorithme du rejet est d'autant plus long que cette probabilité est faible. Ici encore cette probabilité est d'autant plus faible que le cardinal de  $E$  est petit devant le cardinal de  $\Omega$ .

---

**Algorithme 1 : Algorithme du rejet**


---

**Données :**  $\beta > 0$   
 Calculer  $p = (p_{-1}, p_0, p_1)$   
 Générer  $x \sim g(x | p)$   
**tant que**  $x \notin E$  **faire**  
 | Générer  $x \sim g(x | p)$   
**fin**  
**retourner**  $x$

---

L'algorithme du rejet n'ayant pas de bonnes propriétés, nous nous sommes tournées vers deux autres méthodes de simulation : les méthodes MCMC et l'algorithme de Propp-Wilson.

## 2.1 Rappels sur les chaînes de Markov

Dans cette section nous rappelons les notions et résultats (Huvneers (2021)) sur les chaînes de Markov dont nous aurons besoin par la suite.

**Définition 4.** Une chaîne de Markov dans  $E$  est un processus stochastique  $(X_n)_{n \in \mathbb{N}}$  à valeurs dans  $E$  vérifiant la propriété de Markov : pour tout  $n \in \mathbb{N}$  et pour tout  $(y, x_0, x_1, \dots, x_n) \in E^{n+2}$ ,

$$\mathbb{P}(X_{n+1} = y \mid X_0 = x_0, X_1 = x_1, \dots, X_n = x_n) = \mathbb{P}(X_{n+1} = y \mid X_n = x_n).$$

S'il existe une application  $P : E \times E \rightarrow [0, 1]$  telle que  $\forall n \in \mathbb{N}$  et  $\forall (x, y) \in E \times E$ ,

$$P(x, y) = \mathbb{P}(X_{n+1} = y \mid X_n = x),$$

on dit alors que la chaîne de Markov est **homogène en temps**. Cette application s'appelle noyau de transition (ou matrice de transition lorsqu'on a une représentation matricielle).

**Remarque 4.**  $P$  est une **application stochastique**, i.e pour tout  $x \in E$ ,  $\sum_{y \in E} P(x, y) = 1$ .

Connaître la mesure initiale  $\mu_0$  d'une chaîne de Markov et sa matrice de transition  $P$  permet de calculer sa trajectoire.

**Propriété 1** (Équations de Chapman-Kolmogorov). On suppose  $(X_n)_{n \in \mathbb{N}}$  de mesure initiale  $\mu_0$ . Pour tout  $n \geq 1$  et pour tout  $(x_0, \dots, x_n) \in E^{n+1}$  :

$$\mathbb{P}(X_0 = x_0, \dots, X_n = x_n) = \mu_0(x_0)P(x_0, x_1) \dots P(x_{n-1}, x_n).$$

**Définition 5** (Produit / Produit à gauche). Soit  $n \geq 1$ . Pour tout  $x, y \in E$ , avec  $\mu$  une mesure de probabilité,

- $P^n(x, y) = \sum_{z \in E} P^{n-1}(x, z)P(z, y)$ ,
- $\mu P^n(x) = \sum_{z \in E} \mu(z)P^n(z, x)$ .

Une des propriétés attendues des chaînes dont nous allons avoir besoin est qu'il est toujours possible de passer d'un état quelconque à un autre après un certain nombre d'étapes. On introduit donc la notion d'irréductibilité.

**Définition 6** (Chaîne irréductible).  $(X_n)_{n \in \mathbb{N}}$  est **irréductible** si :

$$\forall (x, y) \in E^2, \quad \exists n \geq 1 \text{ tel que } P^n(x, y) > 0.$$

**Définition 7** (Chaîne apériodique).  $(X_n)_{n \in \mathbb{N}}$  est **apériodique** si :

$$\forall (x, y) \in E^2, \quad \exists n_0 \in \mathbb{N}^* \text{ tel que } \forall n \geq n_0, \quad P^n(x, y) > 0.$$

Autrement dit, pour une chaîne apériodique, on peut toujours passer de  $x$  à  $y$  en  $n_0$  étapes ou plus. Une chaîne apériodique est donc irréductible par définition. À l'inverse, si la chaîne est irréductible et admet un état apériodique, alors tous ses états sont apériodiques.

**Définition 8** (Récurrence et récurrence positive). Soit  $(X_n^x)_{n \in \mathbb{N}}$  la chaîne de Markov débutant en  $x$ . On définit pour tout  $x, y \in E$  le temps d'arrêt  $T_{x,y}$  et la variable aléatoire  $N_{x,y}$  par :

$$T_{x,y} = \min\{n \geq 1 \mid X_n^x = y\} \quad \text{et} \quad N_{x,y} = \sum_{k=0}^{\infty} \mathbb{1}_{\{X_k^x = y\}}$$

Alors :

- $x$  est un état **récurrent** si  $\mathbb{P}(T_{x,x} < +\infty) = 1$ ,
- $x$  est un état **récurrent positif** si  $\mathbb{E}(T_{x,x}) < +\infty$ .

**Propriété 2.** Si la chaîne est irréductible et  $E$  fini, alors tous les états sont récurrents. On dit alors que la chaîne est **récurrente**.

**Définition 9** (Mesure invariante). Une mesure  $\mu$  sur  $E$  est dite **invariante** (ou stationnaire) pour  $P$  si  $\mu P = \mu$ .

**Remarque 5.** Alternativement, on pourra dire que  $\mu$  est  $P$ -invariante (ou  $P$ -stationnaire).

**Remarque 6.** Dans ce mémoire on ne s'intéresse qu'aux mesures non-triviales, i.e., non identiquement nulles ou égales à  $+\infty$ .

**Théorème 1.** Soit une chaîne de Markov  $(X_n)_{n \geq 0}$  irréductible et récurrente. Alors toutes ses mesures stationnaires sont égales à une constante multiplicative près.

**Théorème 2.** Si  $E$  est fini, alors :

1. Toute chaîne de Markov admet une mesure de probabilité stationnaire.
2. Si la chaîne est de plus **irréductible**, alors cette mesure est unique.

**Démonstration.** Démontrons le point 1. de cette propriété. Soit  $\nu$  une mesure de probabilité sur  $E$ , et  $Q$  la matrice de transition associée à la chaîne de Markov. On définit pour tout  $n \in \mathbb{N}^*$  :

$$\begin{aligned} \nu_n : E &\rightarrow [0, 1]^{\text{card}(E)} \\ y &\mapsto \frac{1}{n} \sum_{k=1}^n \nu Q^k(y) = \frac{1}{n} \sum_{k=1}^n \sum_{x \in E} \nu(x) Q^k(x, y), \end{aligned}$$

$\nu_n$  est alors une mesure de probabilité sur  $E$  pour tout  $n \in \mathbb{N}^*$ . Pour montrer ce résultat on commence par montrer par récurrence que :

$$\sum_{y \in E} \nu Q^n(y) = 1 \quad \text{pour tout } n \in \mathbb{N}^*.$$

Initialisation :

On a :

$$\sum_{y \in E} \nu Q(y) = \sum_{y \in E} \sum_{x \in E} \nu(x) Q(x, y) = \sum_{x \in E} \nu(x) \sum_{y \in E} Q(x, y).$$

Comme  $Q$  est une matrice stochastique et  $\nu$  est une mesure de probabilité sur  $E$ , on a bien  $\sum_{y \in E} \nu Q(y) = 1$ .

Hérédité : Soit  $n \in \mathbb{N}^*$ , on suppose que  $\sum_{y \in E} \nu Q^n(y) = 1$ .

On a :

$$\sum_{y \in E} \nu Q^{n+1}(y) = \sum_{y \in E} \sum_{x \in E} \nu(x) \sum_{z \in E} Q^n(x, z) Q(z, y) = \sum_{x \in E} \nu(x) \sum_{z \in E} Q^n(x, z) \sum_{y \in E} Q(z, y).$$

Comme  $Q$  est une matrice stochastique et comme par hypothèse

$$\sum_{x \in E} \nu(x) \sum_{z \in E} Q^n(x, z) = \sum_{x \in E} \nu Q^n(x) = 1,$$

on a bien

$$\sum_{y \in E} \nu Q^{n+1}(y) = \sum_{x \in E} \nu(x) \sum_{z \in E} Q^n(x, z) = 1.$$

Conclusion :  $\forall n \in \mathbb{N}^*$ ,  $\sum_{y \in E} \nu Q^n(y) = 1$ .

Ainsi, pour tout  $n \in \mathbb{N}^*$ ,

$$\sum_{y \in E} \nu_n(y) = \frac{1}{n} \sum_{k=1}^n \sum_{y \in E} \nu Q^k(y) = \frac{1}{n} \sum_{k=1}^n 1 = 1.$$

On a donc que  $\nu_n$  est une mesure de probabilité sur  $E$ ,  $\forall n \in \mathbb{N}^*$ .

Comme  $(\nu_n)_{n \in \mathbb{N}^*}$  est à valeurs dans un compact, on peut en extraire une sous-suite  $(\nu_{\phi(n)})_{n \in \mathbb{N}^*}$  convergente avec  $\phi : \mathbb{N} \rightarrow \mathbb{N}$  strictement croissante. On note  $\pi$  sa limite.  **$\pi$  est bien une mesure de probabilité sur  $E$ .**

$$\sum_{y \in E} \pi(y) = \sum_{y \in E} \lim_{n \rightarrow \infty} \nu_{\phi(n)}(y) = \lim_{n \rightarrow \infty} \sum_{y \in E} \nu_{\phi(n)}(y) = 1.$$

On montre que  **$\pi$  est  $Q$ -invariante**. Autrement dit, on montre que

$$\lim_{n \rightarrow \infty} \nu_{\phi(n)} Q(y) = \lim_{n \rightarrow \infty} \nu_{\phi(n)}(y).$$

Pour tout  $n \in \mathbb{N}^*$  :

$$\nu_n Q(y) = \frac{1}{n} \sum_{k=1}^n \nu Q^{k+1}(y) = \frac{1}{n} \sum_{k=1}^n \nu Q^k(y) - \frac{1}{n} \nu Q(y) + \frac{1}{n} \nu Q^{n+1}(y) = \nu_n(y) + \frac{1}{n} [\nu Q^{n+1}(y) - \nu Q(y)],$$

on en déduit :

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

C'est en particulier vrai pour la sous-suite extraite  $(v_{\phi(n)})_{n \in \mathbb{N}}$ , on obtient pour tout  $y \in E$

$$\lim_{n \rightarrow \infty} (v_{\phi(n)}Q - v_{\phi(n)})(y) = 0.$$

Cette sous-suite étant de limite  $\pi$ , il en résulte

$$(\pi Q - \pi)(y) = 0.$$

La chaîne admet donc bien une mesure stationnaire.

Le point 2. est une conséquence de deux résultats. D'après la Proposition 2, la chaîne étant irréductible sur  $E$  fini, elle est récurrente. Il découle du Théorème 1 qu'il existe une unique mesure stationnaire, car il ne peut y avoir qu'une seule mesure de probabilité parmi tous les multiples d'une mesure sur  $E$ .

□

**Théorème 3** (Théorème ergodique). *Si la chaîne est irréductible et récurrente sur  $E$ , alors pour tout  $x \in E$  et quelle que soit la distribution initiale de la chaîne :*

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{k=1}^n \mathbb{1}_{\{X_k=x\}} = \frac{1}{\mathbb{E}(T_{x,x})} \quad \text{presque sûrement.}$$

**Théorème 4.** *Si  $E$  est fini et si la chaîne est irréductible avec pour mesure invariante  $\mu$ , alors pour tout  $x \in E$ ,*

$$\lim_{n \rightarrow \infty} \mathbb{E} \left( \frac{1}{n} \sum_{k=1}^n \mathbb{1}_{\{X_k=x\}} \right) = \mu(x).$$

**Définition 10** (Mesure  $P$ -réversible). *Une mesure  $\mu$  sur  $E$  est  $P$ -réversible (ou  $P$ -symétrique) si pour tout  $x, y \in E$  :*

$$\mu(x)P(x, y) = \mu(y)P(y, x).$$

**Propriété 3.** *Une mesure  $P$ -réversible est  $P$ -stationnaire.*

*Démonstration.* Supposons  $\mu$  une mesure  $P$ -réversible. On a alors pour tout  $(x, y) \in E^2$  :

$$\mu(x)P(x, y) = \mu(y)P(y, x).$$

D'où

$$\sum_{z \in E} \mu(z)P(z, y) = \sum_{z \in E} \mu(y)P(y, z).$$

Comme  $P$  est une application stochastique, on a

$$\mu P(y) = \mu(y).$$

Donc  $\mu$  est bien  $P$ -stationnaire.

□

## 2.2 Algorithmes MCMC

Les algorithmes MCMC (Robert et Casella (2004)) (pour « *Markov Chain Monte Carlo* ») sont des algorithmes de simulation qui se basent sur la théorie des chaînes de Markov. Leur objectif est de simuler les trajectoires d'une chaîne de Markov dont la mesure de probabilité invariante est la loi que l'on veut simuler. On suppose qu'à partir d'un certain rang, les éléments de la chaîne sont des réalisations de la loi stationnaire. On obtient donc une méthode d'échantillonnage approchée suivant la loi souhaitée. Dans le cadre de ce mémoire, nous nous sommes intéressées à trois algorithmes MCMC : l'algorithme de Metropolis-Hastings, l'algorithme de Gibbs et l'algorithme du recuit simulé.

### 2.2.1 Algorithme de Metropolis-Hastings

L'algorithme de Metropolis-Hastings (Metropolis *et al.* (1953), Chib et Greenberg (1995)) permet de construire une chaîne de Markov dont le noyau de transition est basé sur une loi de proposition et une étape d'acceptation-rejet, et dont la loi invariante est  $\mu_\beta$ . L'algorithme se base sur le résultat suivant :

**Théorème 5** (Metropolis-Hastings). *Soit :*

- un noyau de transition irréductible  $Q$  appelé noyau de proposition (ou noyau d'exploration) tel que

$$\forall (x, y) \in E^2, Q(x, y) = 0 \Leftrightarrow Q(y, x) = 0.$$

- une fonction  $\alpha : \mathbb{R}_+ \rightarrow ]0, 1]$  vérifiant pour tout  $u \in \mathbb{R}_+$   $\alpha(u) = u\alpha(\frac{1}{u})$
- une fonction  $\rho : E \times E \rightarrow [0, 1]$  dite "d'acceptation-rejet" telle que  $\forall (x, y) \in E \times E$  :

$$\rho(x, y) = \alpha \left( \frac{\mu_\beta(y)Q(y, x)}{\mu_\beta(x)Q(x, y)} \right) \mathbb{1}_{\{Q(x, y) > 0\}}$$

On définit le noyau  $P : E \times E \rightarrow \mathbb{R}$  tel que pour tout  $x, y \in E$  :

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

$P$  est alors un noyau de transition sur  $E$ , **irréductible récurrent positif, de loi invariante réversible  $\mu_\beta$** . Il est de plus **apériodique** si  $\alpha < 1$  ou si  $Q$  est apériodique.

**Remarque 7.** Il n'est pas nécessaire d'avoir  $Q$  irréductible pour obtenir que  $\mu_\beta$  soit une mesure  $P$ -invariante. L'irréductibilité de  $Q$  permet en revanche de garantir que c'est l'unique probabilité invariante pour l'algorithme. Dans le cas contraire, on risquerait d'obtenir une simulation suivant une autre loi de probabilité.

**Démonstration.  $P$  est un noyau de transition.** Soit  $(x, y) \in E^2$  tels que  $x \neq y$ . On a, comme  $\rho$  est à valeur dans  $[0, 1]$ ,

$$P(x, y) = Q(x, y)\rho(x, y) \leq Q(x, y) \sup_{\substack{u, v \in E \\ u \neq v}} \rho(u, v) \leq Q(x, y)$$

Donc

$$0 \leq P(x, y) \leq Q(x, y) \leq 1,$$

et  $P$  est bien défini sur  $E \times E$  et à valeurs dans  $[0, 1]$ . De plus, par construction

$$P(x, x) = 1 - \sum_{\substack{z \in E \\ z \neq x}} P(x, z).$$

Donc  $\sum_{z \in E} P(x, z) = 1$  et  $P$  est bien un noyau de transition.

**Il existe une mesure  $P$ -réversible.** Soit  $(x, y) \in E^2$  tels que  $x \neq y$ . On a par définition de la fonction  $\alpha$

$$P(x, y) = Q(x, y) \alpha \left( \frac{\mu_\beta(y) Q(y, x)}{\mu_\beta(x) Q(x, y)} \right) \mathbb{1}_{\{Q(x, y) > 0\}} = Q(x, y) \frac{\mu_\beta(y) Q(y, x)}{\mu_\beta(x) Q(x, y)} \alpha \left( \frac{\mu_\beta(x) Q(x, y)}{\mu_\beta(y) Q(y, x)} \right) \mathbb{1}_{\{Q(x, y) > 0\}}.$$

Or, par hypothèse  $Q(x, y) > 0$  si et seulement si  $Q(y, x) > 0$ , d'où

$$P(x, y) = Q(x, y) \frac{\mu_\beta(y) Q(y, x)}{\mu_\beta(x) Q(x, y)} \alpha \left( \frac{\mu_\beta(x) Q(x, y)}{\mu_\beta(y) Q(y, x)} \right) \mathbb{1}_{\{Q(y, x) > 0\}} = \frac{\mu_\beta(y) Q(y, x)}{\mu_\beta(x)} \rho(y, x).$$

Donc

$$\mu_\beta(x) P(x, y) = \mu_\beta(y) Q(y, x) \rho(y, x) = \mu_\beta(y) P(y, x).$$

$\mu_\beta$  est bien une mesure  $P$ -réversible. On déduit de la propriété 3 que  $\mu_\beta$  est une mesure  $P$ -stationnaire.

**$P$  est irréductible.** Comme  $Q$  est irréductible, pour  $x, y \in E^2$  tels que  $x \neq y$ , il existe  $n \in \mathbb{N}^*$  tel que  $Q^n(x, y) > 0$ . Autrement dit, il existe une suite d'éléments distincts  $(x_1, \dots, x_n) \in E^n$  telle que  $x_1 = x$ ,  $x_n = y$ , et  $\mathbb{P}(X_1 = x_1, \dots, X_n = x_n) > 0$ . D'après l'égalité de Chapman-Kolmogorov, pour tout  $i \in \llbracket 1, n-1 \rrbracket$ ,  $Q(x_i, x_{i+1}) > 0$ . On en déduit que

$$P(x_i, x_{i+1}) = \rho(x_i, x_{i+1}) Q(x_i, x_{i+1}) > 0.$$

D'où

$$P^n(x, y) \geq \prod_{i=1}^n P(x_i, x_{i+1}) > 0.$$

Donc  $P$  est aussi irréductible. D'après le Théorème 2, on déduit que  $\mu_\beta$  est l'unique mesure stationnaire de la chaîne.

**$P$  est récurrente positive.** Soit  $x \in E$  tel que  $\mu_\beta(x) > 0$ . La chaîne est irréductible sur  $E$  fini, donc d'après le théorème 4

$$\mu_\beta(x) = \lim_{n \rightarrow \infty} \mathbb{E} \left( \frac{1}{n} \sum_{k=1}^n \mathbb{1}_{\{X_k = x\}} \right).$$

De plus, d'après la propriété 2, la chaîne est bien récurrente, on peut donc appliquer le théorème ergodique

$$\lim_{n \rightarrow \infty} f_n(x) := \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{k=1}^n \mathbb{1}_{\{X_k = x\}} = \frac{1}{\mathbb{E}(T_{x,x})}.$$

Enfin,  $(f_n)_{n \in \mathbb{N}^*}$  est une suite de fonctions mesurables telles que pour tout  $n \in \mathbb{N}^*$  et pour tout  $x \in E$ ,  $f_n(x) \leq 1$ .  $E$  étant fini,  $Y = 1$  p.s est intégrable, donc d'après le théorème de convergence dominée

$$\mu_\beta(x) = \lim_{n \rightarrow \infty} \mathbb{E} \left( \frac{1}{n} \sum_{k=1}^n \mathbb{1}_{\{X_k = x\}} \right) = \mathbb{E} \left( \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{k=1}^n \mathbb{1}_{\{X_k = x\}} \right) = \mathbb{E} \left( \frac{1}{\mathbb{E}(T_{x,x})} \right) = \frac{1}{\mathbb{E}(T_{x,x})}.$$

On a supposé  $\mu_\beta(x) > 0$ , donc on a  $\mathbb{E}(T_{x,x}) < +\infty$ , et la chaîne est bien récurrente positive.

**$P$  est apériodique si  $\alpha < 1$  ou si  $Q$  est apériodique.**

Si  $\alpha < 1$  : Comme la chaîne est irréductible, pour l'apériodicité, il suffit de montrer qu'il existe  $x \in E$  tel que  $P(x, x) > 0$ . Supposons par l'absurde que pour tout  $x \in E$ ,  $P(x, x) = 0$ . On a alors :

$$P(x, x) = 1 - \sum_{\substack{z \in E \\ z \neq x}} P(x, z) = \sum_{z \in E} Q(x, z) - \sum_{z \neq x} \rho(x, z) Q(x, z) = Q(x, x) + \sum_{\substack{z \in E \\ z \neq x}} Q(x, z) (1 - \rho(x, z)).$$



D'où  $Q(x, x) = -\sum_{\substack{z \in E \\ z \neq x}} Q(x, z)(1 - \rho(x, z))$ . Comme  $Q(x, x) \geq 0$ , on a

$$\sum_{\substack{z \in E \\ z \neq x}} Q(x, z)(1 - \rho(x, z)) \leq 0.$$

Et de plus, comme pour  $z \in E$  tel que  $z \neq x$ ,  $Q(x, z) \geq 0$ , on a forcément  $1 - \rho(x, z) \leq 0$ , d'où  $\rho(x, z) \geq 1$ . Par ailleurs, si  $\alpha < 1$ , alors  $\rho < 1$  d'après la définition de  $\rho$ . On a une contradiction.  $P$  est donc bien apériodique.

Si  $Q$  est apériodique : Soit  $(x, y) \in E^2$  tel que  $x \neq y$ . Comme  $Q$  est apériodique, il existe  $n_0 \in \mathbb{N}^*$  tel que pour tout  $n \in \mathbb{N}^*$  tel que  $n \geq n_0$ , on a  $Q^n(x, y) > 0$ . Autrement dit, il existe une suite d'éléments distincts  $(x_1, \dots, x_n) \in E^n$  telle que  $x_1 = x$ ,  $x_n = y$ , et  $\mathbb{P}(X_1 = x_1, \dots, X_n = x_n) > 0$ . D'après l'égalité de Chapman-Kolmogorov, pour tout  $i \in \llbracket 1, n-1 \rrbracket$   $Q(x_i, x_{i+1}) > 0$ . On en déduit, par définition de  $\rho$ , que

$$P(x_i, x_{i+1}) = \rho(x_i, x_{i+1})Q(x_i, x_{i+1}) > 0.$$

D'où pour tout  $n \geq n_0$ ,

$$P^n(x, y) \geq \prod_{i=1}^n P(x_i, x_{i+1}) > 0.$$

Donc  $P$  est apériodique. □

La fonction d'acceptation-rejet  $\rho$  est construite de manière à respecter la condition de réversibilité de la chaîne. Ainsi, pour tout  $x, y \in E$  tels que  $x \neq y$  :

$$\mu_\beta(x)P(x, y) = \mu_\beta(y)P(y, x) \Leftrightarrow \mu_\beta(x)Q(x, y)\rho(x, y) = \mu_\beta(y)Q(y, x)\rho(y, x) \Leftrightarrow \frac{\rho(x, y)}{\rho(y, x)} = \frac{\mu_\beta(y)Q(y, x)}{\mu_\beta(x)Q(x, y)}.$$

En choisissant  $\alpha(u) = \min(1, u)$  la relation précédente est automatiquement vérifiée.

L'algorithme de Metropolis-Hastings consiste donc à simuler les trajectoires d'une chaîne de Markov récurrente, apériodique et de noyau de transition  $P$ , dont la loi invariante est la mesure de Gibbs  $\mu_\beta$  avec  $\beta > 0$ . Cela revient à simuler suivant le noyau  $Q$  et à accepter la nouvelle configuration avec probabilité  $\rho$ . En procédant ainsi, on simule bien suivant le noyau de transition  $P$ . On définit la variable aléatoire  $Z = x^*$  si  $U < \rho(x, Y)$  avec  $U \sim \mathcal{U}([0, 1])$  et  $Y \sim Q(x, \cdot)$  indépendante de  $U$ , et  $Z = x$  sinon. On a pour  $x \neq x^*$  :

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

La complexité de la simulation de la mesure de Gibbs découle de la fonction de partition  $Z_\beta$ . L'avantage de ce théorème est qu'il permet de construire un noyau  $P$  de mesure invariante  $\mu_\beta$  mais qui ne dépend plus de  $Z_\beta$ . En effet :

$$\frac{\mu_\beta(y)}{\mu_\beta(x)} = e^{-\beta(H(y) - H(x))}, \quad \text{donc} \quad \rho(x, y) = \min\left(1, \frac{Q(y, x)}{Q(x, y)} e^{-\beta(H(y) - H(x))}\right).$$

Pour mettre en oeuvre cet algorithme, il suffit donc de savoir évaluer  $H$ , ou  $H(y) - H(x)$ , qui est souvent simple à calculer.

---

**Algorithme 2 : Metropolis-Hastings**

---

**Données :**  $\beta > 0$ ,  $x^{(0)} \in E$  arbitraire,  $n \in \mathbb{N}^*$

**pour**  $t = 0, \dots, n-1$  **faire**

    Générer  $x^* \sim Q(x^{(t)}, \cdot)$

    Générer  $U \sim \mathcal{U}([0, 1])$

**si**  $U \leq \min \left( 1, \frac{\mu_\beta(x^*)Q(x^*, x^{(t)})}{\mu_\beta(x^{(t)})Q(x^{(t)}, x^*)} \right)$  **alors**

$x^{(t+1)} \leftarrow x^*$

**sinon**

$x^{(t+1)} \leftarrow x^{(t)}$

**fin**

**fin**

**retourner**  $x^{(n)}$

---

**Remarque 8.** À chaque itération, on compare le candidat  $x^*$  à l'état  $x^{(t)}$  courant, en conservant le plus vraisemblable des deux par rapport à la densité  $\mu_\beta$ .

### Applications

Nous présentons dans la suite des applications de l'algorithme de Metropolis-Hasting aux modèles du composé chimique, d'Ising et de Potts.

Nous considérons dans un premier temps la loi de proposition suivante :

$$Q(x^{(t)}, \cdot) \sim \begin{cases} \text{Choisir uniformément un site } j \in S, \\ \text{Choisir uniformément une valeur } x_j^* \in E \text{ pour ce site.} \end{cases} \quad (2.1)$$

À chaque étape de l'algorithme, on simule  $x^* \sim Q(x^{(t)}, \cdot)$ . On a donc  $x_i^* = x_i^{(t)}$ , pour tout  $i \neq j$ . Par symétrie de  $Q$  on a :

$$\rho(x^{(t)}, x^*) = \min \left( 1, \frac{\mu_\beta(x^*)Q(x^*, x^{(t)})}{\mu_\beta(x^{(t)})Q(x^{(t)}, x^*)} \right) = \min \left( 1, \frac{\mu_\beta(x^*)}{\mu_\beta(x^{(t)})} \right).$$

### Composé chimique

- Si  $x^* \notin E$ , on a  $\mu_\beta(x^*) = 0$ , cela implique que  $\rho(x^{(t)}, x^*) = 0$ . Ainsi, on rejette la nouvelle configuration et donc  $x^{(t+1)} = x^{(t)}$ .
- Si  $x^* \in E$ , on a :

$$\rho(x^{(t)}, x^*) = \min \left( 1, \exp \left\{ -\beta (|x_j^*| - |x_j^{(t)}|) \right\} \right).$$

Finalement, on a  $\forall t \in \mathbb{N}$  :

$$x^{(t+1)} = \begin{cases} x^* & \text{si } x^* \in E \text{ et } U \leq \min \left( 1, \exp \{ -\beta (|x_j^*| - |x_j^{(t)}|) \} \right), \\ x^{(t)} & \text{sinon.} \end{cases} \text{ avec } U \sim \mathcal{U}([0, 1]),$$

### Modèle d'Ising

$$\frac{\mu_\beta(x^*)}{\mu_\beta(x^{(t)})} = \frac{\exp \left( \beta \sum_{i \sim k} x_i^* x_k^* \right)}{\exp \left( \beta \sum_{i \sim k} x_i^{(t)} x_k^{(t)} \right)} = \exp \left\{ \beta (x_j^* - x_j^{(t)}) \sum_{i \in v(j)} x_i^{(t)} \right\}.$$

On a donc  $\forall t \in \mathbb{N}$  :

$$x^{(t+1)} = \begin{cases} x^* & \text{si } U \leq \min \left( 1, \exp \left\{ \beta \left( x_j^* - x_j^{(t)} \right) \sum_{i \in v(j)} x_i^{(t)} \right\} \right) \text{ avec } U \sim \mathcal{U}([0, 1]), \\ x^{(t)} & \text{sinon.} \end{cases}$$

### Modèle de Potts

$$\frac{\mu_\beta(x^*)}{\mu_\beta(x^{(t)})} = \frac{\exp \left( \beta \sum_{i \sim k} \mathbb{1}_{\{x_i^* = x_k^*\}} \right)}{\exp \left( \beta \sum_{i \sim k} \mathbb{1}_{\{x_i^{(t)} = x_k^{(t)}\}} \right)} = \exp \left\{ \beta \sum_{i \in v(j)} \left( \mathbb{1}_{\{x_i^{(t)} = x_j^*\}} - \mathbb{1}_{\{x_i^{(t)} = x_j^{(t)}\}} \right) \right\}.$$

On a donc  $\forall t \in \mathbb{N}$  :

$$x^{(t+1)} = \begin{cases} x^* & \text{si } U \leq \min \left( 1, \exp \left\{ \beta \sum_{i \in v(j)} \left( \mathbb{1}_{\{x_i^{(t)} = x_j^*\}} - \mathbb{1}_{\{x_i^{(t)} = x_j^{(t)}\}} \right) \right\} \right) \text{ avec } U \sim \mathcal{U}([0, 1]), \\ x^{(t)} & \text{sinon.} \end{cases}$$

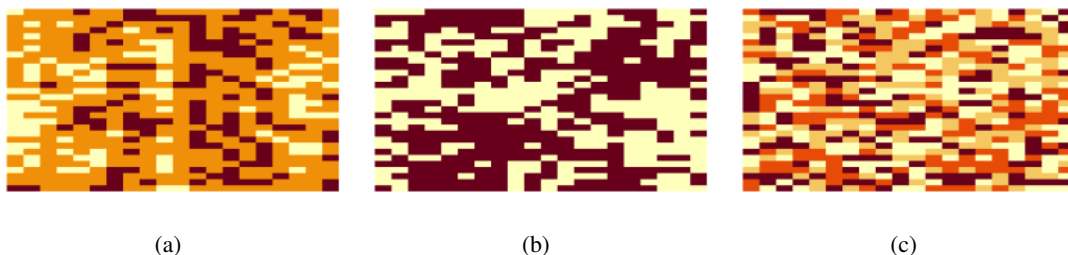


FIGURE 2.1 – Réalisations de mesures de Gibbs de paramètre  $\beta = 0.3$  pour une grille régulière de taille  $20 \times 30$  obtenues avec 600000 itérations de l'algorithme de Metropolis-Hastings (initialisation aléatoire). (a) Composé chimique, (b) Modèle d'Ising, (c) Modèle de Potts à  $K = 4$  états.

### Autre application au composé chimique

Dans le cas du modèle du composé chimique, nous pouvons également choisir la loi de proposition :

$$Q(x^{(t)}, \cdot) \sim \begin{cases} \text{Choisir uniformément un site } j \in S, \\ \text{Choisir une valeur } x_j^* \in \{-1, 0, 1\} \text{ pour ce site avec les probabilités suivantes} \\ p_0 = \mathbb{P}(x_j^* = 0) = \frac{1}{1+2e^{-\beta}} \quad \text{et} \quad p_1 = \mathbb{P}(x_j^* = 1) = \mathbb{P}(x_j^* = -1) = \frac{e^{-\beta}}{1+2e^{-\beta}}. \end{cases} \quad (2.2)$$

- Si  $x^* \notin E$ , comme précédemment  $\mu_\beta(x^*) = 0$ , et donc  $x^{(t+1)} = x^{(t)}$ .
- Si  $x^* \in E$ , en notant à nouveau  $\tilde{g}(\cdot | p = (p_1, p_0, p_1)) = p_1 \cdot \delta_{-1}(\cdot) + p_0 \cdot \delta_0(\cdot) + p_1 \cdot \delta_1(\cdot)$ , on a :

$$\rho(x^{(t)}, x^*) = \min \left( 1, \exp \left\{ -\beta \left( |x_j^*| - |x_j^{(t)}| \right) \right\} \frac{\tilde{g}(x_j^{(t)} | p)}{\tilde{g}(x_j^* | p)} \right).$$

- Si  $|x_j^{(t)}| = |x_j^*|$ , on a  $\tilde{g}(x_j^{(t)} | p) = \tilde{g}(x_j^* | p)$ , donc  $\rho(x^{(t)}, x^*) = 1$ .
- Si  $|x_j^{(t)}| = 1$  et  $x_j^* = 0$ , on a  $\rho(x^{(t)}, x^*) = \min \left( 1, e^{\beta \frac{p_1}{p_0}} \right) = 1$ .

— Si  $x_j^{(t)} = 0$  et  $|x_j^*| = 1$ , on a  $\rho(x^{(t)}, x^*) = \min\left(1, e^{-\beta \frac{p_0}{p_1}}\right) = 1$ .

Autrement dit, si  $x^* \in E$  on accepte toujours la configuration candidate. On a donc pour tout  $t \in \mathbb{N}$  :

$$x^{(t+1)} = \begin{cases} x^* & \text{si } x^* \in E, \\ x^{(t)} & \text{sinon.} \end{cases}$$

## 2.2.2 Algorithme de Gibbs

L'algorithme de Gibbs (Geman et Geman (1984); Robert et Casella (2004)), est un algorithme MCMC dont le noyau repose sur les lois conditionnelles de la loi cible. Il peut être vu comme un cas particulier de Metropolis-Hastings, où pour chaque sommet  $j \in S$  on prend comme loi de proposition  $Q$  la loi conditionnelle suivante :

$$\mu(\cdot | x_{-j}) = \frac{\mu_\beta(x_1, \dots, \cdot, \dots, x_N)}{\sum_{y \in V} \mu_\beta(x_1, \dots, x_{j-1}, y, x_{j+1}, \dots, x_N)} \quad \text{où } x_{-j} = (x_1, \dots, x_N) \text{ privé de } x_j.$$

On obtient alors l'algorithme suivant

---

### Algorithme 3 : Gibbs

---

**Données :**  $\beta > 0$ ,  $x^{(0)} \in E$  arbitraire,  $n \in \mathbb{N}^*$   
**pour**  $t = 0, \dots, n-1$  **faire**  
  **pour chaque sommet**  $j \in S$  **de**  $x^{(t)}$  **faire**  
    Générer  $x_j^* \sim \mu_\beta(\cdot | x_{-j}^{(t)})$   
    Générer  $U \sim \mathcal{U}([0, 1])$   
    **si**  $U \leq \min\left(1, \frac{\mu_\beta(x^*)\mu_\beta(x_j^{(t)} | x_{-j}^*)}{\mu_\beta(x^{(t)})\mu_\beta(x_j^* | x_{-j}^{(t)})}\right)$  **alors**  
       $x_j^{(t+1)} \leftarrow x_j^*$   
    **sinon**  
       $x_j^{(t+1)} \leftarrow x_j^{(t)}$   
    **fin**  
  **fin**  
**fin**  
**retourner**  $x^{(n)}$

---

On remarque qu'à chaque étape et pour chaque sommet la probabilité d'acceptation est alors égale à 1. On accepte donc systématiquement et l'algorithme peut se simplifier de la façon qui suit.

*Démonstration.* Montrons que la probabilité d'acceptation vaut 1. On a :

$$\frac{\mu_\beta(x^*)\mu_\beta(x_j^{(t)} | x_{-j}^*)}{\mu_\beta(x^{(t)})\mu_\beta(x_j^* | x_{-j}^{(t)})} = \frac{\mu_\beta(x_j^*, x_{-j}^*)\mu_\beta(x_j^{(t)} | x_{-j}^*)}{\mu_\beta(x_j^{(t)}, x_{-j}^{(t)})\mu_\beta(x_j^* | x_{-j}^{(t)})} = \frac{\mu_\beta(x_j^* | x_{-j}^*)\mu_\beta(x_{-j}^*)\mu_\beta(x_j^{(t)} | x_{-j}^*)}{\mu_\beta(x_j^{(t)} | x_{-j}^{(t)})\mu_\beta(x_{-j}^{(t)})\mu_\beta(x_j^* | x_{-j}^{(t)})}.$$

Or  $x_{-j}^{(t)} = x_{-j}^*$ , d'où

$$\frac{\mu_\beta(x_j^* | x_{-j}^*)\mu_\beta(x_{-j}^*)\mu_\beta(x_j^{(t)} | x_{-j}^*)}{\mu_\beta(x_j^{(t)} | x_{-j}^{(t)})\mu_\beta(x_{-j}^{(t)})\mu_\beta(x_j^* | x_{-j}^{(t)})} = \frac{\mu_\beta(x_j^* | x_{-j}^*)\mu_\beta(x_{-j}^*)\mu_\beta(x_j^{(t)} | x_{-j}^*)}{\mu_\beta(x_j^{(t)} | x_{-j}^*)\mu_\beta(x_{-j}^*)\mu_\beta(x_j^* | x_{-j}^*)} = 1.$$

Ainsi, avec  $U \sim \mathcal{U}([0, 1])$ , on a  $U \leq \min\left(1, \frac{\mu_\beta(x^*)\mu_\beta(x_j^{(t)} | x_{-j}^*)}{\mu_\beta(x^{(t)})\mu_\beta(x_j^* | x_{-j}^{(t)})}\right) = 1$ .

□

---

**Algorithme 4 : Gibbs**

---

**Données :**  $\beta > 0, x^{(0)} \in E$  arbitraire,  $n \in \mathbb{N}^*$   
**pour**  $t = 0, \dots, n-1$  **faire**  
    **pour** chaque sommet  $j \in S$  de  $x^{(t)}$  **faire**  
         $x_j^{(t+1)} \sim \mu_\beta(\cdot \mid x_{-j}^{(t)})$   
    **fin**  
**fin**  
**retourner**  $x^{(n)}$

---

**Remarque 9.** L'algorithme est présenté pour un parcours des sommets dans l'ordre croissant. Alternativement, on aurait pu les parcourir dans un ordre aléatoire.

### Applications

Dans la suite, nous étudions l'implémentation de l'algorithme de Gibbs aux différents modèles vus jusqu'à présent. Nous calculons les lois conditionnelles pour chacun des modèles. Fixons  $j \in \llbracket 1, N \rrbracket$  le site à modifier.

**Composé chimique** On a  $V = \{-1, 0, 1\}$ . On note  $x_{-j}(\cdot) = (x_1, \dots, x_{j-1}, \cdot, x_{j+1}, \dots, x_N)$ .

$$\mu(\cdot \mid x_{-j}) = \frac{\exp(-\beta \sum_{i \neq j} |x_i| - \beta |\cdot|) \mathbb{1}_{\{x_{-j}(\cdot) \in E\}}}{\sum_{y \in V} \exp(-\beta \sum_{i \neq j} |x_i| - \beta |y|) \mathbb{1}_{\{x_{-j}(y) \in E\}}} = \frac{\exp(-\beta |\cdot|) \mathbb{1}_{\{x_{-j}(\cdot) \in E\}}}{\sum_{y \in V} \exp(-\beta |y|) \mathbb{1}_{\{x_{-j}(y) \in E\}}} \propto \exp(-\beta |\cdot|) \mathbb{1}_{\{x_{-j}(\cdot) \in E\}}.$$

**Modèle d'Ising** On a  $V = \{-1, 1\}$ .

$$\mu(\cdot \mid x_{-j}) = \frac{\exp(\beta \sum_{i \in v(j)} x_i \cdot)}{\sum_{y \in V} \exp(\beta \sum_{i \in v(j)} x_i y)} \propto \exp(\beta \sum_{i \in v(j)} x_i \cdot).$$

**Modèle de Potts** On a  $V = \{1, \dots, K\}$ .

$$\mu(\cdot \mid x_{-j}) = \frac{\exp(\beta \sum_{i \in v(j)} \mathbb{1}_{\{x_i = \cdot\}})}{\sum_{y \in V} \exp(\beta \sum_{i \in v(j)} \mathbb{1}_{\{x_i = y\}})} \propto \exp(\beta \sum_{i \in v(j)} \mathbb{1}_{\{x_i = \cdot\}}).$$

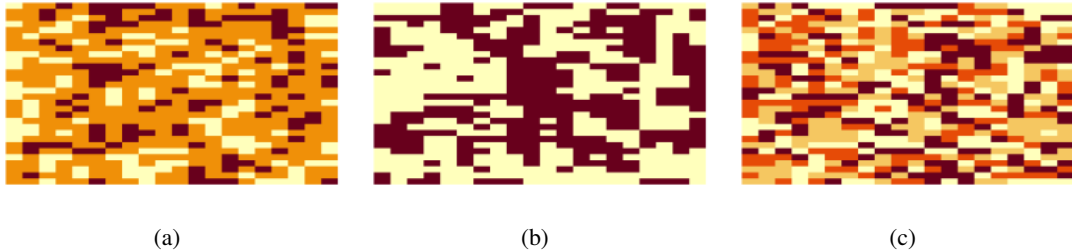


FIGURE 2.2 – Réalisations de mesures de Gibbs de paramètre  $\beta = 0.3$  pour une grille régulière de taille  $20 \times 30$  obtenues avec 1000 itérations de l'algorithme de Gibbs (initialisation aléatoire) (a) Composé chimique. (b) Modèle d'Ising. (c) Modèle de Potts à  $K = 4$  états.

Pour les modèles d'Ising et de Potts nous pouvons observer les résultats de l'algorithme de Gibbs pour différentes valeurs de  $\beta$  sur la Figure 2.3. Nous remarquons que lorsque  $\beta$  augmente, des regroupements par paquets de même

valeur de plus en plus importants se créent jusqu'à obtenir une matrice remplie presque que de la même valeur. Notons  $\mu_\beta^I$  et  $\mu_\beta^P$  les mesures de Gibbs associées respectivement au modèle d'Ising et de Potts. Lorsque  $\beta > 0$ , on a :

$$\mu_\beta^I(\cdot \mid x_{-j}^{(t)}) \propto \exp(\beta \sum_{i \in v(j)} x_i \cdot) \quad \text{et} \quad \mu_\beta^P(\cdot \mid x_{-j}^{(t)}) \propto \exp(\beta \sum_{i \in v(j)} \mathbb{1}_{\{x_i = \cdot\}}).$$

On note  $a \in V$  la valeur la plus fréquente prise par les sites voisins de  $j$ . Pour tout  $y \in V \setminus \{a\}$ , on a :

$$\sum_{i \in v(j)} ax_i > \sum_{i \in v(j)} yx_i \quad \text{et} \quad \sum_{i \in v(j)} \mathbb{1}_{\{x_i = a\}} > \sum_{i \in v(j)} \mathbb{1}_{\{x_i = y\}}.$$

Ainsi,

$$\mu_\beta^I(a \mid x_{-j}^{(t)}) > \mu_\beta^I(y \mid x_{-j}^{(t)}) \quad \text{et} \quad \mu_\beta^P(a \mid x_{-j}^{(t)}) > \mu_\beta^P(y \mid x_{-j}^{(t)}).$$

Plus  $\beta$  est élevé plus l'écart entre ces probabilités est grand. Ainsi, plus  $\beta$  augmente plus grande est la probabilité que le site  $j$  prenne la valeur  $a$ . Ce phénomène s'intensifie jusqu'à atteindre à la limite une distribution uniforme (cf. Résultat 1.1) ne chargeant que les configurations dont tous les sites ont la même valeur.

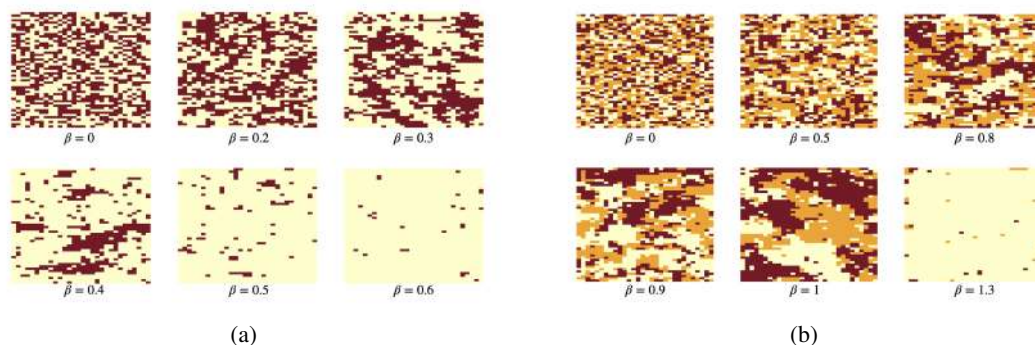


FIGURE 2.3 – Comportement de mesures de Gibbs pour différentes valeurs de  $\beta$  sur une grille régulière de taille  $20 \times 30$ . Réalisations obtenues après 1500 itérations de l'algorithme de Gibbs. (a) Modèle d'Ising. (b) Modèle de Potts à  $K = 3$  états.

### 2.2.3 Algorithme du recuit simulé

La dernière variante de l'algorithme de Metropolis Hastings étudiée est l'algorithme du recuit simulé. Il suit les mêmes étapes que Metropolis-Hastings classique, mais fait croître une suite  $(\beta^{(t)})_{t \in \mathbb{N}}$  au fil du temps. Avec cet algorithme on ne simule donc plus suivant la mesure de Gibbs  $\mu_\beta$  à  $\beta$  fixé. En effet, cet algorithme va nous permettre de simuler des réalisations de la mesure uniforme sur l'ensemble des configurations qui minimisent la fonction énergie  $H$  (cf. Résultat 1.1). Par ailleurs, la chaîne de Markov construite n'est plus homogène en temps car la probabilité de passer d'une configuration à l'autre varie selon l'étape à laquelle on se trouve.

On suppose dans la suite que :

- [H1] : La loi de proposition  $Q$  est symétrique, i.e.,  $\forall (x, y) \in E^2, Q(x, y) = Q(y, x)$ .
- [H2] :  $(\beta^{(t)})_{t \in \mathbb{N}}$  est une suite croissante dans  $\mathbb{R}_+^*$ .

**Remarque 10.** En physique, notre paramètre  $\beta^{(t)}$ ,  $t \in \mathbb{N}$ , s'interprète usuellement comme l'inverse d'une température  $T_t := \frac{1}{\beta^{(t)}}$ . En effet, plus la température d'un composé chimique est élevée moins il est stable. Un  $\beta^{(t)}$  élevé est alors associé à une faible température, et donc à une configuration plus stable.

L'algorithme du recuit simulé se décline de la façon suivante

---

**Algorithme 5 : Recuit simulé**

---

**Données :**  $\beta^{(0)} > 0$ ,  $\lambda > 1$  et  $x^{(0)} \in E$  arbitraire,  $n \in \mathbb{N}$

**pour**  $t = 0, \dots, n-1$  **faire**

    Générer  $x^* \sim \mathcal{Q}(x^{(t)}, \cdot)$

    Générer  $U \sim \mathcal{U}([0, 1])$

**si**  $U \leq \min \left\{ 1, \frac{\mu_{\beta^{(t)}}(x^*)}{\mu_{\beta^{(t)}}(x^{(t)})} \right\}$  **alors**

$x^{(t+1)} \leftarrow x^*$

**sinon**

$x^{(t+1)} \leftarrow x^{(t)}$

**fin**

$\beta^{(t+1)} \leftarrow \lambda \beta^{(t)}$

**fin**

**retourner**  $x^{(n)}$

---

Pour chaque itération  $t$ , avec  $t \in \mathbb{N}$ , on simule une configuration-candidate  $x^*$  suivant  $\mathcal{Q}$ , qu'on accepte avec probabilité :

$$\rho_t(x^{(t)}, x^*) := \min \left\{ 1, \frac{\mu_{\beta^{(t)}}(x^*)}{\mu_{\beta^{(t)}}(x^{(t)})} \right\}.$$

Intuitivement, cela veut dire que si  $H(x^{(t)}) > H(x^*)$ , on accepte toujours  $x^*$ . Sinon, on accepte  $x^*$  avec une probabilité **qui va décroître au fil du temps**. En effet, d'après [H1], on a pour tout  $x \in E$  :

$$\begin{aligned} \rho_t(x, x^*) &= \min \left\{ 1, \exp(-\beta^{(t)}[H(x^*) - H(x)]) \right\} \\ &= \min \left\{ \exp(0), \exp(-\beta^{(t)}[H(x^*) - H(x)]) \right\} \\ &= \exp(-\beta^{(t)} \max\{0, H(x^*) - H(x)\}). \end{aligned}$$

Donc si  $H(x) > H(x^*)$  on a  $\rho_t(x, x^*) = \exp(-\beta^{(t)} \times 0) = 1$ . Tandis que si  $H(x) < H(x^*)$ , la fonction acceptation-rejet dépend du temps et on a  $\rho_t(x, x^*) = \exp(-\beta^{(t)}[H(x^*) - H(x)])$  et donc  $\{\rho_t(x, x^*)\}_{t \geq 0}$  est bien décroissante d'après [H2].

L'énergie des nouvelles configurations aura donc tendance à décroître au fil des étapes. Et il sera de moins en moins probable d'accepter des configurations à plus forte énergie. Cependant, l'algorithme peut être piégé par la présence de modes secondaires, et peut donc ne pas converger vers le minimum global.

**Remarque 11.** *Lors des premières itérations, la température est encore élevée, il y a ainsi une probabilité non nulle de sélectionner un candidat dont l'énergie est plus élevée que celle de la simulation précédente. Cela permet d'éviter d'être piégé dans un minimum local dès les premières itérations de l'algorithme.*

Il n'est pas forcément intéressant d'appliquer l'algorithme du recuit simulé aux modèles du composé chimique, d'Ising, et de Potts : les configurations qui minimisent les fonctions énergies sont connues. Nous choisissons donc d'appliquer cet algorithme à un nouvel exemple : une grille de sudoku.

### Application au jeu du sudoku

Le but est de compléter une grille pré-remplie de chiffres de taille  $9 \times 9$ , et composée de 9 sous-blocs de 9 emplacements. Les contraintes sont alors les suivantes :

1. Un chiffre ne peut pas se trouver deux fois sur la même ligne, sur la même colonne, ou dans un même sous-bloc.
2. Les éléments pré-remplis ne peuvent pas être modifiés.

Dans ce cadre, on peut définir  $E$  et la fonction énergie  $H$  de la manière suivante :

Pour tout  $k \in \llbracket 1, 9 \rrbracket$ ,  $M_k := \{\text{valeurs manquantes du bloc } k\}$  et  $\Sigma_{M_k} := \{\text{permutations de } M_k\}$ . On pose :

$$E = \Sigma_{M_1} \times \dots \times \Sigma_{M_9}.$$

La fonction  $H$  comptabilise le nombre de réplicats par ligne et par colonne.

Le cadre ainsi établi, on procède comme suit :

**Étape 1 - Initialisation** : On complète la grille pré-remplie en tirant aléatoirement un élément de  $E$ . On respecte donc la contrainte par sous-bloc : il ne doit pas y avoir de duplicat dans un même bloc.

**Étape 2 - Acceptation-rejet** : On choisit un sous-bloc arbitrairement, et on échange aléatoirement les valeurs de deux de ses cases. On obtient une nouvelle configuration qu'on acceptera, ou qu'on rejettera, selon la probabilité établie plus haut.

**Étape 3 - Conclusion** : L'algorithme a convergé lorsque  $H$  atteint un minimum local ou global. Si  $H$  vaut 0 alors le puzzle est résolu.

**Remarque 12** (Minimum local). *Il existe un mode secondaire :  $H(x) = 2$ . Si notre algorithme est piégé dans ce mode, on réaugmente la température afin de de permettre à nouveau l'acceptation de configurations à forte énergie et espérer ensuite atteindre  $H(x) = 0$  pour résoudre le puzzle.*

**Exemple 5.** Avec une implémentation sur  $R$  de l'algorithme, on peut résoudre la grille suivante :

			9					6
		3			8	4	5	
				5		9		
		1		4	6	3		2
6								4
9		4	8	3		1		
		2		8				
		5	6	2			7	
8					3			

Grille pré-remplie.

9	2	5	9	6	2	3	2	6
1	8	3	4	3	8	4	5	8
7	6	4	1	5	7	9	7	1
2	8	1	9	4	6	3	5	2
6	3	7	1	2	7	6	8	4
9	5	4	8	3	5	1	9	7
7	4	2	5	8	9	9	1	5
9	5	6	2	4	6	7	6	2
8	3	1	1	7	3	4	3	8

Grille initialisée à  $t = 0$ ,  
 $H(x) = 46$ .

9	2	5	9	6	2	3	2	6
1	8	3	4	3	8	4	5	8
7	6	4	1	5	7	9	7	1
2	8	1	9	4	6	3	5	2
6	3	7	1	2	7	6	8	4
9	5	4	8	3	5	1	9	7
7	4	2	5	8	9	9	1	5
9	5	6	2	4	6	7	3	2
8	3	1	1	7	3	4	6	8

Première permutation acceptée  
à  $t = 4$ ,  $H(x) = 44$ .

5	4	8	9	7	1	2	3	6
1	9	3	6	2	8	4	5	7
2	6	7	3	5	4	9	1	8
7	8	1	5	4	6	3	9	2
6	3	5	1	9	2	8	7	4
9	2	4	8	3	7	1	6	5
3	1	2	7	8	5	6	4	9
4	5	6	2	1	9	7	8	3
8	7	9	4	6	3	5	2	1

Puzzle résolu à  $t = 100759$ ,  
 $H(x) = 0$ .



**En savoir plus.** Les algorithmes MCMC présentés dans ce chapitre convergent à des vitesses différentes et pour un certain nombre d'itérations que nous ne connaissons pas. Bien que nous ne le fassions pas dans ce mémoire, il pourrait être intéressant d'étudier le rang  $n \in \mathbb{N}$  à partir duquel chacun des algorithmes converge.

## 2.3 Algorithme de Propp-Wilson

### 2.3.1 Coupling from the past

Nous avons vu avec les algorithmes MCMC qu'il était possible de créer une chaîne de Markov de mesure invariante  $\mu_\beta$ . Cependant, ces algorithmes sont des algorithmes de simulation « approchée » puisqu'ils nous permettent seulement d'obtenir des réalisations qui suivent une distribution tendant vers la mesure stationnaire. Nous allons à présent étudier l'algorithme de Propp-Wilson (Propp et Wilson (1996)) qui repose sur une technique de couplage (Vigoda (2006)) et qui permet, sous certaines hypothèses, d'obtenir des réalisations de la mesure stationnaire. Le principe est de partir d'un temps dans le passé et de simuler les trajectoires d'une chaîne de Markov de mesure invariante  $\mu_\beta$  à partir de tous les états possibles de  $E$ . Si au temps 0 les trajectoires ont toutes rejoint le même état, alors elles vont continuer d'évoluer ensemble en suivant une même trajectoire et les états parcourus après le temps 0 seront des réalisations de la mesure invariante.

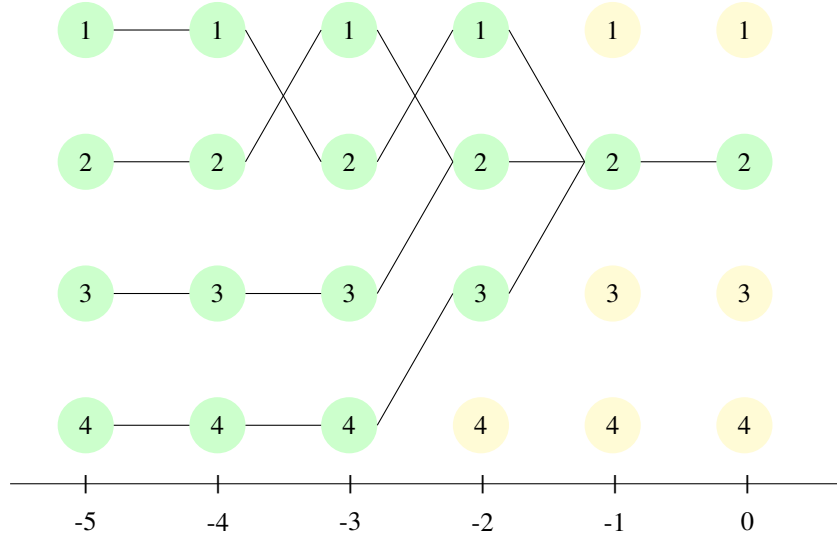


FIGURE 2.4 – Évolution de chaînes de Markov jusqu'au couplage.

Notons  $P$  le noyau de la chaîne admettant  $\mu_\beta$  comme loi stationnaire. Soit  $U \sim \mathcal{U}([0, 1])$  et  $g : E \times [0, 1] \rightarrow E$  une fonction telle que  $g(x, U) \sim P(x, \cdot)$ . Ainsi pour tout  $(x, y) \in E \times E$  :

$$\mathbb{P}(g(x, U) = y) = P(x, y).$$

La chaîne  $(X_t)_{t \in \mathbb{Z}}$  de noyau de transition  $P$  évolue de la manière suivante : au temps  $t$  on se trouve en  $X_t \in E$ , on simule  $U \sim \mathcal{U}([0, 1])$  puis au temps  $t + 1$  la chaîne prend la valeur  $g(X_t, U)$ . Ainsi, soit  $(U_t)_{t \in \mathbb{Z}}$  une suite i.i.d telle que  $\forall t \in \mathbb{Z}, U_t \sim \mathcal{U}([0, 1])$ . On note alors  $g_t(x) := g(x, U_t) \forall x \in E$ , une fonction aléatoire définissant l'évolution de la chaîne  $(X_t)_{t \in \mathbb{Z}} : \forall t \in \mathbb{Z}, X_{t+1} = g_t(X_t)$ . Soit  $t_1, t_2 \in \mathbb{Z}$  tels que  $t_1 < t_2$ . On définit l'évolution de la chaîne du temps  $t_1$  au temps  $t_2$  par l'application aléatoire suivante :

$$\forall x \in E \quad G_{t_1}^{t_2}(x) = (g_{t_2-1} \circ g_{t_2-2} \circ \dots \circ g_{t_1})(x).$$

On a donc :

$$\mathbb{P}(G_{t_1}^{t_2}(x) = y) = P^{t_2-t_1}(x, y).$$

On modélise à l'aide de cette application aléatoire l'ensemble des trajectoires des chaînes partant de tous les états possibles de  $E$ .

**Définition 11** (Temps de couplage). On appelle **temps de couplage**  $C \in \mathbb{N}$ , le temps tel que  $\text{card}(G_0^C(E)) = 1$ . Il s'agit de l'étape à laquelle toutes les chaînes ont atteint le même état, et suivent donc la même trajectoire.

On pose  $T \in \mathbb{N}$  le temps tel que  $\text{card}(G_{-T}^0(E)) = 1$ . Autrement dit, le temps de départ dans le passé pour lequel toutes les chaînes ont coalescé au temps 0. On a alors le théorème suivant :

**Théorème 6.**  $G_{-T}^0$  suit la même loi que  $\mu_\beta$ , la mesure stationnaire de la chaîne.

*Démonstration.* Par homogénéité en temps de la chaîne, pour tout  $t \in \mathbb{N}^*$  et tout  $x, y \in E$ , on a  $\mathbb{P}(G_{-t}^0(x) = y) = \mathbb{P}(G_0^t(x) = y)$ . D'où, par définition de la mesure stationnaire :

$$\lim_{t \rightarrow \infty} \mathbb{P}(G_{-t}^0(x) = y) = \lim_{t \rightarrow \infty} \mathbb{P}(G_0^t(x) = y) = \mu_\beta(y).$$

Ainsi en partant d'un état initial  $x \in E$  et en simulant une trajectoire de la chaîne partant d'un temps infini dans le passé, on obtient au temps 0 une réalisation de  $\mu_\beta$ .

Comme  $T \in \mathbb{N}$  est le temps tel que  $\text{card}(G_{-T}^0(E)) = 1$ , on a pour tout  $x, y \in E$   $G_{-T}^0(x) = G_{-T}^0(y)$ . La fonction  $G_{-T}^0$  est donc constante. Ainsi, pour tout  $t > T$  et pour tout  $x \in E$  on a :

$$G_{-t}^0(x) = (g_{-1} \circ \dots \circ g_{-T} \circ g_{-T-1} \circ \dots \circ g_{-t})(x) = (G_{-T}^0 \circ G_{-t}^{-T})(x) = G_{-T}^0(x).$$

On en déduit que

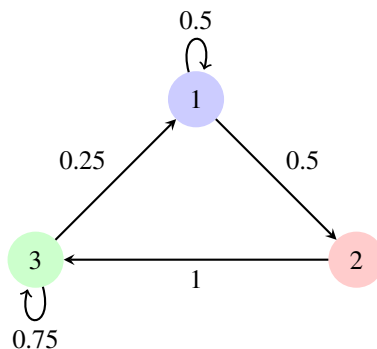
$$\mathbb{P}(G_{-T}^0(x) = y) = \lim_{t \rightarrow \infty} \mathbb{P}(G_{-T}^0(x) = y) = \lim_{t \rightarrow \infty} \mathbb{P}(G_{-t}^0(x) = y) = \lim_{t \rightarrow \infty} \mathbb{P}(G_0^t(x) = y) = \mu_\beta(y).$$

□

*Mais pourquoi retourner dans le passé, plutôt que d'aller dans le futur ?*

Revenons à présent sur la différence entre le couplage dans le passé et le couplage dans le futur. On pourrait intuitivement penser que pour tout  $x \in E$ ,  $G_0^C(x)$  est distribué selon la mesure stationnaire cependant ce n'est pas le cas. En voici un contre exemple simple :

**Contre-exemple 1.** Soit  $(X_t)_{t \geq 0}$  une chaîne de Markov sur  $E = \{1, 2, 3\}$  définie par le graphe suivant :



Soit  $P$  la matrice de transition de la chaîne :

$$P = \begin{bmatrix} 0.5 & 0.5 & 0 \\ 0 & 0 & 1 \\ 0.25 & 0 & 0.75 \end{bmatrix}$$

En résolvant  $\mu P = \mu$  on obtient que  $\mu$  est la mesure uniforme sur  $E = \{1, 2, 3\}$ . On définit la fonction  $f : E \rightarrow E$  telle que  $\forall n \geq 0 : X_{t+1} = f(X_t)$ . On définit également trois chaînes de Markov, copie de  $(X_t)$  :  $(X_t^{(1)}), (X_t^{(2)}), (X_t^{(3)})$ , telles que  $X_0^{(1)} = 1, X_0^{(2)} = 2, X_0^{(3)} = 3$ . On applique simultanément  $f$  aux trois chaînes jusqu'à ce qu'elles se rencontrent. On remarque que l'état 2 n'a qu'un seul prédécesseur, donc le couplage a lieu avant d'arriver en 2. La mesure obtenue au temps de couplage ne charge donc pas le point 2. La mesure obtenue au temps de couplage ne charge donc pas le point 2. Cependant 2 a une probabilité non-nulle dans la mesure stationnaire, de ce fait l'état atteint au temps du couplage n'est clairement pas distribué selon la mesure stationnaire.

Par ailleurs, pour tout  $n > C$  et  $x \in E$  coupler dans le futur ne garantit pas d'avoir  $G_0^n(x) = G_0^C(x)$ , puisqu'après avoir couplé les chaînes continuent d'évoluer. On vient également de montrer que le temps de couplage dans le futur n'est pas le moment où l'on atteint la stationnarité. En revanche, on a montré qu'avec le couplage dans le passé on a bien pour tout  $t > T$ ,  $G_{-t}^0(x) = G_{-T}^0(x)$  et à partir du temps 0 on obtient des réalisations de la mesure stationnaire.

Ainsi afin de simuler  $\mu_\beta$  avec Propp-Wilson, il convient de simuler les  $\text{card}(E)$  trajectoires de la chaîne  $(X_t)_{t \in \mathbb{Z}}$  avec  $G_{-t}^0$  pour  $t \in \mathbb{N}$ , chacune d'elles partant d'un état distinct dans  $E$ . S'il y a eu coalescence des trajectoires en  $x^*$  au temps 0, alors  $x^*$  est bien une réalisation de  $\mu_\beta$ . Si ce n'est pas le cas, on recule d'un pas  $p \in \mathbb{N}^*$  dans le passé et on simule à nouveau les trajectoires jusqu'à obtenir leur coalescence au temps 0. L'algorithme de Propp-Wilson s'implémente donc ainsi :

---

**Algorithme 6 : Propp-Wilson**

---

**Données :**  $n$  et  $p$  deux entiers,  $(x^{(1)}, x^{(2)}, \dots, x^{(\Lambda)})$  le vecteur de tous les éléments distincts de  $E$  avec  $\Lambda = \text{card}(E)$ , un vecteur  $U = (U_{-n}, \dots, U_{-1}, U_0) \sim \mathcal{U}([0, 1])^{\otimes (n+1)}$

**répéter**

**pour**  $t$  allant de  $n$  à 0 **faire**

$x^{(1)} \leftarrow g(x^{(1)}, U_{-t})$

$x^{(2)} \leftarrow g(x^{(2)}, U_{-t})$

$\vdots$

$x^{(\Lambda)} \leftarrow g(x^{(\Lambda)}, U_{-t})$

**fin**

**pour**  $t$  allant de  $n+1$  à  $n+p$  **faire**

    Générer  $U_{-t} \sim \mathcal{U}([0, 1])$

$U \leftarrow (U_{-t}, U)$

**fin**

$n \leftarrow n + p$

**jusqu'à**  $x^{(1)} = x^{(2)} = \dots = x^{(\Lambda)}$ ;

**retourner**  $x^{(1)}$

---

### 2.3.2 La méthode du sandwich

L'algorithme de Propp-Wilson suggère qu'il faut suivre autant de chaînes de Markov qu'il existe de configurations dans l'espace  $E$ . Cette solution n'est donc pas réaliste pour un ensemble  $E$  de grand cardinal. Lorsque  $E$  peut être muni d'une relation d'ordre partielle, si la transformation  $g_t$  vérifie une propriété de monotonie, la méthode du sandwich consiste à appliquer l'algorithme uniquement au  $\min$  et au  $\max$  de  $E$  pour cette relation d'ordre. On définit une relation d'ordre  $\preceq$  sur  $E$  avec  $\omega \in E$  et  $\gamma \in E$  tels que pour tout  $x \in E$  on a :

$$\omega \preceq x \preceq \gamma.$$

Si pour tout  $t \in \mathbb{N}$ ,  $g_t$  préserve cette relation d'ordre  $\preceq$  :

$$g_t(\omega) \preceq g_t(x) \preceq g_t(\gamma),$$

alors pour tout  $n \in \mathbb{N}$ , et pour tout  $x \in E$  on a que :

$$G_{-t}^0(\omega) \preceq G_{-t}^0(x) \preceq G_{-t}^0(\gamma).$$

Donc si on note  $T$  le temps de couplage de  $\omega$  et  $\gamma$  on a  $G_{-T}^0(\omega) = G_{-T}^0(\gamma)$  et la relation d'ordre implique que pour tout  $x \in E$  on a également  $G_{-T}^0(\omega) = G_{-T}^0(\gamma) = G_{-T}^0(x)$ . Ainsi, si  $\omega$  et  $\gamma$  ont coalescé au temps  $T$ , alors toutes les chaînes partant des autres états de  $E$  aussi.

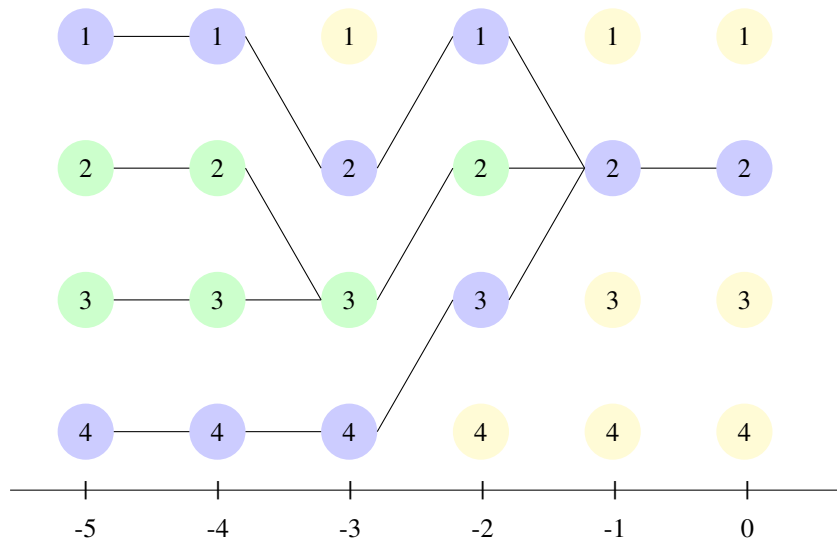


FIGURE 2.5 – Chaînes de Markov bornées par deux chaînes jusqu'au couplage.

## Applications

Nous allons à présent appliquer cette technique aux modèles du composé chimique et d'Ising. Pour cela, nous définissons une relation d'ordre sur les matrices de l'ensemble  $E$  du modèle.

**Définition 12.** Considérons la relation d'ordre sur  $E$  suivante :

$$x \preceq y \Leftrightarrow \forall i \in \llbracket 1, N \rrbracket : x_i \leq y_i.$$

En notant  $-\mathbb{1}$  la matrice dont tous les coefficients valent  $-1$  et  $\mathbb{1}$  la matrice dont tous les coefficients valent  $1$ , nous avons donc pour le modèle du composé chimique et d'Ising :

$$\forall x \in E : -\mathbb{1} \preceq x \preceq \mathbb{1}.$$

De ce fait, en prenant les  $g_t$  définis ci-dessous pour le modèle du composé chimique et d'Ising, toutes les trajectoires sont bornées par les trajectoires partant des configurations  $-\mathbb{1}$  et  $\mathbb{1}$ . Nous pouvons ainsi appliquer la technique du sandwich et considérer uniquement ces deux trajectoires.

**Composé chimique** On définit  $g_t$  grâce au noyau de proposition défini précédemment (2.2). Et on a  $\forall t \in \mathbb{N}$  :

$$z^{(t+1)} = g_t(z^{(t)}) = \begin{cases} z^* & \text{si } z^* \in E, \\ z^{(t)} & \text{sinon.} \end{cases}$$

Montrons que  $g_t$  préserve la relation d'ordre  $\preceq$  définie plus haut. Soient  $x^{(t)}, y^{(t)} \in E$  tels que  $x^{(t)} \preceq y^{(t)}$ . On a 3 cas possibles :

- Si  $x_j^* = y_j^* = 0$  :  $x^{(t+1)} = x^*$  et  $y^{(t+1)} = y^*$  et comme  $x^* \preceq y^*$  on a bien :  $x^{(t+1)} \preceq y^{(t+1)}$ .
- Si  $x_j^* = y_j^* = 1$  : on doit montrer que si  $x^{(t+1)} = x^*$  alors  $y^{(t+1)} = y^*$ . Or on a  $x^{(t+1)} = x^*$  si et seulement si  $x_j^{(t)}$  n'a pas de voisins qui valent  $-1$ . Or cela implique que  $y_j^{(t)}$  n'a pas de voisins qui valent  $-1$  et donc que  $y^{(t+1)} = y^*$ .
- Si  $x_j^* = y_j^* = -1$  : on doit montrer que si  $y^{(t+1)} = y^*$  alors  $x^{(t+1)} = x^*$ . Or on a  $y^{(t+1)} = y^*$  si et seulement si  $y_j^{(t)}$  n'a pas de voisins qui valent  $1$ . Or cela implique que  $x_j^{(t)}$  n'a pas de voisins qui valent  $1$  et donc que  $x^{(t+1)} = x^*$ .

Finalement,  $g_t$  préserve bien la relation d'ordre, on a alors :

$$\text{card}(G_{-T}^0(E)) = 1 \quad \Leftrightarrow \quad G_{-T}^0(-\mathbb{1}) = G_{-T}^0(\mathbb{1}).$$

**Modèle d'Ising** On définit  $g_t$  grâce au noyau de proposition défini précédemment (2.1). Et on a  $\forall t \in \mathbb{N}$  :

$$z^{(t+1)} = g_t(z^{(t)}) = \begin{cases} z^* & \text{si } U \leq \min \left( 1, \exp\{\beta(z_j^* - z_j^{(t)}) \sum_{i \in v(j)} z_i^{(t)}\} \right) \text{ avec } U \sim \mathcal{U}([0, 1]) \\ z^{(t)} & \text{sinon.} \end{cases}$$

Montrons que  $g_t$  préserve la relation d'ordre  $\preceq$  définie plus haut. Soient  $x^{(t)}, y^{(t)} \in E$  tels que  $x^{(t)} \preceq y^{(t)}$ . On a 2 cas possibles :

- Si  $x_j^* = y_j^* = 1$  : dans le cas où  $y_j^{(t)} = 1$  la relation d'ordre est forcément préservée. Cependant, dans le cas où  $y_j^{(t)} = -1$ , on doit montrer que si  $x^{(t+1)} = x^*$  alors  $y^{(t+1)} = y^*$ . Or si  $y_j^{(t)} = -1$  alors  $x_j^{(t)} = -1$  et on a :

$$\frac{\exp \left( \beta(x_j^* - x_j^{(t)}) \sum_{i \in v(j)} x_i^{(t)} \right)}{\exp \left( \beta(y_j^* - y_j^{(t)}) \sum_{i \in v(j)} y_i^{(t)} \right)} = \exp \left\{ 2\beta \left( \sum_{i \in v(j)} x_i^{(t)} - \sum_{i \in v(j)} y_i^{(t)} \right) \right\} \leq 1$$

Donc si  $y_j^{(t)} = -1$  et  $x^{(t+1)} = x^*$  alors  $y^{(t+1)} = y^*$ .

- Si  $x_j^* = y_j^* = -1$  : dans le cas où  $x_j^{(t)} = -1$  la relation d'ordre est forcément préservée. Cependant, dans le cas où  $x_j^{(t)} = 1$  on doit montrer que si  $y^{(t+1)} = y^*$  alors  $x^{(t+1)} = x^*$ . Or si  $x_j^{(t)} = 1$  alors  $y_j^{(t)} = 1$  et on a :

$$\frac{\exp \left( \beta(y_j^* - y_j^{(t)}) \sum_{i \in v(j)} y_i^{(t)} \right)}{\exp \left( \beta(x_j^* - x_j^{(t)}) \sum_{i \in v(j)} x_i^{(t)} \right)} = \exp \left\{ 2\beta \left( \sum_{i \in v(j)} x_i^{(t)} - \sum_{i \in v(j)} y_i^{(t)} \right) \right\} \leq 1$$

Donc si  $x_j^{(t)} = 1$  et  $y^{(t+1)} = y^*$  alors  $x^{(t+1)} = x^*$ .

Finalement,  $g_t$  préserve bien la relation d'ordre, on a alors :

$$\text{card}(G_{-T}^0(E)) = 1 \quad \Leftrightarrow \quad G_{-T}^0(-\mathbb{1}) = G_{-T}^0(\mathbb{1}).$$

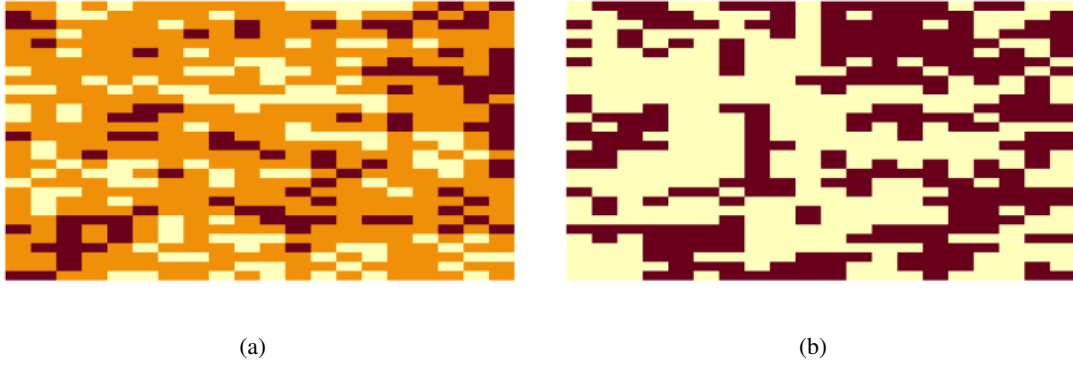


FIGURE 2.6 – Réalisations de mesures de Gibbs de paramètre  $\beta = 0.3$  pour une grille régulière de taille  $20 \times 30$  obtenues avec l’algorithme de Propp-Wilson. (a) Composé chimique. (b) Modèle d’Ising.

## 2.4 Comparaison des méthodes

Nous disposons d’algorithmes de simulation approchée, et d’un algorithme de simulation exacte. Nous proposons de valider nos algorithmes MCMC, pour savoir notamment si nous avons pris un nombre d’itérations suffisant. Nous comparons donc la distribution empirique du hamiltonien simulée avec un algorithme MCMC, et simulée avec Propp-Wilson à l’aide d’un diagramme quantile-quantile.

Les Figures 2.7 et 2.8 présentent les résultats obtenus pour les modèles du composé chimique et d’Ising définis sur une grille régulière de taille  $20 \times 30$  pour  $\beta = 0.3$ .

Pour construire la distribution empirique de  $H(x)$ , on a simulé 500 réalisations  $20 \times 30 \times 1000$  itérations pour l’algorithme de Metropolis-Hastings et 1000 itérations pour l’algorithme de Gibbs.

Nous remarquons que les points sont alignés le long de la première bissectrice, ainsi les distributions obtenues avec les algorithmes de Metropolis-Hastings et Gibbs suivent probablement la distribution obtenue avec l’algorithme de Propp-Wilson et donc  $\mu_\beta$ . Il est donc pertinent d’utiliser les algorithmes de Metropolis-Hastings et Gibbs plutôt que l’algorithme de Propp-Wilson car ils donnent des résultats satisfaisants et sont plus rapides.

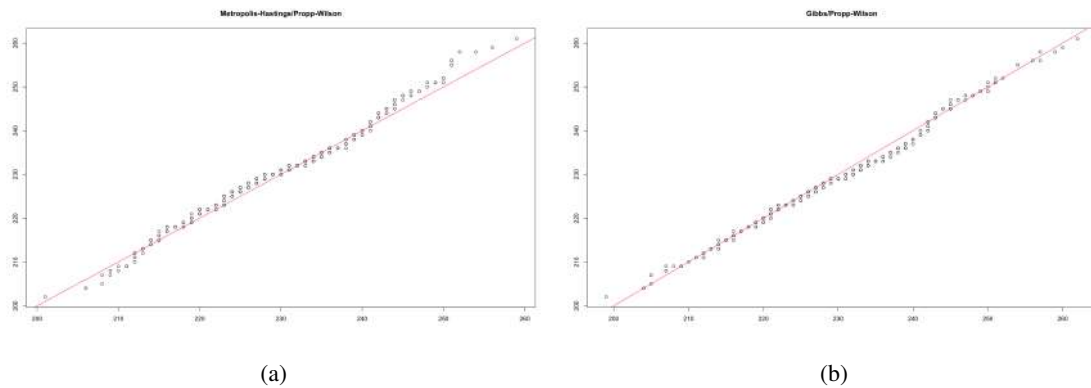


FIGURE 2.7 – Diagramme quantile-quantile pour 500 réalisations du hamiltonien du modèle du composé chimique avec  $\beta = 0.3$  et une grille régulière de taille  $20 \times 30$ . En abscisse, quantiles obtenus avec l’algorithme de Propp-Wilson. En ordonnée, quantiles obtenus avec (a) 600000 itérations de Metropolis-Hastings et (b) 1000 itérations de Gibbs.

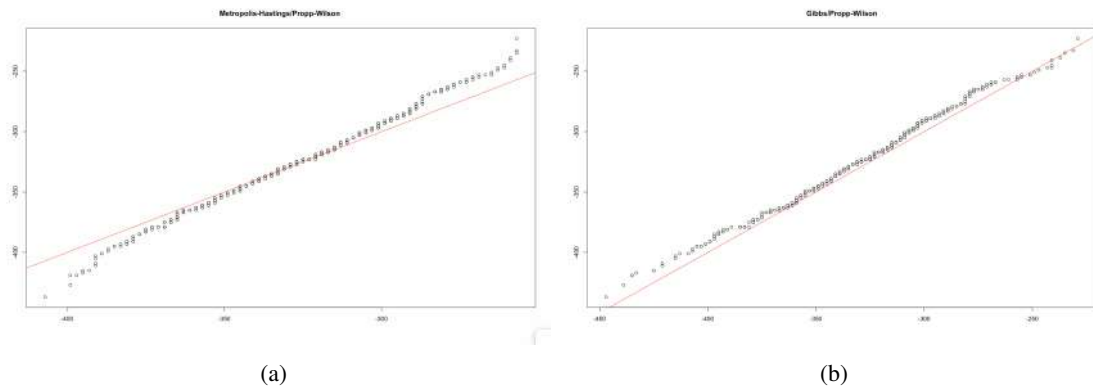


FIGURE 2.8 – Diagramme quantile-quantile pour 500 réalisations du hamiltonien du modèle d’Ising avec  $\beta = 0.3$  et une grille régulière de taille  $20 \times 30$ . En abscisse, quantiles obtenus avec l’algorithme de Propp-Wilson. En ordonnée, quantiles obtenus avec (a) 600000 itérations de Metropolis-Hastings et (b) 1000 itérations de Gibbs.

# 3

## Inférence du paramètre de la mesure de Gibbs

Dans ce chapitre, nous nous intéressons aux problèmes d'estimation dans les paradigmes fréquentiste et bayésien lorsque le modèle statistique est supposé être une mesure de Gibbs.

### 3.1 Estimateur du maximum de vraisemblance

**Définition 13** (Fonction de vraisemblance). *Pour un modèle statistique  $\mathcal{P} = \{f(\cdot | \theta); \theta \in \Theta \subset \mathbb{R}^d\}$  la fonction de vraisemblance correspond à la loi jointe de  $(X^{(1)}, \dots, X^{(n)})$  avec  $X^{(1)}, \dots, X^{(n)}$  distribuées suivant  $f(\cdot | \theta)$ , vue comme fonction de  $\theta$ . Si on dispose de  $x^{(1)}, \dots, x^{(n)}$  réalisations indépendantes suivant  $f$ , elle s'écrit :*

$$L(\theta | x^{(1)}, \dots, x^{(n)}) = \prod_{i=1}^n f(x^{(i)} | \theta).$$

On suppose un modèle statistique paramétrique  $\mathcal{P} = \{\mu_\beta; \beta \in \mathbb{R}_+\}$  d'observation  $x^{\text{obs}}$ . On souhaite trouver une estimation du paramètre  $\beta$  inconnu par la méthode du maximum de vraisemblance, i.e., trouver

$$\hat{\beta} = \underset{\beta \in \mathbb{R}_+}{\operatorname{argmax}} L(\beta | x^{\text{obs}}) = \underset{\beta \in \mathbb{R}_+}{\operatorname{argmax}} \ell(\beta | x^{\text{obs}}) \quad \text{où } \ell(\cdot | x^{\text{obs}}) \text{ est la log-vraisemblance.}$$

Pour la mesure de Gibbs la log-vraisemblance est donnée pour  $\beta \in \mathbb{R}_+$  par :

$$\ell(\beta | x^{\text{obs}}) = \log\{\mu_\beta(x^{\text{obs}})\} = -\beta H(x^{\text{obs}}) - \log(Z_\beta).$$

En dérivant la log-vraisemblance par rapport à  $\beta$  on obtient :

$$\begin{aligned} \frac{\partial}{\partial \beta} \ell(\beta | x^{\text{obs}}) &= \frac{\partial}{\partial \beta} [-\beta H(x^{\text{obs}}) - \log(Z_\beta)] \\ &= -H(x^{\text{obs}}) - \frac{\frac{\partial}{\partial \beta} Z_\beta}{Z_\beta} \\ &= -H(x^{\text{obs}}) - \frac{1}{Z_\beta} \sum_{z \in E} -H(z) e^{-\beta H(z)} \\ &= -H(x^{\text{obs}}) + \mathbb{E}_\beta[H(X)] \quad \text{avec } X \sim \mu_\beta. \end{aligned}$$

Le gradient dépend d'une espérance par rapport à la mesure de Gibbs. On ne dispose pas d'une forme analytique pour les points critiques (s'ils existent). On va donc utiliser un algorithme de descente de gradient pour lequel l'espérance sera estimée par une méthode de Monte-Carlo (Robbins et Monro (1951)). On a l'estimateur de  $\mathbb{E}_\beta[H(X)]$



suivant :

$$\hat{\delta}_k = \frac{1}{k} \sum_{i=1}^k H(x^{(i)}) \quad \text{avec } k \in \mathbb{N} \text{ et } x^{(1)}, \dots, x^{(k)} \stackrel{iid}{\sim} \mu_\beta.$$

Finalement, on obtient l'estimateur du gradient suivant :

$$\hat{\nabla} := \frac{1}{k} \sum_{i=1}^k H(x^{(i)}) - H(x^{\text{obs}}) \quad \text{avec } k \in \mathbb{N}^* \quad \text{et } x^{(1)}, \dots, x^{(k)} \stackrel{iid}{\sim} \mu_\beta.$$

Pour simuler suivant  $\mu_\beta$ , on utilise les méthodes vues dans le Chapitre 2. Le schéma de descente de gradient est alors donné pour  $n \in \mathbb{N}^*$  par  $\beta^{(n+1)} \leftarrow \beta^{(n)} + \eta^{(n)} \hat{\nabla}$  avec  $\eta^{(n)}$  un pas qui vérifie les propriétés suivantes :

$$\sum_{i=1}^{\infty} \eta^{(i)} > \infty \quad \text{et} \quad \sum_{i=1}^{\infty} (\eta^{(i)})^2 < \infty.$$

Dans l'Algorithme 7 on choisit de prendre le pas à l'itération  $n \in \mathbb{N}^*$  suivant :

$$\eta^{(n)} = \frac{1}{n \left| \max_{x \in E} H(x) \right|}.$$

---

**Algorithme 7 : Descente de gradient**

---

**Données :**  $x^{\text{obs}} \in E$ ,  $\varepsilon$  et  $\beta^{(1)}$  des réels positifs et  $k$  un entier

**pour**  $t$  allant de 1 à  $k$  **faire**

    | Générer  $x^{(t)} \sim \mu_{\beta^{(1)}}$

**fin**

$n \leftarrow 1$

Calculer  $\hat{\nabla} = \frac{1}{k} \sum_{t=1}^k H(x^{(t)}) - H(x^{\text{obs}})$

**tant que**  $|\hat{\nabla}| > \varepsilon$  **faire**

    |  $\beta^{(n+1)} \leftarrow \beta^{(n)} + \frac{1}{n \left| \max_{x \in E} H(x) \right|} \hat{\nabla}$

**pour**  $t$  allant de 1 à  $k$  **faire**

        | Générer  $x^{(t)} \sim \mu_{\beta^{(n+1)}}$

**fin**

    Calculer  $\hat{\nabla} = \frac{1}{k} \sum_{t=1}^k H(x^{(t)}) - H(x^{\text{obs}})$

$n \leftarrow n + 1$

**fin**

**retourner**  $\beta^{(n)}$

---

**En savoir plus.** Nous aurions pu faire une calibration du pas par une méthode de recherche linéaire. Cependant, cette méthode est difficile à mettre en oeuvre dans notre cas. En effet, elle nécessite le calcul de

$$\ell(\beta^{(n+1)}) - \ell(\beta^{(n)}) = H(x^{\text{obs}}) \left( \beta^{(n)} - \beta^{(n+1)} \right) - \log \left( \frac{Z_{\beta^{(n+1)}}}{Z_{\beta^{(n)}}} \right).$$

Cette différence dépend du ratio des fonctions de partition qui est impossible à calculer. Il faudrait donc utiliser à nouveau un estimateur de Monte-Carlo.

## Application

Nous avons implémenté cette méthode pour un modèle d'Ising défini sur une grille régulière de taille  $30 \times 20$ . Notre observation a été simulée grâce à l'algorithme de Propp-Wilson pour  $\beta = 0.3$ . Partant de  $\beta^{(1)} = 0.8$ , à

chaque itération  $t$  de l'algorithme, on a simulé  $k$  réalisations suivant  $\mu_{\beta^{(t)}}$  à l'aide de l'algorithme de Metropolis-Hastings (avec 600000 itérations). La Figure 3.1 représente la trajectoire du paramètre  $\beta^{(t)}$ . Pour  $\varepsilon = 1$ , on obtient comme estimation finale de  $\beta$  :

$$\hat{\beta} = \begin{cases} 0.339 & \text{pour } k = 1, \\ 0.340 & \text{pour } k = 500, \\ 0.336 & \text{pour } k = 1000. \end{cases}$$

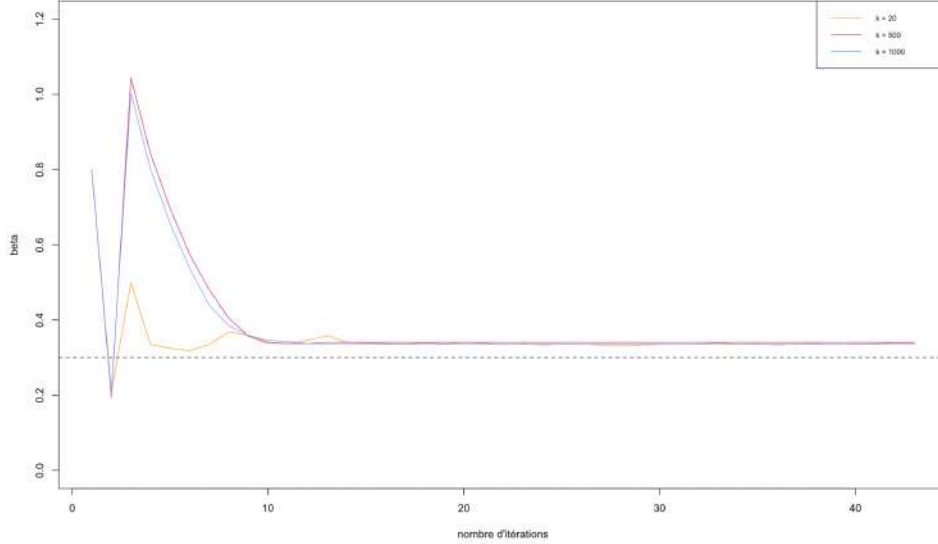


FIGURE 3.1 – Evolution de  $\beta^{(t)}$  au cours de l'algorithme de Robins Monro pour un modèle d'Ising sur une grille régulière de taille  $30 \times 20$ . Le gradient est estimé par  $k$  tirages suivant  $\mu_{\beta^{(t)}}$ . L'algorithme s'arrête lorsque le gradient estimé est inférieur à  $\varepsilon = 1$ . L'observation correspond à des données simulées pour la mesure de Gibbs du modèle d'Ising de paramètre  $\beta = 0.3$ .

## 3.2 Estimation de la loi *a posteriori*

Dans le cadre bayésien, on munit l'espace des paramètres d'une distribution de probabilité qui décrit l'incertitude sur le paramètre.

**Définition 14** (Modèle bayésien). *Un modèle bayésien est la donnée d'un modèle paramétrique  $f(\cdot | \theta)$  avec  $\theta \in \Theta \subset \mathbb{R}^d$  et d'une loi a priori de densité  $\pi(\cdot)$  sur  $\Theta$ .*

Ainsi le paramètre  $\theta$  devient lui-même une variable aléatoire de densité  $\pi$ .

**Définition 15** (Loi *a posteriori*). *La loi conditionnelle de  $\theta$  sachant  $x$  est appelée distribution *a posteriori* et est définie par*

$$\pi(\theta | x) = \frac{L(\theta | x)\pi(\theta)}{\int_{\Theta} L(u | x)\pi(u)du}.$$

Comme on ne sait pas calculer l'intégrale sur  $\Theta$ , en pratique on note souvent :

$$\pi(\theta | x) \propto L(\theta | x)\pi(\theta).$$

La distribution *a posteriori* joue un rôle central dans l'inférence bayésienne et permet de traiter les questions d'inférence liées à  $\theta$ . Supposons que l'on souhaite obtenir un estimateur ponctuel de  $\theta$ . Pour une observation  $x^{\text{obs}}$ , un estimateur bayésien de  $\theta$  est

$$\delta(x^{\text{obs}}) = \underset{u \in \Theta}{\operatorname{argmin}} \mathbb{E}_{\pi(\cdot | x^{\text{obs}})}[C(u, \theta)],$$

où  $C : \Theta \times \Theta \rightarrow \mathbb{R}_+$  est une fonction appelée fonction de coût.

**Remarque 13.** La fonction de coût mesure l'erreur commise lorsqu'on estime  $\theta$  par  $u$ . L'estimateur bayésien est donc celui qui minimise cette erreur en espérance.

Il existe plusieurs choix de fonction de coût  $C$ . Dans le cas particulier où l'on prend la perte quadratique, *i.e.*,  $C(u, \theta) = \|u - \theta\|^2$ , l'estimateur bayésien, s'il est bien défini, est

$$\delta(x^{\text{obs}}) = \mathbb{E}_{\pi(\cdot | x^{\text{obs}})}[\theta] = \int_{\Theta} \theta \pi(\theta | x^{\text{obs}}) d\theta.$$

Il est néanmoins difficile, si ce n'est impossible, de calculer l'espérance de la loi *a posteriori*. En effet, intégrer sur  $\Theta$  s'avère être compliqué et, de plus, on ne connaît généralement  $\pi(\cdot | x)$  qu'à une constante près. On peut cependant estimer cette espérance en utilisant une généralisation du théorème ergodique.

**Théorème 7** (Théorème ergodique généralisé). Soit  $Q$  un noyau irréductible sur  $E$  de probabilité  $Q$ -invariante  $\mu$ . Si  $(X_n)_{n \in \mathbb{N}}$  est une chaîne de Markov de loi initiale  $\mu_0$  et de matrice de transition  $Q$ , alors pour toute fonction  $h$  qui est  $\mu$ -intégrable :

$$\frac{1}{n} \sum_{k=1}^n h(X_k) \xrightarrow[n \rightarrow +\infty]{} \int h(x) \mu(dx) \quad p.s.$$

Ainsi, si l'on sait construire une chaîne de Markov de transition irréductible et ayant  $\pi(\cdot | x)$  comme probabilité invariante, on pourra approcher l'espérance *a posteriori* par la moyenne empirique calculée sur la trajectoire. Dans le cadre bayésien le problème d'estimation se ramène au problème de simulation suivant  $\pi(\cdot | x)$ . Nous avons vu dans le Chapitre 2 comment répondre à ce problème. Dans la suite, on utilise l'algorithme de Metropolis-Hastings. On note néanmoins que l'on ne travaille plus avec des chaînes à espace d'états finis ( $\theta \in \Theta \subset \mathbb{R}^d$ ). On admet alors la généralisation des résultats au cas des espaces d'états infini.

Dans le cas où le modèle paramétrique considéré est une mesure de Gibbs  $\mu_\beta$  on rencontre néanmoins des difficultés dans la mise en oeuvre de l'algorithme de Metropolis-Hastings. En effet, dans le quotient d'acceptation on retrouve le ratio suivant :

$$\frac{\pi(\beta^* | x^{\text{obs}})}{\pi(\beta | x^{\text{obs}})} = \frac{\pi(\beta^*) \mu(\beta^* | x^{\text{obs}})}{\pi(\beta) \mu(\beta | x^{\text{obs}})} = \frac{\pi(\beta^*) \exp\{-\beta^* H(x^{\text{obs}})\} Z_\beta}{\pi(\beta) \exp\{-\beta H(x^{\text{obs}})\} Z_{\beta^*}}.$$

La probabilité d'acceptation dépend ici du ratio de fonctions de partition, incalculable en pratique, ce qui rend l'implémentation de l'algorithme impossible. Une solution est donc de remplacer le noyau de transition par une version bruitée, *i.e.*, en remplaçant ce ratio par un estimateur de Monte-Carlo (Murray *et al.* (2012) Alquier *et al.* (2016)). En effet, on peut remarquer que :

$$\begin{aligned} r(\beta, \beta^*) &= \frac{Z_\beta}{Z_{\beta^*}} = \frac{1}{Z_{\beta^*}} \sum_{z \in E} \exp\{-\beta H(z)\} \\ &= \sum_{z \in E} \frac{\exp\{-\beta H(z)\}}{\exp\{-\beta^* H(z)\}} \frac{1}{Z_{\beta^*}} \exp\{-\beta^* H(z)\} \\ &= \sum_{z \in E} \exp\{(\beta^* - \beta) H(z)\} \mu_{\beta^*}(z) \\ &= \mathbb{E}_{\beta^*} [\exp\{(\beta^* - \beta) H(X)\}] \quad \text{avec } X \sim \mu_{\beta^*}. \end{aligned}$$

On dispose donc d'un estimateur d'échantillonnage préférentiel du ratio  $r(\beta, \beta^*)$  à condition de savoir simuler suivant  $\mu_{\beta^*}$ . Or nous avons vu dans le Chapitre 2 comment simuler un échantillon suivant cette mesure. On obtient alors l'estimateur de Monte-Carlo suivant de  $r(\beta, \beta^*)$

$$\hat{r}_k = \frac{1}{k} \sum_{j=1}^k \exp\{(\beta^* - \beta)H(y^{(j)})\} \quad \text{avec } k \in \mathbb{N} \quad \text{et} \quad y^{(1)}, \dots, y^{(k)} \stackrel{iid}{\sim} \mu_{\beta^*}.$$

Le cas  $k = 1$  correspond à l'algorithme d'échange proposé par Murray *et al.* (2012). Il s'implémente de la façon suivante pour un estimateur avec  $k$  observations :

---

**Algorithme 8 :** Algorithme d'échange bruité

---

**Données :**  $\beta^{(0)}$ , et  $k$  et  $n$  des entiers,  $x^{\text{obs}} \in E$   
**pour**  $t$  allant de 0 à  $n - 1$  **faire**  
    Générer  $\beta^* \sim Q(\beta^{(t)}, \cdot)$   
    **pour**  $s$  allant de 0 à  $k$  **faire**  
        Générer  $y^{(s)} \sim \mu_{\beta^*}$   
    **fin**  
    Calculer  $\hat{r}_k = \frac{1}{k} \sum_{s=1}^k \exp\{(\beta^* - \beta^{(t)})H(y^{(s)})\}$   
    Générer  $U \sim \mathcal{U}[0, 1]$   
    **si**  $U \leq \min\left(1, \hat{r}_k \cdot \frac{\pi(\beta^* | x^{\text{obs}})}{\pi(\beta^{(t)} | x^{\text{obs}})} \cdot \frac{Q(\beta^*, \beta^{(t)})}{Q(\beta^{(t)}, \beta^*)}\right)$  **alors**  
         $\beta^{(t+1)} \leftarrow \beta^*$   
    **sinon**  
         $\beta^{(t+1)} \leftarrow \beta^{(t)}$   
    **fin**  
**fin**  
**retourner**  $\beta^{(0)}, \dots, \beta^{(n)}$

---

### Application

On reprend l'application de la Section 3.1 sur le modèle d'Ising. On prend comme loi *a priori* sur  $\beta$  la loi uniforme  $\mathcal{U}([0, b])$  avec  $b = 1$ . La loi *a posteriori* est alors

$$\pi(\beta | x^{\text{obs}}) \propto \mu(\beta | x^{\text{obs}}) \mathbb{1}_{\{\beta \in [0, 1]\}}.$$

La loi *a posteriori* est égale à une constante près à la vraisemblance tronquée à  $[0, 1]$ . Ainsi,

$$\operatorname{argmax}_{\beta \in \mathbb{R}_+} \pi(\beta | x^{\text{obs}}) = \operatorname{argmax}_{\beta \in \mathbb{R}_+} \mu(\beta | x^{\text{obs}}) \mathbb{1}_{\{\beta \in [0, 1]\}}.$$

L'estimateur du maximum *a posteriori* coïncide avec l'estimateur du maximum de vraisemblance si celui-ci appartient à  $[0, 1]$ .

Pour implémenter l'algorithme d'échange on peut choisir différentes lois de propositions. En voici deux exemples.

**Choix 1 :** Une loi de proposition indépendante de l'état précédent :  $Q(\beta, \cdot) = \pi(\cdot)$ .

**Choix 2 :** Une loi de proposition qui dépend de l'état précédent :  $Q(\beta, \cdot) \sim \mathcal{N}(\beta, \sigma)$  avec  $\sigma \in \mathbb{R}_+^*$ .

Comme ces lois de proposition sont symétriques on a  $Q(\beta, \beta^*) = Q(\beta^*, \beta)$  et donc :

$$\frac{\pi(\beta^* | x^{\text{obs}})}{\pi(\beta | x^{\text{obs}})} \cdot \frac{Q(\beta^*, \beta)}{Q(\beta, \beta^*)} = \exp(-(\beta^* - \beta)H(x^{\text{obs}})) \frac{Z_\beta}{Z_{\beta^*}}.$$

Les Figures 3.2 et 3.3 représentent les résultats obtenus après 5000 itérations de l'Algorithme 8, pour  $k = 1$  et  $k = 100$  respectivement, sur un modèle d'Ising avec une grille régulière de taille  $30 \times 20$ . Pour simuler les variables auxiliaires suivant  $\mu_{\beta^*}$  on utilise l'algorithme de Metropolis-hastings avec une période de chauffe de

$30 \times 20 \times 1000$  itérations. En effet, dans la Section 2.4 nous avons vu que ce nombre d'itérations était suffisant. Puis, une fois cette période de chauffe écoulée, pour avoir des réalisations de  $\mu_\beta$  on conserve les états de la chaîne obtenus avec un pas de 100. On observe qu'après une période de chauffe inférieure à 1000 itérations, la chaîne construite dans l'Algorithme 8 semble avoir atteint son état stationnaire. On conserve alors seulement les réalisations de  $\beta$  obtenues après la 1000<sup>me</sup> itération. On remarque que la loi *a posteriori* se concentre bien autour de l'estimateur du maximum de vraisemblance  $\hat{\beta} = 0.34$  (cf. Section 3.1). La moyenne *a posteriori* est :

$$\delta(x^{\text{obs}}) = \begin{cases} 0.336 & \text{pour } k = 1, \\ 0.336 & \text{pour } k = 100. \end{cases}$$

Et l'estimateur du maximum *a posteriori* est :

$$\hat{\beta}^{\text{MAP}} = \begin{cases} 0.341 & \text{pour } k = 1, \\ 0.310 & \text{pour } k = 100. \end{cases}$$

On constate que ces estimateurs sont cohérents avec l'estimateur du maximum de vraisemblance.

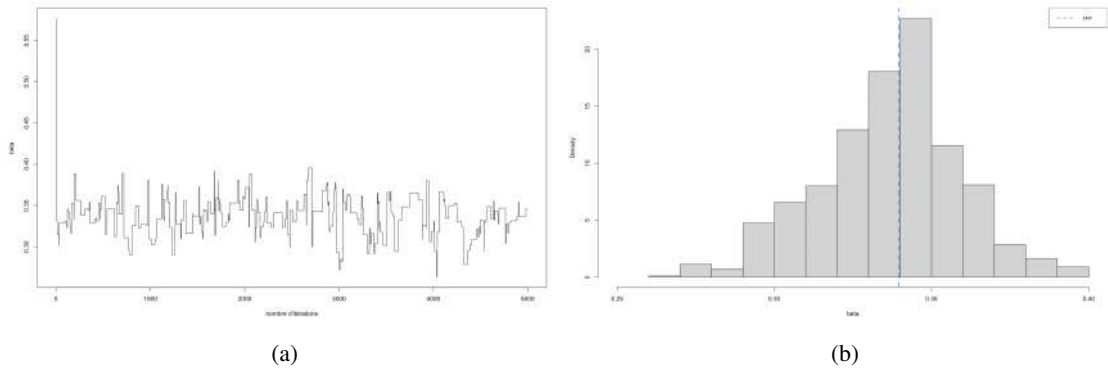


FIGURE 3.2 – Modèle d'Ising sur une grille régulière de taille  $30 \times 20$ . Observation simulée pour  $\beta = 0.3$ . (a) Trajectoire de la chaîne de Markov produite par l'Algorithme 8 ( $k = 1$ ) pour  $Q(\beta, \cdot) \sim \mathcal{U}([0, 1])$ . (b) Distribution de l'échantillon associé. La ligne verticale représente l'estimateur du maximum de vraisemblance.

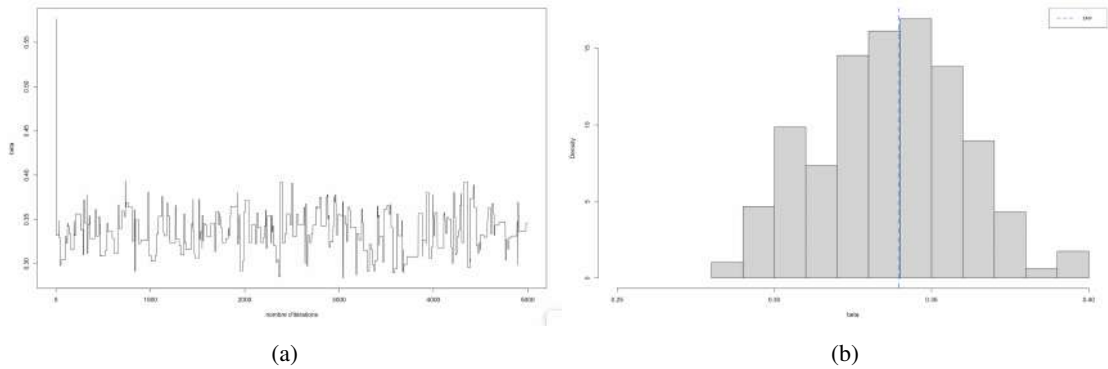


FIGURE 3.3 – Modèle d'Ising sur une grille régulière de taille  $30 \times 20$ . Observation simulée pour  $\beta = 0.3$ . (a) Trajectoire de la chaîne de Markov produite par l'Algorithme 8 ( $k = 100$ ) pour  $Q(\beta, \cdot) \sim \mathcal{U}([0, 1])$ . (b) Distribution de l'échantillon associé. La ligne verticale représente l'estimateur du maximum de vraisemblance.

## Conclusion

Dans ce mémoire, nous avons étudié la mesure de Gibbs, issue d'une famille de lois paramétrées, et présenté plusieurs exemples d'applications propres à la physique statistique. L'échantillonnage suivant cette mesure et l'inférence de son paramètre se sont avérés être délicats, la constante de normalisation rendant la densité difficile à calculer.

Dans un premier temps, pour simuler la mesure de Gibbs, la méthode du rejet s'est naturellement imposée. Cependant, la forte dépendance de l'efficacité de l'algorithme au cardinal de l'ensemble  $E$  nous a encouragées à considérer des méthodes d'échantillonnage plus adaptées. Grâce au théorème de Metropolis-Hastings, nous avons simulé plusieurs mesures de Gibbs en implémentant trois algorithmes MCMC : les algorithmes de Metropolis-Hastings, de Gibbs, et du recuit simulé. Cependant, ils ne fournissent qu'une simulation approchée suivant la mesure de Gibbs. Nous nous sommes donc tournées vers l'algorithme de Propp-Wilson, un algorithme de couplage permettant d'obtenir un échantillonnage exact. La comparaison de ces méthodes a montré que, bien qu'étant uniquement asymptotiquement exactes, les méthodes MCMC produisent des échantillons proches (en terme de distribution empirique) de ceux produits par Propp-Wilson si le nombre d'itérations est suffisant. Nous avons pleinement conscience que nous ne connaissons pas au préalable le nombre d'itérations nécessaires à l'obtention d'une bonne approximation de la mesure stationnaire. C'est un point que nous aurions aimé approfondir, mais que nous n'avons pas eu le temps d'aborder dans ce mémoire.

Enfin, nous avons consacré le troisième chapitre de ce mémoire à l'inférence du paramètre  $\beta$  de la mesure de Gibbs. Nous avons tout d'abord étudié l'approche fréquentiste qui consiste à maximiser la vraisemblance et utilisé à cet effet une méthode de descente de gradient. Par la suite, nous nous sommes intéressées à l'inférence bayésienne, et avons simulé la loi *a posteriori* à l'aide d'un algorithme d'échange. Cependant, aussi bien dans l'approche fréquentiste que bayésienne, nous avons utilisé des estimateurs de Monte-Carlo afin d'estimer des quantités que la fonction de partition ne nous permettait pas de calculer. Ces estimateurs nécessitant de simuler des réalisations de la mesure de Gibbs, nous avons mis en application les algorithmes développés dans le deuxième chapitre. Nous avons pu constater en appliquant l'algorithme d'échange au modèle d'Ising que l'estimateur bayésien et l'estimateur du maximum *a posteriori* étaient très proches de l'estimateur du maximum de vraisemblance. Finalement, il serait intéressant d'étudier l'influence du nombre d'observations dans le calcul des estimateurs de Monte-Carlo sur l'efficacité des algorithmes.

## Bibliographie

- P. ALQUIER, N. FRIEL, R. EVERITT et A. BOLAND : Noisy Monte Carlo : Convergence of Markov chains with approximate transition kernels. *Statistical Computing*, 26(1):29–49, 2016.
- D. CHAFAI et F. MALRIEU : *Recueil de modèles aléatoires*. 2018.
- S. CHIB et E. GREENBERG : Understanding the Metropolis-Hastings Algorithm. *The American Statistician*, 49 (4):327–335, 1995.
- S. GEMAN et D. GEMAN : Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1984.
- F. HUveneERS : Processus discrets, Cours sur les chaînes de Markov. Université Paris Dauphine-PSL, 2021.
- N. METROPOLIS, A. W. ROSENBLUTH, M. N. ROSENBLUTH, A. H. TELLER et E. TELLER : Equation of State Calculations by Fast Computing Machines. , 21(6):1087–1092, 1953.
- I. MURRAY, Z. GHahRAMANI et D. MACKay : MCMC for doubly-intractable distributions, 2012.
- J. G. PROPP et D. B. WILSON : Exact Sampling with Coupled Markov Chains and Applications to Statistical Mechanics. *Random Structures Algorithms*, 9(1-2):223–252, 1996.
- H. ROBBINS et S. MONRO : A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, 22 (3):400–407, 1951.
- C. P. ROBERT et G. CASELLA : *Monte Carlo Statistical Methods*. 2004.
- E. VIGODA : Markov Chain Monte Carlo Methods, Lecture Notes : Coupling from the past. Georgia Tech, 2006.

# Annexe

## Lien d'accès vers les codes

<https://github.com/bstnmaria/Mesure-de-Gibbs-et-algorithmes-MCMC>

## Implémentation des algorithmes dans R

### Modèle du Composé chimique

```
verif_e <- function(x) {  
  # fonction qui verifie si la matrice x appartient à E  
  return(sum(x[-nrow(x), ] * x[-1, ] == -1) + sum(x[, -ncol(x)] * x[, -1] == -1) == 0)  
}  
  
verif_e_loc <- function(x, i, j) {  
  # fonction qui vérifie si la contrainte de E est vérifiée pour le site i de x  
  #qui prend la valeur j et ses voisins  
  return(sum(j * voisins(x, i) == -1) == 0)  
}  
  
voisins <- function(x, i) {  
  # fonction qui renvoie les voisins du site i de la matrice x  
  h <- nrow(x)  
  v <- rep(2, 4)  
  if (i %% h != 1) {  
    v[1] <- x[i - 1]  
  }  
  if (i %% h != 0) {  
    v[2] <- x[i + 1]  
  }  
  if (i > h) {  
    v[3] <- x[i - h]  
  }  
  if (i < length(x) - h + 1) {  
    v[4] <- x[i + h]  
  }  
  return(v[v != 2])  
}  
  
hamiltonien <- function(x) {  
  # hamiltonien
```



```

    return(sum(abs(x)))
}

prob_cond <- function(x, i, beta) {
  # calcul des probas conditionnelles
  p <- exp(-beta * c(1, 0, 1))
  p <- p * (colSums(voisins(x, i) %*% t(-1:1) == -1) == 0)
  return(p / sum(p))
}

mapping <- function(x, n_iter, j, v, u, beta, uniform = TRUE) {
  test <- TRUE
  for (i in 1:n_iter) {
    if (uniform) {
      test <- u[i] <= exp(-beta * (abs(v[i]) - abs(x[j[i]])))
    }
    if (test & verif_e_loc(x, j[i], v[i])) {
      x[j[i]] <- v[i]
    }
  }
  return(x)
}

rinit_cc <- function(h, w) {
  x <- matrix(rep(0, h * w), h)
  i <- 0
  while (i < h) {
    x[i, ] <- rep(sample(-1:1, 1, replace = TRUE, prob = c(2 / 5, 1 / 5, 2 / 5)), w)
    i <- i + 2
  }
  return(x)
}

rejet_cc <- function(x) {
  # fonction rejet
  p <- exp(-beta * c(1, 0, 1))
  p <- p / sum(p)
  while (verif_e(x) == FALSE) {
    x <- matrix(sample(-1:1, length(x), replace = TRUE, prob = p), nrow = nrow(x))
  }
  return(x)
}

metropolis_hastings_cc <- function(n_iter, beta, uniform = FALSE, x_0) {
  # fonction metropolis hastings
  if (uniform) {
    p <- rep(1 / 3, 3)
  } else {
    p <- exp(-beta * c(1, 0, 1))
    p <- p / sum(p)
  }
  j <- sample(seq_len(length(x_0)), n_iter, replace = TRUE)
  v <- sample(-1:1, n_iter, prob = p, replace = TRUE)
  u <- runif(n_iter)
  return(mapping(x_0, n_iter, j, v, u, beta, uniform))
}

```

```

}

gibbs_cc <- function(n_iter, beta, random, x_0) {
  # fonction gibbs
  taille <- length(x_0)
  u <- 1:taille
  for (j in 1:n_iter) {
    if (random) {
      # random scan
      u <- sample(1:taille, taille)
    }
    for (i in u) {
      x_0[i] <- sample(-1:1, 1, prob = probb_cond(x_0, i, beta))
    }
  }
  return(x_0)
}

propp_wilson_cc <- function(pas, beta, h, w) {
  # fonction propp wilson
  taille <- h * w
  m1 <- matrix(rep(-1, taille), h)
  m2 <- matrix(rep(1, taille), h)
  nb_iter <- 0
  j <- c()
  v <- c()
  u <- c()
  p <- exp(-beta * c(1, 0, 1))
  p <- p / sum(p)
  while (identical(m1, m2) == FALSE) {
    nb_iter <- nb_iter + pas
    m1 <- matrix(rep(-1, taille), h)
    m2 <- matrix(rep(1, taille), h)
    j <- c(sample(seq_len(taille), pas, replace = TRUE), j) # choix du site
    v <- c(sample(-1:1, pas, prob = p, replace = TRUE), v)
    # choix de la nouvelle valeur du site
    u <- c(runif(pas), u)
    m1 <- mapping(m1, nb_iter, j, v, u, beta, FALSE)
    m2 <- mapping(m2, nb_iter, j, v, u, beta, FALSE)
  }
  return(m1)
}

```

## Modèle d'Ising

```

voisins <- function(x, i) {
  # fonction qui renvoie les voisins du site i de la matrice x
  h <- nrow(x)
  v <- rep(0, 4)
  if (i %% h != 1) {
    v[1] <- x[i - 1]
  }
  if (i %% h != 0) {
    v[2] <- x[i + 1]
  }
}

```

```

    if (i > h) {
      v[3] <- x[i - h]
    }
    if (i < length(x) - h + 1) {
      v[4] <- x[i + h]
    }
    return(v[v != 0])
  }

hamiltonien <- function(x) {
  # hamiltonien
  return(-sum(x[-1, ] * x[-nrow(x), ]) - sum(x[, -1] * x[, -ncol(x)]))
}

prob_cond <- function(x, i, beta) {
  # calcul des probas conditionnelles
  p <- exp(beta * c(-1, 1) * sum(voisins(x, i)))
  return(p / sum(p))
}

mapping <- function(x, n_iter, j, v, u, beta) {
  for (i in 1:n_iter) {
    if (u[i] <= exp(beta * sum(voisins(x, j[i]) * (v[i] - x[j[i]])))) {
      x[j[i]] <- v[i]
    }
  }
  return(x)
}

mapping_h <- function(x, n_iter, j, v, u, beta) {
  # fonction mapping qui renvoie le hamiltonien de chaque iteration
  hamil <- numeric(n_iter)
  hamil[1] <- hamiltonien(x)
  for (i in 2:n_iter) {
    temp <- sum(voisins(x, j[i]) * (v[i] - x[j[i]]))
    if (u[i] <= exp(beta * temp)) {
      hamil[i] <- hamil[i - 1] - temp
      x[j[i]] <- v[i]
    } else {
      hamil[i] <- hamil[i - 1]
    }
  }
  return(hamil)
}

rinit_ising <- function(h, w) {
  # initialisation d'une matrice dans E
  return(matrix(sample(c(-1, 1), h * w, replace = TRUE), h))
}

metropolis_hastings_ising <- function(n_iter, beta, x_0) {
  # fonction metropolis hastings
  j <- sample(seq_len(length(x_0)), n_iter, replace = TRUE) # choix du site
  v <- sample(c(-1, 1), n_iter, replace = TRUE) # choix de la nouvelle valeur du site
  u <- runif(n_iter)

```

```

    return(mapping(x_0, n_iter, j, v, u, beta))
}

gibbs_ising <- function(n_iter, beta, random, x_0) {
  # fonction gibbs
  taille <- length(x_0)
  u <- 1:taille
  for (j in 1:n_iter) {
    if (random) {
      # random scan
      u <- sample(1:taille, taille)
    }
    for (i in u) {
      x_0[i] <- sample(c(-1, 1), 1, prob = probab_cond(x_0, i, beta))
    }
  }
  return(x_0)
}

propp_wilson_ising <- function(pas, beta, h, w) {
  # fonction propp wilson
  taille <- h * w
  m1 <- matrix(rep(-1, taille), h)
  m2 <- matrix(rep(1, taille), h)
  n_iter <- 0
  j <- c()
  v <- c()
  u <- c()
  while (identical(m1, m2) == FALSE) {
    n_iter <- n_iter + pas
    m1 <- matrix(rep(-1, taille), h)
    m2 <- matrix(rep(1, taille), h)
    j <- c(sample(seq_len(taille), pas, replace = TRUE), j) # choix du site
    v <- c(sample(c(-1, 1), pas, replace = TRUE), v)
    # choix de la nouvelle valeur du site
    u <- c(runif(pas), u)
    m1 <- mapping(m1, n_iter, j, v, u, beta)
    m2 <- mapping(m2, n_iter, j, v, u, beta)
  }
  return(m1)
}

metropolis_hastings_ising_h <- function(n_iter, beta, x_0) {
  # fonction metropolis hastings qui renvoie le hamiltonien de chaque iteration
  j <- sample(seq_len(length(x_0)), n_iter, replace = TRUE) # choix du site
  v <- sample(c(-1, 1), n_iter, replace = TRUE) # choix de la nouvelle valeur du site
  u <- runif(n_iter)
  return(mapping_h(x_0, n_iter, j, v, u, beta))
}

gibbs_ising_h <- function(n_iter, beta, random, x_0) {
  # fonction gibbs qui renvoie le hamiltonien de chaque iteration
  taille <- length(x_0)
  u <- 1:taille
  hamil <- numeric(n_iter)

```

```

for (j in 1:n_iter) {
  if (random) { # random scan
    u <- sample(1:taille, taille)
  }
  for (i in u) {
    x_0[i] <- sample(c(-1, 1), 1, prob = prob_cond(x_0, i, beta))
  }
  hamil[j] <- hamiltonien(x_0)
}
return(hamil)
}

propp_wilson_ising_h <- function(pas, n_iter, beta, h, w) {
  # fonction propp wilson qui renvoie le hamiltonien de chaque iteration
  taille <- h * w
  m1 <- matrix(rep(-1, taille), h)
  m2 <- matrix(rep(1, taille), h)
  n_iter <- 0
  j <- c()
  v <- c()
  u <- c()
  while (identical(m1, m2) == FALSE) {
    n_iter <- n_iter + pas
    m1 <- matrix(rep(-1, taille), h)
    m2 <- matrix(rep(1, taille), h)
    j <- c(sample(seq_len(taille), pas, replace = TRUE), j)
    v <- c(sample(c(-1, 1), pas, replace = TRUE), v)
    u <- c(runif(pas), u)
    m1 <- mapping(m1, n_iter, j, v, u, beta)
    m2 <- mapping(m2, n_iter, j, v, u, beta)
  }
  j <- sample(seq_len(taille), n_iter, replace = TRUE)
  v <- sample(c(-1, 1), n_iter, replace = TRUE)
  u <- runif(n_iter)
  return(mapping_h(m1, n_iter, j, v, u, beta))
}

h_sample <- function(h, w, beta, n, n_burn, lag, sampler = "gibbs") {
  # retourne un echantillon du hamiltonien de realisations du modele
  if (sampler == "gibbs") {
    # en utilisant gibbs
    keep <- n_burn + seq(lag, (n * lag), lag)
    init <- rinit_ising(h, w)
    hamil <- gibbs_ising_h(n_burn + n * lag, beta, FALSE, init)
  } else if (sampler == "mh") {
    # en utilisant metropolis hastings
    keep <- n_burn + seq(lag, (n * lag), lag)
    init <- rinit_ising(h, w)
    hamil <- metropolis_hastings_ising_h(n_burn + n * lag, beta, init)
  } else {
    # en utilisant propp wilson
    keep <- seq(lag, (n * lag), lag)
    pas <- 1000
    hamil <- propp_wilson_ising_h(pas, n * lag, beta, h, w)
  }
}

```

```

    return(hamil[keep])
}

descente_gradient <- function(epsilon, b_0, x_obs, n_MC, n_burn, lag, sampler) {
  # descente de gradient
  h <- nrow(x_obs)
  w <- ncol(x_obs)
  h_obs <- hamiltonien(x_obs)
  hamil <- h_sample(h, w, b_0, n_MC, n_burn, lag, sampler)
  grad <- mean(hamil) - h_obs
  b <- c(b_0, b_0 + grad / abs(max(hamil)))
  i <- 2
  while (abs(grad) > epsilon) {
    hamil <- h_sample(h, w, b[i], n_MC, n_burn, lag, sampler)
    grad <- mean(hamil) - h_obs
    b <- c(b, b[i] + grad / (i * abs(max(hamil))))
    i <- i + 1
  }
  return(b)
}

algo_echange <- function(x_obs, n_iter, n_burn, n_MC, d, b_0 = runif(1, max = d),
lag = 100, sampler = "gibbs") {
  # fonction algo echange
  h <- nrow(x_obs)
  w <- ncol(x_obs)
  h_obs <- hamiltonien(x_obs)
  b <- c(b_0, runif(n_iter - 1, 0, d))
  u <- runif(n_iter)
  for (i in 2:n_iter) {
    hamil <- h_sample(h, w, beta, n_MC, n_burn, lag, sampler)
    est_mc <- mean(exp(hamil * (b[i] - b[i - 1])))
    if (u[i] > est_mc * exp(h_obs * (b[i - 1] - b[i]))) {
      b[i] <- b[i - 1]
    }
  }
  return(b)
}

```

## Modèle de Potts

```

voisins <- function(x, i) {
  # fonction qui renvoie les voisins du site i de la matrice x
  h <- nrow(x)
  v <- rep(-1, 4)
  if (i %% h != 1) {
    v[1] <- x[i - 1]
  }
  if (i %% h != 0) {
    v[2] <- x[i + 1]
  }
  if (i > h) {
    v[3] <- x[i - h]
  }
  if (i < length(x) - h + 1) {

```

```

    v[4] <- x[i + h]
  }
  return(v[v != -1])
}

hamiltonien <- function(x) {
  # hamiltonien
  return(-sum(x[-1, ] == x[-nrow(x), ]) - sum(x[, -1] == x[, -ncol(x)]))
}

prob_cond <- function(x, i, beta, k) {
  # calcul des probas conditionnelles
  p <- numeric(k)
  vois <- voisins(x, i)
  for (j in 1:k) {
    p[j] <- exp(beta * sum(vois == j))
  }
  return(p / sum(p))
}

mapping <- function(x, n_iter, j, v, u, beta) {
  for (i in 1:n_iter) {
    vois <- voisins(x, j[i])
    if (u[i] <= exp(beta * sum((vois == v[i]) - (vois == x[j[i]])))) {
      x[j[i]] <- v[i]
    }
  }
  return(x)
}

rinit_potts <- function(h, w, k) {
  # initialisation d'une matrice dans E
  return(matrix(sample(1:k, h * w, replace = TRUE), h))
}

metropolis_hastings_potts <- function(n_iter, beta, k, x_0) {
  # fonction metropolis hastings
  j <- sample(seq_len(length(x_0)), n_iter, replace = TRUE) # choix du site
  v <- sample(1:k, n_iter, replace = TRUE) # choix de la nouvelle valeur du site
  u <- runif(n_iter)
  return(mapping(x_0, n_iter, j, v, u, beta))
}

gibbs_potts <- function(n_iter, beta, k, random, x_0) {
  # fonction gibbs
  taille <- length(x_0)
  u <- 1:taille
  for (j in 1:n_iter) {
    if (random) { # random scan
      u <- sample(1:taille, taille)
    }
    for (i in u) {
      x_0[i] <- sample(1:k, 1, prob = prob_cond(x_0, i, beta, k))
    }
  }
}

```

```

    return(x_0)
}

```

## Sudoku

```

empty_cells <- function(x) {
  # indice des cases vides par carre
  empty_cells <- which(x == 0)
  col <- (empty_cells - 1) %/% 9
  row <- (empty_cells - 1) %/% 9
  return(split(empty_cells, col %/% 3 * 3 + row %/% 3))
}

missing_values <- function(x) {
  # valeurs manquantes par carre
  given_cells <- which(x > 0)
  col <- (given_cells - 1) %/% 9
  row <- (given_cells - 1) %/% 9
  given_cells <- split(given_cells, col %/% 3 * 3 + row %/% 3)
  return(sapply(1:9, function(i) {
    setdiff(1:9, x[given_cells[[i]])
  })))
}

hamiltonien <- function(x) {
  # hamiltonien : nombre de replicats par lignes et colonnes
  s <- 0
  for (i in 1:9) {
    s <- s + sum(duplicated(x[i, ]) + duplicated(x[, i]))
  }
  return(s)
}

recuit_simule_sudoku <- function(s, n_iter, T_) {
  # fonction recuit simule sudoku
  empty <- empty_cells(s)
  missing <- missing_values(s)
  for (i in 1:9) {
    # remplissage de la grille
    s[empty[[i]]] <- sample(missing[[i]])
  }
  h_old <- hamiltonien(s)
  j <- sample(1:9, n_iter, replace = TRUE) # on choisit un carre uniformement
  u <- runif(n_iter)
  for (i in 1:n_iter) {
    T_ <- T_ * 0.99999
    if (h_old == 2) {
      # on remonte la temperature car min local
      T_ <- 0.5
    }
    s_new <- s
    z <- sample(empty[[j[i]]], 2) # on choisit 2 cases a echanger
    s_new[z] <- s_new[z] %*% matrix(c(0, 1, 1, 0), 2) # on echange
    h_new <- hamiltonien(s_new)
    if (u[i] <= exp((1 / T_) * (h_old - h_new))) {

```



```
s <- s_new
h_old <- h_new
}
if (h_old == 0) {
  # on a trouve la solution
  break
}
}
return(s)
}
```