

Modèle d'Ising : Magnétisme et Algorithme de Monte Carlo

Ce projet numérique a été élaboré en urgence dans le cadre de la continuité pédagogique liée au Covid-19. Des coquilles peuvent donc se glisser dans l'énoncé.

Table des matières

1	Introduction	2
2	Concepts théoriques	2
2.1	Magnétisme localisé et Hamiltonien de Heisenberg	2
2.2	Modèle d'Ising bidimensionnel	3
3	Modélisation du réseau de spins	3
3.1	Principe	3
3.2	Implémentation	4
4	Méthode de Monte-Carlo	5
4.1	Algorithme de Metropolis-Hastings	5
4.2	Implémentation	5
4.3	Évolution qualitative	6
4.4	Observables : aimantation moyenne $\langle M \rangle_M$ et énergie moyenne $\langle E \rangle_M$ de la matrice de spin	7
4.5	Vitesse de convergence	8
5	Application 1 : Évolution du système en fonction de la température	8
5.1	Implémentation	8
5.2	Observables : C_v , χ , U_4 et $G(\vec{r})$	10
5.3	Analyse de taille finie, loi d'échelle et exposants critiques	11
5.4	Mesure de la température critique	12
5.5	Détermination des exposants critiques	12
6	Application 2 : Évolution du système en fonction du champ magnétique	13
6.1	Implémentation	13
6.2	Fonction de Brillouin	13
6.3	Étude du phénomène d'hystérésis : $\langle M \rangle(H)$	14
7	Perspectives	14

1 Introduction

Dans le cadre de ce projet numérique nous proposons d'étudier le magnétisme à partir du modèle d'Ising classique [1, 2, 3] appliqué à un réseau bidimensionnel infini de spins [4, 5, 6]. Pour ce faire, vous trouverez en partie 2 les concepts physiques nécessaires pour réaliser ce projet avec l'introduction du modèle d'Ising. Les parties 3 et 4, plus techniques, abordent l'implémentation du calcul numérique à partir de l'algorithme de Monte-Carlo que vous devrez développer sous Python. Enfin, le cœur de ce projet numérique va consister à étudier le comportement du modèle d'Ising lors de la variation de deux paramètres que sont la température (partie 5) et le champ magnétique (partie 6). Enfin, si le temps le permet, vous pourrez adapter votre programme et étendre cette étude à un ou plusieurs problèmes plus complexes de votre choix dont quelques pistes sont évoquées dans la partie 7.

2 Concepts théoriques

2.1 Magnétisme localisé et Hamiltonien de Heisenberg

Une très grande variété de problème de magnétisme classique peuvent être traités en ne considérant que l'interaction dipolaire entre deux moments magnétiques $\vec{\mu}_1$ et $\vec{\mu}_2$, séparés d'une distance r , dont l'énergie correspondante s'écrit :

$$E = \frac{\vec{\mu}_1 \cdot \vec{\mu}_2}{r^3} - 3 \frac{(\vec{\mu}_1 \cdot \vec{r})(\vec{\mu}_2 \cdot \vec{r})}{r^5} \quad (1)$$

Ce n'est pas le cas de tous les problèmes physiques liés au magnétisme *de la matière*, en particulier ceux qui cherchent à comprendre le magnétisme créé par les électrons dans des solides isolants (magnétisme localisé), comme celui des aimants permanents ferromagnétiques. Dans ce cas, l'interaction entre les moments magnétiques de spin des électrons est purement d'origine quantique : il s'agit de l'interaction d'échange [3]. L'énergie d'interaction, ou Hamiltonien de Heisenberg, entre ces spins s'écrit :

$$E = -J \sum_{\langle i,j \rangle} \vec{S}_i \vec{S}_j \quad (2)$$

où J est la constante de couplage, \vec{S}_i et \vec{S}_j les opérateurs de spins du site noté i et j et la somme court sur les paires de premiers voisins uniquement, comptées une seule fois (i.e. si la paire $\langle i, j \rangle$ est prise en compte alors $\langle j, i \rangle$ ne l'est pas). Si $J > 0$, alors les spins auront intérêt à s'aligner (interaction ferromagnétique) alors que si $J < 0$ l'anti-alignement entre plus proches voisins sera favorisé (interaction antiferromagnétique). En présence d'un champ magnétique extérieur \vec{H} appliqué, l'énergie totale devient :

$$E = -J \sum_{\langle i,j \rangle} \vec{S}_i \vec{S}_j - g\mu_B \vec{H} \sum_i \vec{S}_i \quad (3)$$

avec μ_B le magnéton de Bohr et g le facteur de Landé de l'électron.

2.2 Modèle d'Ising bidimensionnel

La résolution du Hamiltonien de Heisenberg, pourtant relativement simple, s'avère pourtant extrêmement difficile dans des cas réels, du fait du nombre très important de paramètres pour les matériaux macroscopiques ($N_A = 6,022 \cdot 10^{23}$ spins à traiter !) ainsi que de la non-commutativité des opérateurs de spins \vec{S}_i et \vec{S}_j . Pour simplifier encore le problème, nous allons donc considérer uniquement la projection de ces spins suivant une direction et considérer un système bidimensionnel avec un nombre raisonnable de spins $N \sim 400$ pour les calculs numériques. L'énergie s'écrit alors :

$$E = -J \sum_{\langle i,j \rangle} S_i S_j - g\mu_B H \sum_i S_i \quad (4)$$

Il s'agit de l'énergie définissant le modèle d'Ising. Le but de ce modèle sera d'étudier, a minima, le comportement d'un tel système en fonction de différents paramètres que sont la *température*, la *valeur du couplage* J (en particulier son signe) et le *champ magnétique extérieur*. Dans la suite, on posera $H = 0$ pour simplifier : l'étude de l'effet du champ sera étudié spécifiquement dans la partie 6.

Bien que le modèle d'Ising ait été initialement introduit par Ernst Ising en 1925 pour mieux comprendre la transition paramagnétique-ferromagnétique dans les matériaux aimantés, il s'avère en fait être très général et permet d'étudier des problèmes en physique statistique, économie, écologie...

3 Modélisation du réseau de spins

3.1 Principe

Afin de modéliser un réseau de spins nous pouvons utiliser une matrice dont les éléments $M(i, j)$ seront les valeurs de spins $+1$ ou -1 . Au lieu de considérer un 'vrai' réseau infini, nous allons travailler sur un réseau périodique carré $N \times N$, tel que $S(i, j) = S(i + n_1 * N, j + n_2 * N)$, $\forall (n_1, n_2) \in \mathbb{Z}$. Ce tableau carré rempli de spins peut être visualisé comme une matrice coloré en bleu (spin $+1$) et en rouge (spin -1) comme sur la figure 1.

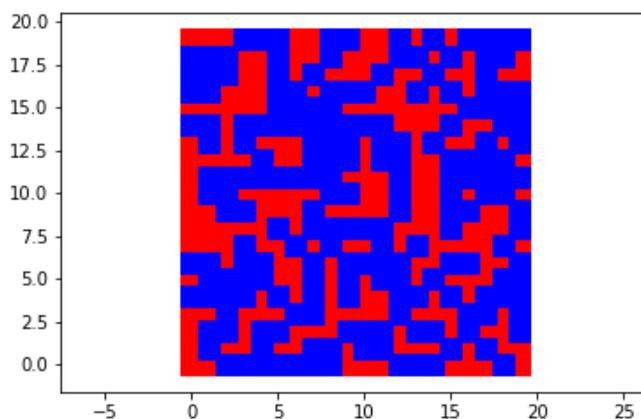


FIGURE 1 – Représentation matricielle d'un réseau (20×20) de spins $+1$ (bleu) et -1 (rouge).

3.2 Implémentation

Voici un programme Python simple permettant de créer, d'initialiser et d'afficher la matrice sus-cité :

```

1 from numpy import *
2 from pylab import *
3 from scipy.signal import medfilt, argrelemin, argrelmax
4 from scipy.optimize import leastsq, curve_fit
5 from scipy.constants import constants as c
6 import random
7 import math
8 import numpy as np
9 import time
10 import matplotlib.pyplot as plt
11
12
13 # Set Matrix spin size
14 tailleX=20
15 tailleY=20
16
17 #Initialisation Matrix filled with -1
18 myarray=np.ones((tailleX,tailleY))*(-1)
19
20 #Initialisation Matrix filled with random spin +/-1
21 for i in range(tailleX):
22     for j in range (tailleY):
23         a = (random.randint(0.,1.))*2-1
24         myarray[i,j] = a
25
26 # Matrix plot
27 x_neg = []
28 y_neg = []
29 x_pos = []
30 y_pos = []
31 for i in range(tailleX):
32     for j in range (tailleY):
33         if myarray[i,j] < 0:
34             x_neg.append(i)
35             y_neg.append(j)
36         else:
37             x_pos.append(i)
38             y_pos.append(j)
39 scatter(x_neg,y_neg,c= 'red', marker = 's', s=100) # spin -1 in red
40 scatter(x_pos,y_pos,c= 'blue', marker = 's', s=100) # spin +1 in blue
41 axis('equal')
42 show()

```

Listing 1 – Tracé de la matrice

Le résultat attendu d'un tel programme (listing 1) est la matrice de spin présentée en figure 1. Afin de prendre en main les notations et Python, veuillez :

- Recopier ce code dans un editeur de votre choix (Notepad++ est très pratique), sauver le sous la forme d'un fichier Python, le compiler et transformer ce code d'affichage en fonction (`def(...)`).
- Faire varier les dimensions de la matrice ainsi que la taille des marqueurs pour vous sensibiliser aux limites de la représentation graphique.
- Retourner manuellement un spin de la matrice $M(i, j)$ et visualiser ce changement.
- Rajouter une ligne de code permettant d'enregistrer sous la forme d'un fichier image la matrice aléatoire générée.

4 Méthode de Monte-Carlo

4.1 Algorithme de Metropolis-Hastings

L'algorithme de Monte-Carlo est très général [7] : il peut s'appliquer à tout modèle pour lequel il est possible (i) de faire des modifications aléatoires du modèle et (ii) d'associer une variation d'énergie à chacune de ces modifications. L'algorithme procède alors de la manière suivante :

- soit une modification aléatoire qui transforme le modèle M_i en modèle M_{i+1} , associée à une variation d'énergie ΔE .
- Si $\Delta E < 0$, la modification est acceptée.
- Si $\Delta E > 0$, la modification est acceptée avec la probabilité $P = e^{-\Delta E/k_B T}$, où T est la température de l'algorithme et k_B la constante de Boltzmann (on prendra $k_B = 1$ pour simplifier).

Après avoir répété de nombreuses fois cet algorithme, un état stationnaire est atteint et la distribution des états générés M_i correspond à une distribution de Boltzmann.

4.2 Implémentation

A partir de la matrice aléatoire générée plus haut (par défaut, on étudiera une matrice de taille 20×20), l'algorithme Monte-Carlo va consister en une grande boucle permettant de répéter un nombre important de fois (par défaut, on utilisera 50000) un test sur un spin de manière aléatoire dans le but d'atteindre un état stationnaire. Pour chaque itération de cette boucle, il faudra 1- piocher aléatoirement un spin, 2- le retourner, 3- calculer le coût en énergie du retournement et 4- valider le retournement du spin ou revenir en arrière. Pour cela est proposé ci-dessous un corps de code (listing 2) que nous vous conseillons de remplir en suivant nos indications de manière méthodique :

Identifier aléatoirement un spin de la matrice et le retourner

A l'intérieur de la boucle Monte-Carlo, écrire quelques lignes permettant d'identifier un spin $S_i = M[\alpha, \beta]$ de manière aléatoire à l'intérieur de la matrice $M[i, j]$ de taille $N \times N$. On utilisera pour cela la fonction `random.randint()` puis on réalisera un premier test permettant de savoir si le spin est positif ou négatif via la fonction `If Cond : ... else :`. Si le spin est négatif, on le retourne en le faisant devenir positif et inversement.

Calculer le coût en énergie du retournement

Dans sa nouvelle configuration, le spin choisi a vu son énergie d'échange évoluer. Calculer cette variation d'énergie à partir de l'équation 4 en prenant en compte l'interaction entre les seuls premiers voisins. Attention de bien prendre en compte la condition de périodicité de la matrice $S(i, j) = S(i + n_1 * N, j + n_2 * N)$, $\forall (n_1, n_2) \in \mathbb{Z}$.

Validation du changement et probabilité d'acceptation

En fonction de l'énergie obtenue, choisissez de garder la nouvelle configuration ou de revenir en arrière. Pour cela utiliser de nouveau la fonction `If Cond : ... else :`. Pour rappel, si la variation d'énergie est négative la transformation est acceptée, si elle est positive alors elle doit être acceptée avec

une probabilité $P = e^{-\Delta E/k_B T}$. Cette dernière probabilité pourra être prise en compte en utilisant la fonction `random.uniform(0.,1.)`.

```

1 | # Set the amplitude and sign of the spin-spin interaction together with
  |   the temperature
2 | J=+1
3 | T=0.2
4 |
5 | #####Monte Carlo#####
6 |
7 | # Number of repetition
8 | Repet=50000
9 |
10 | for r in range(Repet):
11 |     # preparation to pick up randomly a spin with coordinates (alpha
  |     ,beta) in interval[0, tailleX/Y-1]
12 |     #??????????
13 |     if ????: # if the spin S[alpha,beta]<0
14 |         # return the spin to Si=+1
15 |         #??????????
16 |         # calculate energy cost
17 |         #??????????
18 |         if ????: #if change in energy is negative
19 |             # keep the new spin Si=+1
20 |             #??????????
21 |         else: # if not (change in energy is positive)
22 |             # reverse or not the spin with probability exp(-E/T)
23 |             #??????????
24 |     else:# if the spin S[alpha,beta]>0
25 |         # return the spin to Sj=-1
26 |         #??????????
27 |         # calculate energy cost
28 |         #??????????
29 |         if ????: #if change in energy is negative
30 |             # keep the new spin Sj=-1
31 |             #??????????
32 |         else: # if not (change in energy is positive)
33 |             # reverse or not the spin with probability exp(-E/T)
34 |             #??????????

```

Listing 2 – Algorithme de Metropolis-Hastings

4.3 Évolution qualitative

Afin de vérifier si votre code fonctionne, nous allons étudier le comportement de la simulation en fonction du nombre de répétitions. Pour cela il sera intéressant d’afficher la matrice au fur et à mesure que le cycle Monte-Carlo se répète. Afin d’éviter une divergence dans le temps de calcul, veillez à n’afficher que périodiquement cette matrice. Par exemple, si vous choisissez un nombre de répétition de 50000, un affichage toutes les 1000 répétitions peut suffire. Vous pouvez pour cela vous guider du code 3 ci-dessous.

Étudiez alors le comportement de la matrice dans les situations suivantes :

- A basse température ($T = 0.5$ par exemple) pour les deux situations $J = 1$ et $J = -1$. À quoi correspondent ces deux situations et qu’attendez-vous ? Est-ce ce que vous observez ? À chaque fois ?

- Quels sont les différents régimes observés en fonction de la température ?
- Pour $J = 1$, faire évoluer le système à température fixe $T = 0.5$ (basse température) pour une “grande” dimension (50×50 ou 100×100) de la matrice. Observer la formation de domaines magnétiques. Proposer une explication qualitative de la (méta)stabilité des domaines observés à partir de la forme des parois et de la taille des domaines.
- Rajouter une fonctionnalité permettant d’enregistrer sous la forme d’un fichier image la matrice au cour du cycle de répétition (attention seulement quelques-unes).
- Créer une courte vidéo (pas forcément avec Python) ou réaliser un empilement des matrices permettant de visualiser l’évolution des domaines magnétiques.

4.4 Observables : aimantation moyenne $\langle M \rangle_M$ et énergie moyenne $\langle E \rangle_M$ de la matrice de spin

Dans le but d’analyser plus quantitativement vos observations, il peut être intéressant de regarder l’évolution de l’aimantation moyenne $\langle M \rangle_M$ ainsi que l’énergie moyenne par spin $\langle E \rangle_M$ au cours des répétitions. Ces quantités sont définies comme :

$$\langle M \rangle_M = \frac{1}{N_T} \sum_{i=1}^{N_T} S_i \quad (5)$$

$$\langle E \rangle_M = \frac{1}{2N_T} \left(-J \sum_{i=1}^{N_T} \sum_{j,pv} S_i S_j \right). \quad (6)$$

Ici, $N_T = N \times N$ est le nombre total de spins. Pour l’énergie moyenne, la somme court sur les premiers voisins et le facteur 2 prend en compte le fait que dans la somme, chaque spin est compté deux fois.

Recopier dans votre programme le code ci-dessous (listing 3) auquel vous ajouterez une fonctionnalité permettant de calculer et tracer en fonction du nombre de répétition ces deux quantités. Attention, à ne le faire que de temps en temps pour éviter une saturation de votre machine.

```

1 |     ##Calculate and show only partially the data
2 |     if r%(Repet/50) == 0: # show a matrix and calculate related
   |         quantities every Repet/100. example: Repet=5000. repet
   |         /100=50. It will show the matrix at r=0,50,100,150...5000
3 |
4 |         ## <M>_m and <E>_m
5 |         #???????????
6 |
7 |         ## Plot of all relevant quantities
8 |
9 |         # Matrix plot preparation
10 |         x_neg = []
11 |         y_neg = []
12 |         x_pos = []
13 |         y_pos = []
14 |         for i in range(tailleX):
15 |             for j in range (tailleY):
16 |                 if myarray[i,j] < 0:
17 |                     x_neg.append(i)
18 |                     y_neg.append(j)
19 |                 else:
```

```

20         x_pos.append(i)
21         y_pos.append(j)
22
23     # Matrix plot
24     plt.figure(1)
25     plt.scatter(x_neg,y_neg,c= 'red', marker = 's', s=100)
26     plt.scatter(x_pos,y_pos,c= 'blue', marker = 's', s=100)
27     axis('equal')
28     plt.title('Spin matrix')
29     draw()
30     plt.show()
31
32     # <M>_m plot
33     #????????????
34
35     # <E>_m plot
36     #????????????
37
38     time.sleep(1.0) # slow down windowing
39 ## Saving final data in .dat file
40 #????????????

```

Listing 3 – Calcul et tracé de $\langle M \rangle_M$ et $\langle E \rangle_M$ de manière partielle

4.5 Vitesse de convergence

Atteindre un état stationnaire requiert un nombre important d'essais qui varie fortement en fonction de la taille de la matrice. Évaluer quantitativement cette vitesse de convergence à partir de $\langle M \rangle_M$ et $\langle E \rangle_M$ pour des matrices de taille 2×2 , 4×4 , 8×8 , 16×16 , 20×20 et 50×50 en prenant $T = 0.5$. Pensez bien à vérifier la corrélation entre $\langle M \rangle_M$, $\langle E \rangle_M$ et la matrice de spin et ne pas confondre un état stationnaire avec un état pseudo-stationnaire (domaines magnétiques d'une géométrie particulière).

5 Application 1 : Évolution du système en fonction de la température

L'évolution en température permet de mettre en avant le phénomène de transition de phase. Pour une interaction positive $J = 1$, une transition de phase apparaît à la température de Curie :

$$T_C = \frac{2J}{k_B \ln(1 + \sqrt{2})}. \quad (7)$$

En dessous de cette température, le matériau est ferromagnétique, en dessus il est paramagnétique. Pour une interaction $J = -1$ négative, cette transition apparaît à la température de Néel T_N , température au-dessous de laquelle le matériau est antiferromagnétique.

5.1 Implémentation

Ajouter dans le programme principal une boucle en fonction de la température (voir listing 4 ci-dessous), en laissant le système évoluer suffisamment longtemps à chaque température, et stocker dans des tableaux 1D les valeurs moyennes thermiques de l'aimantation $\langle M \rangle$, $\langle E \rangle$ et de l'énergie pour $J = 1$.

```

1 ##### Loop on temperature#####
2 #Tmin=???
3 #Tmax=???
4 #pas=?
5 for T in np.arange(Tmin,Tmax,pas) :
6     ## Create temperature list
7     #???????????
8     #MonteCarlo repetitions
9     Repet=50000
10    Equi=int(0.8*Repet) # equilibration time
11    #Prepare a list of M and E for each repetition after \
        Equilibration
12    #???????????
13    for r in range(Repet):
14        ## MonteCarlo
15        #???????????
16        #???????????
17        ## After equilibration steps "Equi"
18        listrfin=[]
19        sweep=0
20        if r>Equi:
21            #Calculate average spin (magnetization) of a \
                single matrix and append it to a list
22            #???????????
23            #???????????
24
25            #Calculate spatial average energy of a single \
                matrix and append it to a list
26            #???????????
27            #???????????
28
29            ## Calculate mean thermal energies <E>, <E^2> and specific heat Cv
30            ## Add to a list for temperature dependence
31            #???????????
32            #???????????
33
34            ## Calculate mean thermal magnetization <M>, <M^2>, susceptibility X\
                , Binder cumulant U_4
35            ## Add to a list for temperature dependence
36            #???????????
37            #???????????
38
39            ## Calculate Correlations G(r)
40            ## Add to an array for temperature dependence
41            #???????????
42            #???????????
43
44            T=T+pas # increment temperature
45
46            ## Matrix plot
47            ## <M>,<M^2>,X,U_4 plots
48            #???????????
49            ## <E>,<E^2>, Cv plots
50            #???????????
51            time.sleep(1.0) # slow down windowing
52 ## Saving final data in .dat file
53 #???????????

```

Listing 4 – Boucle en température et observables

Attention : On utilisera à partir de maintenant les moyennes thermiques $\langle M \rangle$, $\langle E \rangle$ à ne pas confondre avec les moyennes sur une matrice unique $\langle M \rangle_M$ et $\langle E \rangle_M$ utilisées précédemment. En pratique, on définit :

$$\langle M \rangle = \frac{1}{N_r} \sum_{r=1}^{N_r} \langle M \rangle_{M,r} = \frac{1}{N_r} \sum_{r=1}^{N_r} \left(\frac{1}{N_T} \sum_{i=1}^{N_T} S_i \right)_r \quad (8)$$

$$\langle E \rangle = \frac{1}{N_r} \sum_{r=1}^{N_r} \langle E \rangle_{M,r} = \frac{1}{N_r} \sum_{r=1}^{N_r} \left(\frac{1}{2N_T} \left(-J \sum_{i=1}^{N_T} \sum_{j,pv} S_i S_j \right) \right)_r . \quad (9)$$

Cette moyenne thermique correspond à une moyenne de la quantité considérée sur le cycle de répétition r de la simulation Monte-Carlo de longueur N_r . Cette moyenne n'est thermique que si l'on moyenne sur un état stationnaire, à l'équilibre thermique. En pratique on ne calculera cette moyenne thermique que vers la fin de la boucle Monte Carlo, le temps que l'équilibrage se fasse, par exemple les vingt derniers pourcent restants de la boucle.

À chaque température, afficher la matrice finale ainsi que les graphiques aimantation moyenne et énergie. Interpréter vos résultats et extraire une première valeur de la température de Curie. Cette température correspond-elle à la relation 7 ?

5.2 Observables : C_v , χ , U_4 et $G(\vec{r})$

Proche de la transition de phase, on s'attend à une recrudescence des fluctuations thermiques des orientations de spin dues à la compétition entre agitation thermique et énergie d'échange. Ces fluctuations trouvent leurs représentations dans la chaleur spécifique, la susceptibilité du paramètre d'ordre, le cumulants de Binder du paramètre d'ordre ainsi que dans la longueur de corrélation. Les trois premières quantités sont définies par les relations :

$$C_v(T) = \frac{\partial E}{\partial T} = \frac{\langle E \rangle^2 - \langle E^2 \rangle}{N_T k_B T^2} \quad (10)$$

$$\chi(T) = \frac{\partial M}{\partial H} = \frac{N_T (\langle M \rangle^2 - \langle M^2 \rangle)}{k_B T} \quad (11)$$

$$U_4 = 1 - \frac{\langle M^4 \rangle}{3 \langle M^2 \rangle} \quad (12)$$

où on définit la valeur moyenne thermique de la quantité Q^α par : $\langle Q^\alpha \rangle = \frac{1}{N_r} \sum_{r=1}^{N_r} \left(\langle Q \rangle_{M,r} \right)^\alpha$.

Les deux premières fonctions sont des fonctions piquées au voisinage de la transition. Le cumulants de Binder sera utile pour déterminer la température de Curie de manière précise. La fonction de corrélation de spin peut elle s'écrire comme :

$$G(\vec{r}) = G(\vec{r}_i - \vec{r}_j) = \langle S_i S_j \rangle - \langle S_i \rangle^2 \quad (13)$$

et correspond à la valeur moyenne renormalisée du produit $S_i S_j$ pour des spins i et j situés à une distance \vec{r} . À longue distance, $G(\vec{r})$ a une forme exponentielle décroissante :

$$G(\vec{r}) \approx e^{-\frac{r}{\xi}} \quad (14)$$

où ξ est la longueur de corrélation du système, qui peut donc s'écrire pour r grand :

$$\xi \approx \frac{r}{\langle S_i S_j \rangle} \quad (15)$$

où r est la distance séparant les deux spins S_i et S_j . Rappelons que cette analyse n'a de sens qu'au voisinage de la température critique car sinon l'exponentielle décroît trop vite ($T \ll T_C$), ou bien le système est désordonné et les spins ne sont alors plus corrélés ($G(\vec{r}) \approx 0$ pour $T \gg T_C$).

Rajouter dans votre programme (se référer au listing 4) les fonctionnalités nécessaires permettant de tracer C_v , χ , U_4 et $G(\vec{r})$. Pour la longueur de corrélation, on choisira un petit nombre (10% du nombre total) de spins au hasard et pour chacun on calculera leur corrélation avec les spins pris à une distance maximale égale à la moitié de la taille de la matrice dans la direction des x , des y et/ou dans une direction aléatoire.

Essayer alors d'extraire la longueur de corrélation au voisinage de la transition. Pour cela il vous faudra tracer la fonction $\ln(G) = f(r)$ pour déterminer la pente $-1/\xi$ que vous pourrez ainsi tracer en fonction de la température. Qu'observez-vous ?

5.3 Analyse de taille finie, loi d'échelle et exposants critiques

La présence d'une transition de phase se manifeste par une divergence de la susceptibilité du paramètre d'ordre (ici l'aimantation) et de la chaleur spécifique à la température critique. Cette divergence est alors gouvernée par des lois d'échelles comportants des exposants critiques. Ces lois d'échelles sont, de manière générale pour des systèmes de grande taille et une transition du second ordre :

$$M = M_0 t^\beta \quad (16)$$

$$\chi = \chi_0 t^{-\gamma} \quad (17)$$

$$C_v = C_0 t^{-\alpha} \quad (18)$$

$$\xi = \xi_0 t^{-\nu} \quad (19)$$

$$G(\vec{r}) = \eta_0 \frac{e^{-r/\xi}}{r^{d-2+\eta}} \quad (20)$$

où $t = \frac{|T-T_C|}{T_C}$, d est la dimension du système et β , γ , α , η et ν sont les exposants critiques. En étudiant précisément la dépendance en température de quantités tels que la susceptibilité et la chaleur spécifique, il est possible de déterminer les exposants critiques associées à ces lois d'échelles. Ces exposants critiques, pour le modèle d'Ising carré à deux dimensions ont été découvertes par Ralf Onsager en 1944 [8] : $\beta = 0.25$, $\gamma = 1.5$, $\eta = 0.5$, $\nu = 1$ et $\alpha = 0$. Notons qu'ici le coefficient α est nul car la chaleur spécifique ne suit pas une loi de puissance mais une loi logarithmique. Par convention, et pour coller à la généralité de ces exposants critiques, on utilise $\alpha = 0$.

Il est souvent suffisant de ne mesurer que deux exposants critiques pour connaître tous les autres. Cela est possible grâce aux relations d'échelle suivantes :

$$\alpha + 2\beta + \gamma = 2 \quad (21)$$

$$d\nu = 2 - \alpha \quad (22)$$

$$\gamma = \nu(2 - \eta) \quad (23)$$

En revanche, la transition de phase n'a vraiment lieu que dans la limite thermodynamique. Pour un système de taille finie, la susceptibilité et la chaleur spécifique ne divergent pas mais présentent un pic.

Il est toutefois possible de tenir compte de ces effets de taille finie et d'extraire les exposants critiques à partir du maximum du pic à la température critique. En effet, à cette température on obtient une constante qui dépend de la taille de la matrice L :

$$M(L, T) = L^{-\beta/\nu} M_0(tL^{1/\nu}) \xrightarrow{T=T_C} M(L) \propto L^{-\beta/\nu} \quad (24)$$

$$\chi(L, T) = L^{\gamma/\nu} \chi_0(tL^{1/\nu}) \xrightarrow{T=T_C} \chi(L) \propto L^{\gamma/\nu} \quad (25)$$

$$C_v(L, T) = L^{-\alpha/\nu} C_0(tL^{1/\nu}) \xrightarrow{T=T_C} C_v(L) \propto L^{\alpha/\nu} \quad (26)$$

Grâce à ces expressions on peut remonter aux différents rapports β/ν , γ/ν et α/ν à partir de M , χ et C_v . On tracera pour ce faire l'amplitude maximum des différents pics en fonction de L (sauf pour la chaleur spécifique car dans le cas particulier du réseau carré $2D$, $C_v(L) \propto \ln(L)$ et $\alpha = 0$).

5.4 Mesure de la température critique

Nous allons ici déterminer la température critique "intrinsèque" grâce au tracé, en fonction de la température, du cumulante de Binder du paramètre d'ordre défini plus haut pour différentes tailles de réseau ($L=8,10,16,20$ par exemple). En théorie, ces courbes doivent toutes se couper en un même point correspondant à la température critique (de Curie ici) que l'on notera $T_C(\infty)$. En pratique, vous devriez obtenir plusieurs points d'intersections qui vous permettront d'estimer la précision de cette mesure. Cette mesure de température critique intrinsèque est à opposer à la température critique que l'on obtient en notant la position du pic de susceptibilité $T_C(L)$ qui reflète pour sa part la dimension finie du réseau.

5.5 Détermination des exposants critiques

Détermination de ν

Il existe une relation simple permettant de relier la taille du réseau, la température critique intrinsèque et la position du pic de susceptibilité :

$$T_C(L) = T_C(\infty) + KL^{-1/\nu} \quad (27)$$

où K est une constante. En traçant $\ln(T_C(L) - T_C(\infty))$ en fonction de $\ln(L)$, on obtient une droite de pente $-1/\nu$ qui permet de remonter à un premier exposant critique ν .

Détermination de γ

D'après la relation 25, on a à la température critique $T_C(\infty)$, $\chi(L)L^{-\gamma/\nu} = cte$. On peut alors écrire une relation entre les différentes valeurs de la susceptibilité mesurées à la température critique pour différentes tailles de matrice L ou L' :

$$\frac{\ln(\chi(L)/\chi(L'))}{\ln(L/L')} = \frac{\gamma}{\nu}. \quad (28)$$

Cette équation permet de remonter au rapport $\frac{\gamma}{\nu}$ et donc γ .

Notons qu'un raisonnement strictement identique sur l'aimantation permet de mesurer le rapport $\frac{\beta}{\nu}$.

Déduction des autres exposants critiques α , β et η

Il est immédiat de déduire les exposants critiques manquants à partir des équations 21, 22 et 23. Comparez alors vos exposants critiques avec ceux connus théoriquement.

Cas particulier de la chaleur spécifique C_v

Dans le cadre du modèle d'Ising 2D sur réseau carré, le pic de chaleur spécifique est censé vérifier une loi du type :

$$C_v(L) \propto \ln(L). \quad (29)$$

Vérifiez cette proposition.

6 Application 2 : Évolution du système en fonction du champ magnétique

En présence d'un champ magnétique H vertical (parallèle à la direction des spins), on doit considérer le terme énergétique lié au champ magnétique (idem équation 4) :

$$E = -J \sum_{\langle i,j \rangle} S_i S_j - g\mu_B H \sum_i S_i \quad (30)$$

dans lequel μ_B est le magnéton de Bohr (on prendra $\mu_B = 1$ dans la suite) et g le facteur de Landé (on prendra $g = 1$). Ce terme force les spins à s'aligner au champ H appliqué.

6.1 Implémentation

Rajouter dans votre programme ce nouveau terme énergétique et une boucle permettant de balayer le champ magnétique de $H = -2$ à $H = +2$ et inversement pour une température donnée T et $J = 1$. Pour chaque valeur du champ magnétique, laisser évoluer le système suffisamment longtemps (50000) et après chaque variation du champ magnétique, bien enregistrer les valeurs moyennes thermiques d'aimantation et d'énergie que vous afficherez sous la forme d'un graphique du type $\langle M, E \dots \rangle (H)$ et enregistrerez.

Comparer les courbes $\langle M \rangle (H)$ pour les deux situations $T < T_C$ et $T > T_C$. Quelles différences qualitatives observez-vous ?

6.2 Fonction de Brillouin

La dépendence en champ magnétique de l'aimantation d'un système paramagnétique ($T > T_C$) suit, sous certaines conditions, la formule $\langle M \rangle = g\mu_B S B_S(x)$ où $B_S(x)$ est la fonction de Brillouin :

$$B_S(x) = \frac{2S+1}{2S} \coth\left(\frac{2S+1}{2S}x\right) - \frac{1}{2S} \coth\left(\frac{1}{2S}x\right) \quad (31)$$

avec $x = g\mu_B S B / k_B T$, $B = \mu_0 H$ et S le spin.

- Tracer $\langle M \rangle (H)$ pour différentes températures ($T > T_C$) et comparer vos courbes numériques à la fonction de Brillouin pour un spin $S = 1$.

- Pour $H \rightarrow 0$, et pour une température donnée, montrez qu’il existe un régime quasi-linéaire pour $\langle M \rangle (H)$. On peut alors définir une susceptibilité magnétique χ_m telle que $\chi_m(T) = \frac{\partial \langle M \rangle (H, T)}{\partial H}$.
- Etudiez la dépendance en température de $\chi_m(T)$ pour $T > T_C$. On prendra soin de calculer beaucoup de points lorsque T devient proche de T_C .

6.3 Étude du phénomène d’hystérésis : $\langle M \rangle (H)$

Vous allez maintenant comparer les deux sens de variation croissant ou décroissant de H .

- Êtes-vous capable d’observer un phénomène d’hystérésis ? Quelle est la condition sur la température pour pouvoir l’observer ?
- Étudier l’amplitude de l’hystérésis en fonction de la température aux environs de T_C .

7 Perspectives

Les taches sus-mentionnées correspondent aux attentes minimales du projet numérique. Une fois que votre programme fonctionne, il peut être très intéressant d’étendre son champ d’application. Nous listons ici une liste non exhaustive de simulations qui pourraient être menées en extension de ce projet :

- Etude de la présence d’impuretés magnétiques ou non-magnétiques localisées, notamment en fonction de leur taille et de leur forme.
- Changer la forme ou le type des interactions
 - Introduction d’un couplage J qui dépend de la direction $J \rightarrow J_x, J_y$
 - Introduction d’un couplage avec le second voisin $J \rightarrow J_1, J_2$
 - Introduction d’un couplage de type ”Ruderman-Kittel-Kasuya-Yoshida” ou ”RKKY”^[9]
 $J \rightarrow \frac{J}{R^4}(2kR \cos(kR) - \sin(2kR))$
- Changer le type de réseau
 - Réseau 1D
 - Réseau de type hexagonal ^[10]
 - Réseau de type Kagomé ^[11]
- ...

Références

- [1] Solid State Physics, by N. W. Ashcroft and D. N. Mermin, Holt-Sanders
- [2] Introduction to Solid State Physics, by Charles Kittel, Wiley
- [3] Magnetism in Condensed Matter, by Stephen Blundell, Oxford University Press
- [4] Baxter, Rodney J. (1982), Exactly solved models in statistical mechanics, London : Academic Press Inc. [Harcourt Brace Jovanovich Publishers], ISBN 978-0-12-083180-7, MR 0690578
- [5] K. Binder (2001), ”Ising model”, in Hazewinkel, Michiel (ed.), Encyclopedia of Mathematics, Springer Science+Business Media B.V. / Kluwer Academic Publishers, ISBN 978-1-55608-010-4
- [6] Stephen G. Brush (1967), History of the Lenz-Ising Model. Reviews of Modern Physics (American Physical Society) vol. 39, pp 883–893. (10.1103/RevModPhys.39.883)

-
- [7] Nicholas Metropolis et Stanislaw Ulam, Journal of the American Statistical Association, vol.44, no 247, (1949), p.335-341
- [8] Onsager, Lars (1944), "Crystal statistics. I. A two-dimensional model with an order-disorder transition", Physical Review, Series II, 65 (3-4) : 117-149, doi :10.1103/PhysRev.65.117
- [9] Ruderman, M. A. ; Kittel, C. (1954). "Indirect Exchange Coupling of Nuclear Magnetic Moments by Conduction Electrons". Physical Review. 96 : 99.
- [10] https://en.wikipedia.org/wiki/Hexagonal_lattice
- [11] https://en.wikipedia.org/wiki/Trihexagonal_tiling#Kagome_lattice