

Projet de simulations aléatoires : Modèles d'Ising

Aurélien Enfroy, Shmuel Rakotonirina--Ricquebourg

13 janvier 2018

Table des matières

| | | |
|----------|--|-----------|
| 1 | Introduction | 1 |
| 2 | Choix d'implémentation | 2 |
| 3 | Simulation par l'échantillonneur de Gibbs | 3 |
| 4 | Simulation par Metropolis-Hastings | 3 |
| 5 | Couplage par le passé | 4 |
| 6 | Vérification des algorithmes | 6 |
| 7 | Cas en petite taille : simulation naïve et température critique | 8 |
| 8 | Observation de la transition de phase selon la taille du réseau | 9 |
| 9 | Conclusion | 14 |

1 Introduction

Ce document rend compte d'expériences sur les modèles d'Ising sur un réseau carré, avec pour objectifs principaux d'observer la transition de phase à la température critique, en fonction de la taille du réseau. En théorie, à température sur-critique, les états les plus probables ont des spins assez mélangés, alors qu'à température sous-critique, les états les plus probables rassemblent entre eux les spins identiques.

Nous avons fait le choix, pour s'approcher le mieux possible d'un réseau infini, de prendre des conditions aux bords périodiques (par exemple, les points du côté droit du carré sont voisins du côté gauche).

Pour l'observation de la température critique (réseau de grande taille), nous avons implémenté l'échantillonneur de Gibbs (section 3), l'algorithme de

Metropolis-Hastings (section 4), et deux méthodes de couplage par le passé, chacune basée sur une des méthodes précédentes (section 5). La section 6 présente la magnétisation et le test qui en découle pour vérifier que ces algorithmes fonctionnent.

En très petite taille (au plus 16 sommets sur le graphe), nous avons aussi implémenté la méthode naïve (section 7) pour observer à quel point les états $\pm(1, \dots, 1)$ étaient probable (et donc "résistants" aux transitions de Metropolis-Hastings).

Enfin, la section 8 présente les observations des transitions de phase autour de la température critique théorique, pour différentes tailles de réseau.

2 Choix d'implémentation

On rappelle la définition du modèle d'Ising sur un réseau carré :

Définition 2.1. On fixe C le réseau carré de dimension 2 de taille N^2 . Le modèle d'Ising est la distribution sur l'espace d'état $\{\pm 1\}^C$ dont la loi est donnée par

$$\forall x \in \{\pm 1\}^C, \pi(x) = \frac{1}{Z_T} \exp \left(\left(\sum_{u \sim v} J_{u,v} x_u x_v + \sum_u h_u x_u \right) / T \right)$$

où $T > 0$ est appelée la température, Z_T est une constante de normalisation, $J_{u,v}$ est la force d'interaction entre u et v et h_u est le champ magnétique extérieur en u .

Pour l'implémentation, on remarque qu'il n'y a pas besoin du paramètre T , qu'on peut inclure dans J et h . On représente alors ces paramètres en prenant $x \in \mathcal{M}_{N,N}(\{\pm 1\})$, $\tilde{h} = h/T \in \mathcal{M}_{N,N}(\mathbb{R})$ et J comme une matrice à trois entrées $\tilde{J} \in \mathcal{M}_{N,N,2}(\mathbb{R})$ où

$$\tilde{J}_{i,j,1} = J_{(i,j),(i+1,j)}/T \text{ et } \tilde{J}_{i,j,2} = J_{(i,j),(i,j+1)}/T$$

où $i+1$ et $j+1$ sont à comprendre modulo N pour obtenir des conditions aux bords périodiques.

Dans le but de créer le script `animations.sce` (qui montre en animation l'évolution des différentes chaînes), nous avons dû stocker de très grandes matrices (à $N^2 n$ entrées, où n est le temps d'évolution des chaînes). Cependant, le type par défaut `double` stocke les données sur beaucoup plus de bits que nécessaire, causant donc des erreurs d'allocation mémoire quand les chaînes mémorisées devenaient trop longues et les matrices trop grandes. Nous avons donc fait le choix de stocker tous les spins (donc toutes les variables ne pouvant prendre que les valeurs ± 1) sur 8 bits (type `int8`, qui est le type pouvant stocker ± 1 prenant le moins de mémoire), ce qui nécessite

de changer des `double` en `int8` et réciproquement pour certains calculs (d'où l'apparition répétée dans nos codes des fonctions `double` et `int8`).

Il aurait été possible de prendre encore moins de mémoire en remplaçant les ± 1 par des booléens `%t` et `%f`, en remplaçant donc les `*` par des `xor`. Mais cela rendait le code moins lisible, notamment quand on voulait multiplier un booléen par un `double` (par exemple pour calculer $J_{u,v}x_u x_v$, où J est stocké en `double` et x en booléen). De plus, nous n'avons pas eu de problème mémoire avec le type `int8`, il n'a donc pas été nécessaire de faire plus d'économie.

3 Simulation par l'échantillonneur de Gibbs

En reprenant les notations du cours, on a pour $u \in C$ et $x \in \{\pm 1\}^C$, les transitions coordonnées par coordonnées suivantes

$$\begin{aligned}\pi_u(x_u \mid x^u) &= \frac{\pi(x_u, x^u)}{\pi(1, x^u) + \pi(-1, x^u)} \\ &= \frac{e^{x_u V_u(x)}}{e^{V_u(x)} + e^{-V_u(x)}}\end{aligned}$$

où on note encore $V_u(x) \doteq \left(\sum_{v \sim u} J_{u,v} x_v + h_u \right) / T$. Donc $\pi_u(\cdot \mid x^u) = \mathcal{B}\left(\frac{e^{V_u(x)}}{e^{V_u(x)} + e^{-V_u(x)}}\right) = B\left(\frac{1}{1 + e^{-2V_u(x)}}\right)$. L'échantillonneur de Gibbs consiste à actualiser une entrée de la matrice en laissant inchangé le reste de la matrice et en donnant à cette entrée un spin tiré selon la loi $\pi_u(\cdot \mid x^u)$.

Nous avons implémenté ici les deux versions de l'échantillonneur de Gibbs :

- Le balayage séquentiel : à chaque itération de l'algorithme, on actualise toutes les entrées, dans un ordre fixé.
- Le balayage aléatoire : à chaque itération de l'algorithme, on ne modifie qu'une entrée, choisie uniformément sur tout le réseau.

Pour obtenir des temps d'exécutions similaires, et pour équilibrer les vitesses à laquelle les algorithmes convergent, pour chaque itération de l'échantillonneur de Gibbs à balayage séquentiel, on effectue N^2 itérations de celui aléatoire.

4 Simulation par Metropolis-Hastings

Nous avons implémenté l'algorithme de Metropolis-Hastings en utilisant la fonction de rejet de Metropolis-Hastings. Pour le choix du noyau instrumental Q , nous avons choisi de donner à une coordonnée aléatoire (uniforme) une valeur aléatoire (uniforme), ce qui permet d'avoir un noyau symétrique irréductible apériodique (probabilité strictement positive de rester sur place).

Avec ce noyau instrumental, on peut réduire les calculs effectués : si x est un état et y l'état proposé à partir de x , alors

- soit $y = x$, auquel cas on peut considérer qu'on accepte le mouvement,
- soit $y \neq x$, auquel cas ils ne diffèrent que d'au plus une coordonnée (celle tirée uniformément). Si on note u cette coordonnée et s la nouvelle valeur, alors $s = y_u = -x_u$ et, en considérant que T a été pris en compte dans J et h ,

$$\begin{aligned} \frac{\pi(y)}{\pi(x)} &= \frac{\exp \left(\left(\sum_{v \sim u} J_{u,v} s x_v + h_u s \right) / T \right)}{\exp \left(\left(\sum_{v \sim u} J_{u,v} (-s) x_v + h_u (-s) \right) / T \right)} \\ &= \frac{e^{s V_u(x)}}{e^{-s V_u(x)}} \\ &= e^{2s V_u(x)} \end{aligned}$$

où on a noté $V_u(x) \doteq \left(\sum_{v \sim u} J_{u,v} x_v + h_u \right) / T$.

Dans le cas de l'échantillonneur de Gibbs et de Metropolis-Hastings, le temps passé pour les actualisations est presque entièrement dû aux calculs des $V_u(x)$ et des exponentielles. Cette disjonction de cas donne donc un avantage à l'algorithme de Metropolis-Hastings sur l'échantillonneur de Gibbs : dans la moitié des cas, il n'y a pas de calcul à faire pour Metropolis-Hastings. Cela se traduit par un temps d'exécution deux fois plus court en moyenne pour Metropolis-Hastings.

5 Couplage par le passé

Pour le couplage par le passé, deux fonctions d'actualisation sont possibles.

- celle vue en cours, issue de l'échantillonneur de Gibbs (par balayage séquentiel)
- celle de [1], issue de l'algorithme de Metropolis-Hastings (i.e. même transition qu'en section 4).

Dans les deux cas, la méthode n'a de garantie théorique que si J est constant et h nul. On supposera donc cela vérifié.

L'algorithme consiste à répéter les algorithmes (Metropolis Hastings ou échantillonneur de Gibbs), vers le futur, en partant des états $\pm(1, \dots, 1)$ jusqu'à un temps fixé à l'avance. Si il n'y a pas eu coalition, on ajoute des nouvelles actualisations au début du processus (en conservant les actualisations précédentes à la fin du processus), ce qui demande de recalculer le processus dans son entièreté. Pour obtenir le temps de coalition exact, il faudrait en théorie n'ajouter qu'une actualisation à la fois. Cependant, on ne

cherche pas à obtenir ce temps, on se contente donc de doubler le nombre d'actualisations à chaque échec de coalition.

Pour les deux méthodes (Metropolis-Hastings et échantillonneur de Gibbs), les choses se passent bien au-delà de la température critique. Cependant, à (très) basse température, les états $\pm(1, \dots, 1)$ sont très probables et ont donc tendance à rester inchangés par les fonctions d'actualisation précédentes : dans les deux cas, la probabilité de réussir à changer un point x_u égaux à ses quatre voisins est très faible : si $J > 0$ (toujours supposé constant), alors elle vaut $\exp(-8J/T)$.

En conséquence, l'algorithme met extrêmement longtemps à terminer (nous ne l'avons pas vu terminer) quand la température est trop basse (par rapport à la température critique). En observant l'évolution des chaînes de Markov démarrant de $(1, \dots, 1)$ et de $(-1, \dots, -1)$, on voit que souvent les deux chaînes reviennent à leur état initial, ce qui signifie que l'algorithme pourrait bien continuer de boucler ainsi très longtemps. À titre d'exemple, à $T = 2.2$ (sous-critique mais proche de la température critique) et $N = 50$ (donc 2500 entrées par matrices), on peut noter e le nombre d'entrées par lequel les deux chaînes diffèrent encore après n actualisation de type Metropolis-Hastings, ce qui donne les tables 5. Plus généralement, quand T augmente, on a besoin de moins en moins d'actualisations.

| (a) Cas sous-critique $T = 2.2$ (interrompu au bout de $\sim 1500s$) | | (b) Cas sur-critique $T = 3$ | |
|---|------|---------------------------------|------|
| n | e | n | e |
| 2500 | 2434 | 2500 | 2331 |
| 5000 | 2408 | 5000 | 2193 |
| 10000 | 2353 | 10000 | 1965 |
| 20000 | 2272 | 20000 | 1573 |
| 40000 | 2189 | 40000 | 1028 |
| 80000 | 2130 | 80000 | 567 |
| 160000 | 2088 | 160000 | 103 |
| 320000 | 2066 | 320000 | 0 |
| 640000 | 2066 | | |
| 1280000 | 2066 | | |

TABLE 1 – Tentatives de coalition consécutives de l'algorithme de couplage par le passé par Metropolis-Hastings. Pour passer de n à $2n$, on conserve les n actualisation et on en ajoute n nouvelles avant. e est le nombre d'entrées différentes entre les deux chaînes à la fin de la tentative à n fixé ($e \leq N^2 = 2500$, il y a coalition à $e = 0$)

Pour cette raison, on préférera à cet algorithme les algorithmes précédant, non exacts mais pour lesquels on peut décider à l'avance du temps utilisé par

l'algorithme.

On vérifie quand même que les comportements des différents algorithmes sont similaires au cas sans couplage figure 1, avec des temps d'exécution donnés figure 2. On indique le nombre d'actualisations calculées (appelées *transitions* en figure 1) à côté du temps.

Remarque. — Chaque balayage séquentiel coûte N^2 actualisations. Donc en figure 1, 100 balayages séquentiels correspondent à 250000 transitions.

- Pour les algorithmes de couplage, environ la moitié des actualisations calculées sont utiles. Les autres ont été oubliées après n'avoir pas conduit à une coalition. Par exemple, à $T = 5$ avec Metropolis-Hastings, 80000 actualisations sont utiles, mais l'algorithme a fait $N^2 = 2500$ actualisations, puis 5000, 10000, 20000, 40000, chaque fois sans qu'il y ait coalition, puis 80000 où il y a eu coalition, d'où un total de 157500 actualisations calculées.

| Méthode | $T = 3$ | $T = 5$ |
|------------------------------|----------------|----------------|
| Gibbs séquentiel | 209s [250 000] | 209s [250 000] |
| Gibbs aléatoire | 209s [250 000] | 208s [250 000] |
| Metropolis-Hastings | 115s [250 000] | 98s [250 000] |
| Couplage Metropolis-Hastings | 520s [637 500] | 115s [157 500] |
| Couplage Gibbs séquentiel | 532s [317 500] | 118s [77 500] |

TABLE 2 – Comparaison des temps d'exécution des algorithmes pour $N = 50$. Entre crochets le nombre d'actualisations calculées

Les résultats des algorithmes sont bien similaires pour différentes températures (sur-critiques), et à nombre d'actualisation fixée, on vérifie bien que l'algorithme de Metropolis-Hastings est deux fois plus rapide.

6 Vérification des algorithmes

Pour vérifier que les algorithmes MCMC fonctionnent, on utilise la fonction $M(x) = \frac{1}{N^2} \sum_u x_u$. M est appelée la magnétisation. Dans le cas où $h = 0$, on a $\pi(-x) = \pi(x)$ et $M(-x) = -M(x)$ donc si $X \sim \pi$, alors $\mathbb{E}(M(X)) = 0$. Or la chaîne $(X^{(n)})_n$ (obtenue par Metropolis-Hastings ou échantillonneur de Gibbs) est ergodique, car c'est une chaîne de Markov à espace d'états fini, irréductible et aperiodique. π étant la mesure invariante, on aura donc la convergence presque sûre :

$$\frac{1}{n} \sum_i M(X^{(i)}) \xrightarrow{n \rightarrow +\infty} \int M(x) d\pi(x) = \mathbb{E}(M(X)) = 0$$

En pratique, à basse température, les algorithmes ont tendance à rester longtemps dans les états $(1, \dots, 1)$ et $(-1, \dots, -1)$ (qui sont très probables à

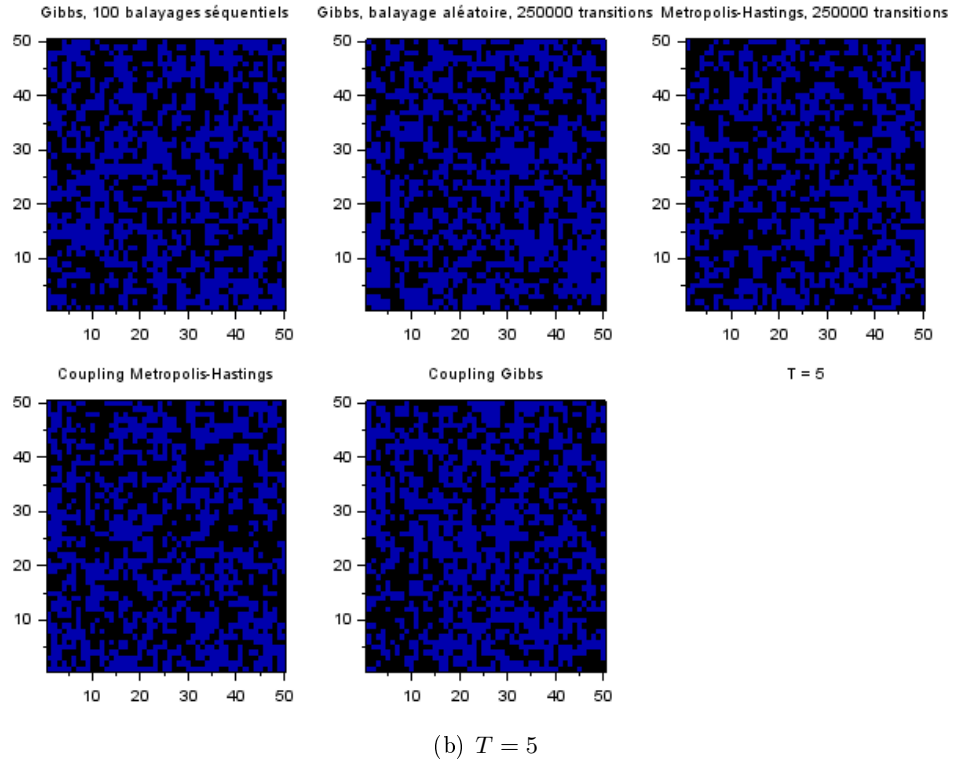
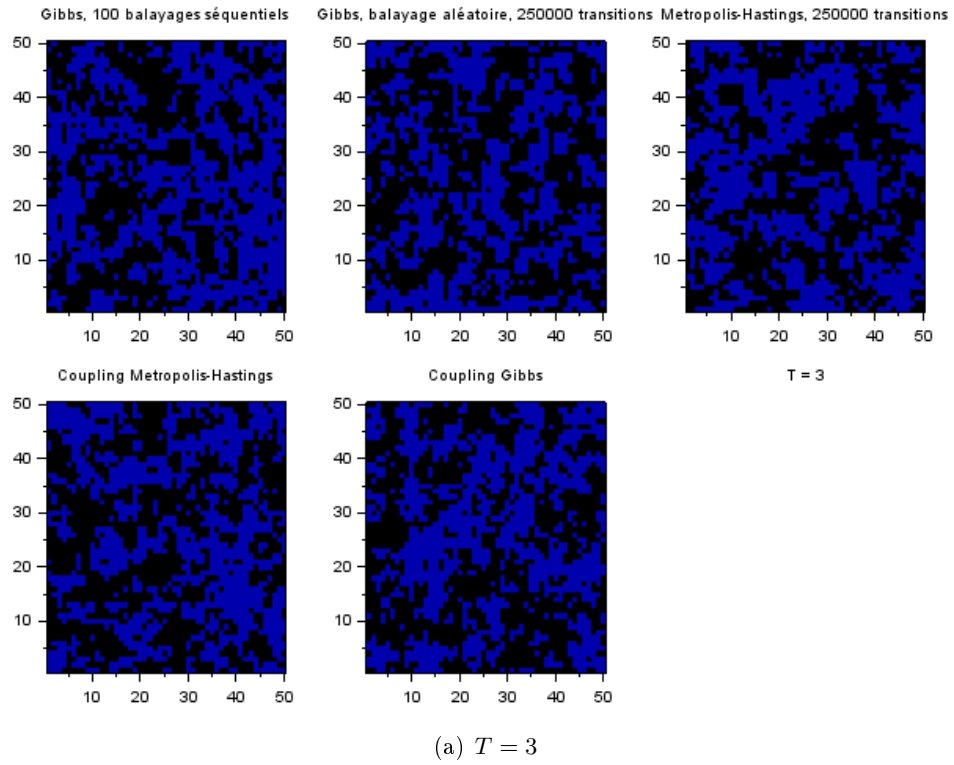


FIGURE 1 – Comparaison visuelle de résultats des algorithmes

basse température) donc il faudrait attendre un temps très long pour voir cette convergence. On se place donc à haute température.

Pour l'algorithme exact (couplage par le passé), on simule de manière indépendante un échantillon $(X^{(n)})_n$ et on vérifie que la loi des grands nombres s'applique :

$$\frac{1}{n} \sum_i M(X^{(i)}) \xrightarrow{n \rightarrow +\infty} \mathbb{E}(M(X)) = 0.$$

On obtient les résultats de la figure 2.

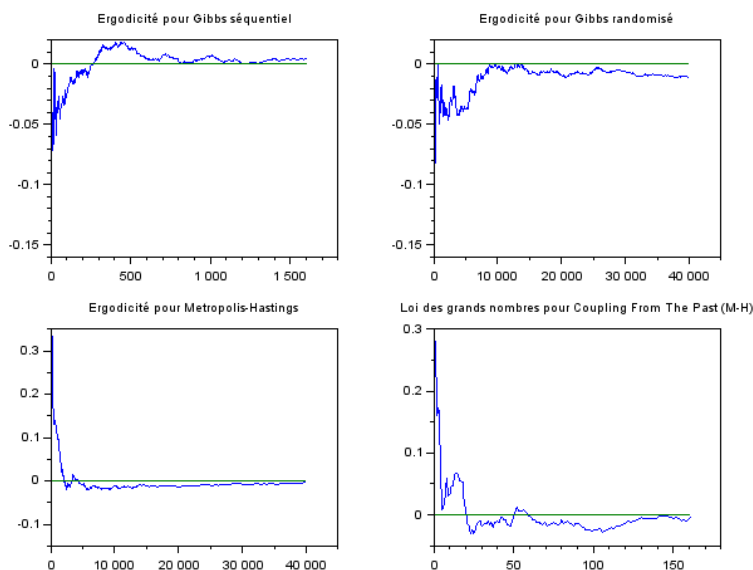


FIGURE 2 – Vérification de l'ergodicité et de la loi des grands nombres sur la magnétisation

7 Cas en petite taille : simulation naïve et température critique

En petite taille, on peut faire une simulation naïve. Pour cela, on numérote les 2^{N^2} états (dans l'ordre lexicographique en lisant les matrices colonne par colonne), on calcule exactement la loi et on choisit un numéro en utilisant cette loi et une loi uniforme sur $[0, 1]$.

On a donc une représentation (dans un vecteur \mathbf{p} dans le code SciLab) de toutes les probabilités de tous les états, et des fonctions (`etat2num` et `num2etat`) qui associent une matrice $x \in \{\pm 1\}^C$ à sa numérotation $m \in \llbracket 1, 2^{N^2} \rrbracket$ ou une numérotation à sa matrice.

Le calcul des probabilités exactes permet de tracer, en fonction de T , la probabilité d'un état donné. La figure 3 trace la probabilité de l'état

$(1, \dots, 1)$ (qui est celle de $(-1, \dots, -1)$). On observe une décroissance assez

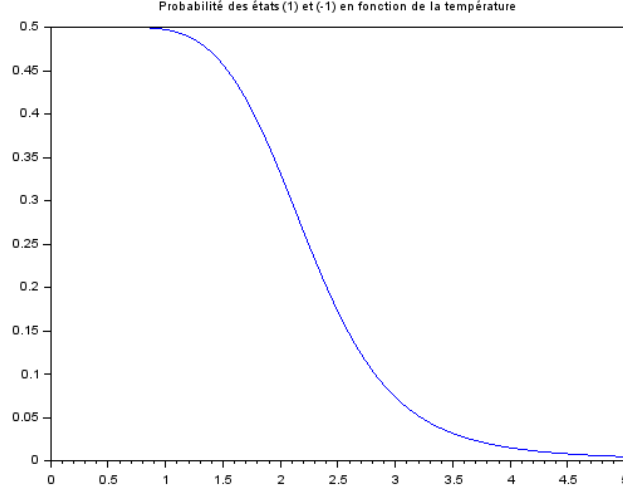


FIGURE 3 – $\pi(1, \dots, 1)$ en fonction de T pour $N = 4$

lente autour d'une température critique inférieure à la température critique théorique $T_c \simeq 2.269$. La théorie dit que quand $N \rightarrow +\infty$, cette décroissance devient de plus en plus forte et de plus en plus concentrée autour de T_c . En effet, les spins différents ont plus de chance de coexister en haute température, les états avec un spin majoritaire sont donc moins probables.

8 Observation de la transition de phase selon la taille du réseau

Comme dit en section 7, en très petite taille, la température critique est en dessous de la température critique théorique. On s'attend donc qu'avec l'augmentation de la taille du réseau, la température critique augmente en tendant vers la température critique théorique.

Un inconvénient pour observer cela est qu'en petite taille, il peut être difficile de distinguer une température sous-critique et une sur-critique, sauf quand elles sont loin de la température critique, en témoigne la figure 4. On y voit une différence claire de comportement entre $T = 2.1$ (où les spins sont peu mélangés) et les températures supérieures, mais les motifs observés sont trop petits pour distinguer des différences entre les autres configurations. La température critique effective semble alors se situer entre 2.1 et 2.2.

En augmentant la taille du réseau (figure 5), on distingue mieux les comportements différents.

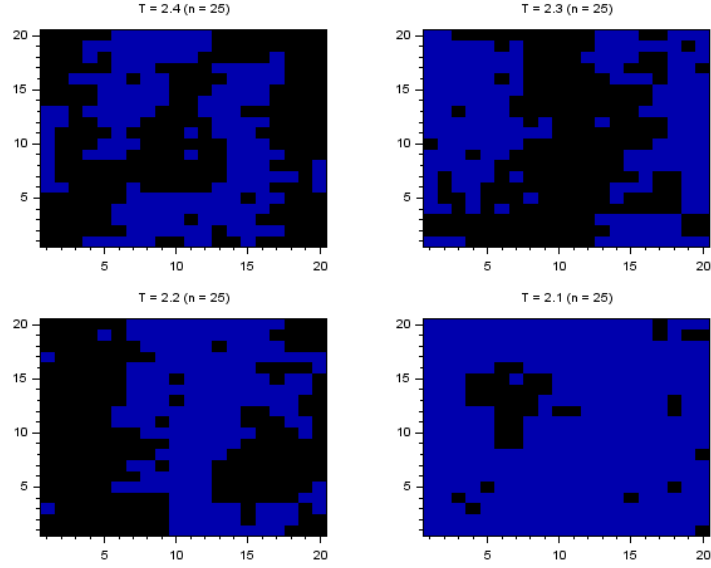


FIGURE 4 – Simulation par échantillonneur de Gibbs ($n = 25$ balayages séquentiels), $N = 20$

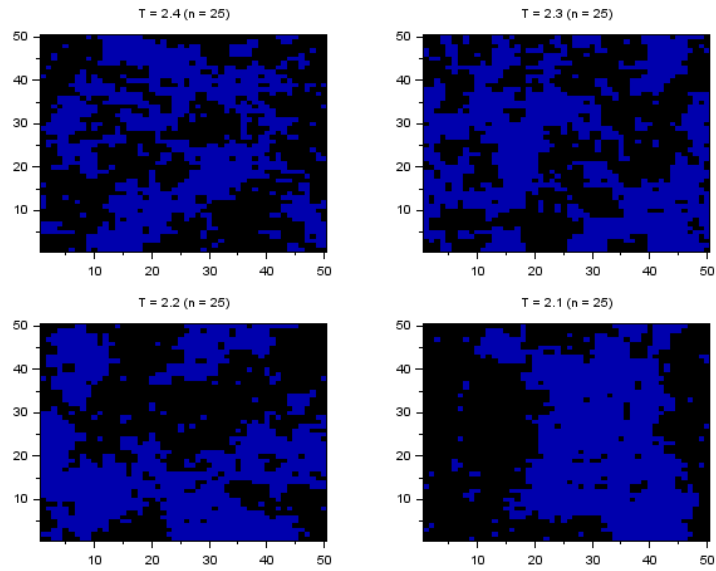


FIGURE 5 – Simulation par échantillonneur de Gibbs ($n = 25$ balayages séquentiels), $N = 50$

- À $T = 2.1$, on distingue très clairement deux blocs de spins (les conditions aux bords étant périodiques, le bloc de droite composé de -1 (noir) est en contact avec le bord noir de gauche, tout comme les côtés noir haut et bas coïncident). La température est donc trop basse pour que les spins différents se mélangent, au point où les états probablent rassemblent les spins identiques en un unique bloc.
- À $T = 2.2$, il n'y a toujours que deux blocs mais le comportement est plus proche de celui observé en figure 4 : ces blocs ne sont pas aussi nettement délimités qu'à $T = 2.1$ et sont intriqués l'un dans l'autre.
- À $T \geq 2.3$, il commence à y avoir un mélange des différents spins, même si localement les spins identiques sont rassemblés en blocs. On observe ainsi de nombreux blocs, ainsi que des spins isolés, et il y en a d'autant plus que T augmente.

Ces observations poussent à placer la température critique entre 2.2 et 2.3 mais un autre comportement est possible (figure 6). Ici, les comportements

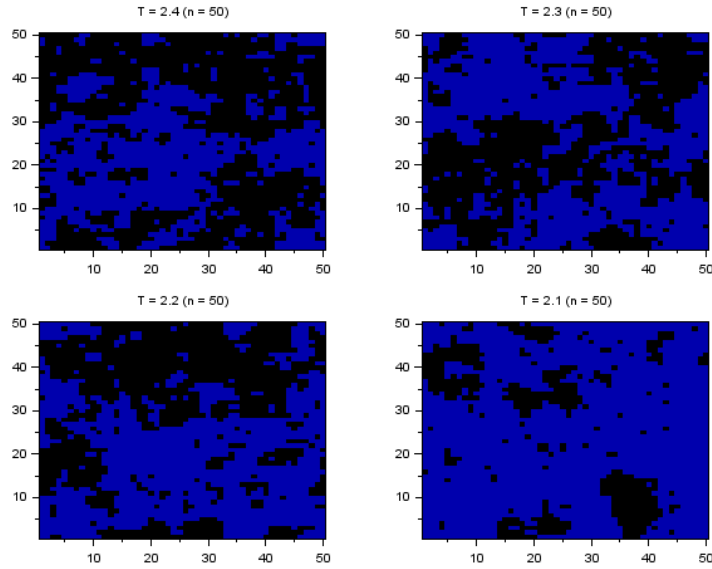


FIGURE 6 – Simulation par échantillonneur de Gibbs ($n = 50$ balayages séquentiels), $N = 50$

à $T \geq 2.2$ sont assez similaires à précédemment (des blocs de spins et des spins isolés, avec des blocs d'autant plus petit et des spins isolés d'autant plus nombreux que la température est haute), mais la différence entre $T = 2.3$ et $T = 2.2$ n'est plus aussi nette. Au contraire, à $T = 2.1$ on observe un état avec très peu de spins -1 , rassemblés en petit blocs plongés dans une majorité de spin $+1$. C'est un comportement attendu sous la température critique (une majorité d'un spin donné), on pourrait donc placer la température critique entre 2.1 et 2.2.

En augmentant encore la taille du réseau, la transition de phase est beaucoup plus évidente, même en temps court (figure 7). On peut ici séparer les

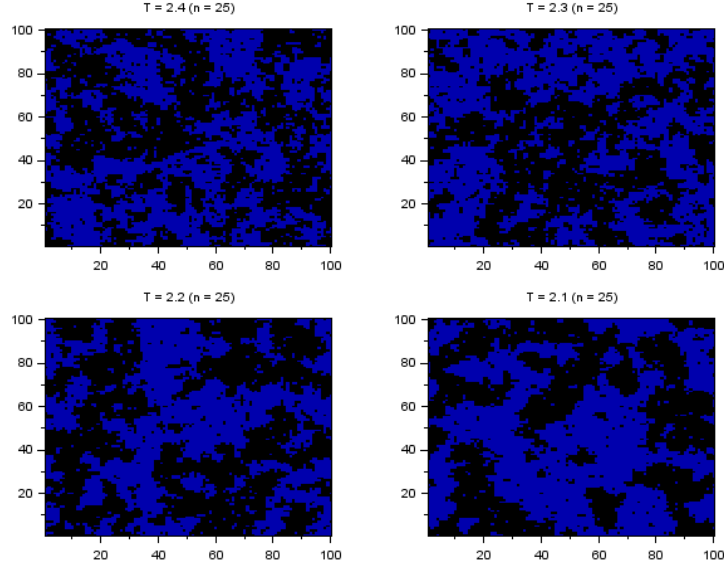


FIGURE 7 – Simulation par échantillonneur de Gibbs ($n = 25$ balayages séquentiels), $N = 100$

quatre figures en deux groupes assez distincts :

- un groupe $T \leq 2.2$, où des blocs bien délimités sont formés. La majorité des blocs formés sont de grande taille, et les spins isolés ou les blocs de petite taille sont peu nombreux et loin des frontières des blocs macroscopiques, participant à définir plus nettement les délimitations. Les blocs macroscopiques sont aussi connectés entre eux (par des "couloirs" ou des blocs microscopiques).
- un groupe $T \geq 2.3$, où les blocs n'ont pas de frontières précises, et chaque bloc d'un spin donné contient de nombreux spins opposés isolés. Il y a aussi beaucoup de blocs de petites tailles qui "contaminent" les blocs de grande taille, causant ce flou sur les frontières (et donc sur la définition) des blocs macroscopiques. Les blocs macroscopiques étant plus petits et plus flous que précédemment, ils ne sont pas ou peu connectés entre eux.

Le comportement est identique (et d'autant plus marqué) quand on laisse l'algorithme tourner plus longtemps (figures 8 et 9). Les figures 7, 8 et 9 donnent donc une température critique entre 2.2 et 2.3, ce qui correspond bien à la température critique théorique ($\simeq 2.27$).

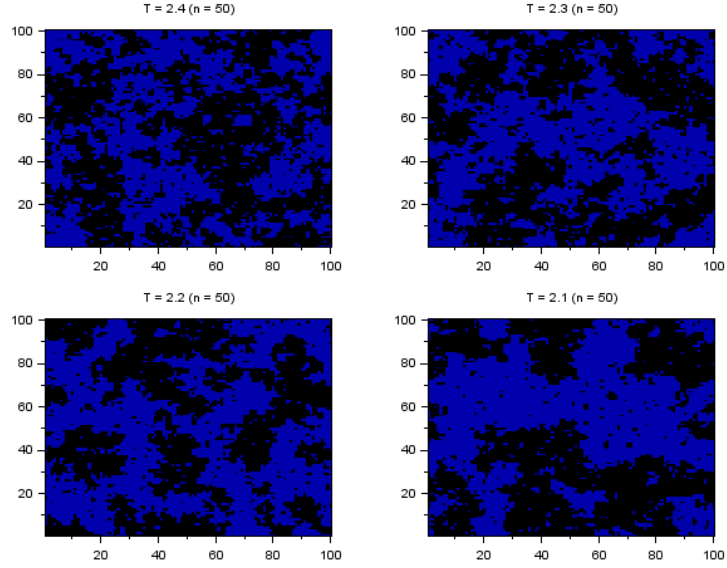


FIGURE 8 – Simulation par échantillonneur de Gibbs ($n = 50$ balayages séquentiels), $N = 100$

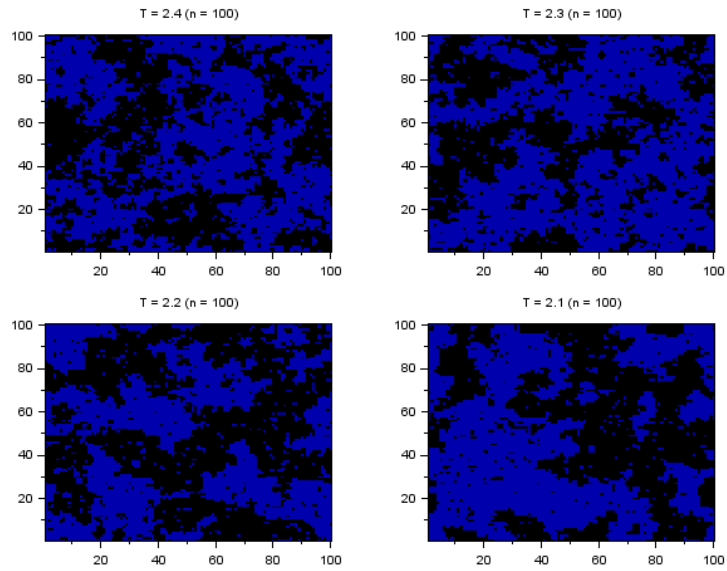


FIGURE 9 – Simulation par Metropolis-Hastings ($nN^2 = 500000$ actualisations), $N = 100$

9 Conclusion

À $J = 1$ et $h = 0$, on dispose d'une méthode exacte (couplage par le passé par Metropolis-Hastings ou par échantillonneur de Gibbs) mais qui peut être très longue, notamment pour les températures sous-critiques et pour les réseaux de grande taille ; la durée d'exécution de cet algorithme n'est contrôlée qu'en espérance, ce qui empêche de les utiliser efficacement en pratique. En revanche, les algorithmes approchés (Metropolis-Hastings et échantillonneur de Gibbs, le premier étant plus rapide que le second) permettent d'observer la transition de phase et l'influence de la taille du réseau sur celle-ci : les expériences ont bien confirmé que la température critique s'observait d'autant mieux que le réseau était grand, et qu'elle tendait de manière croissante vers la température critique théorique $T_c \simeq 2.269$.

Références

- [1] James Propp and David Wilson. Coupling from the past : a user's guide. *Microsurveys in discrete probability*, 41 :181–192, 1998.
- [2] James Gary Propp and David Bruce Wilson. Exact sampling with coupled markov chains and applications to statistical mechanics. *Random structures and Algorithms*, 9(1-2) :223–252, 1996.