

ANNÉE UNIVERSITAIRE 2022/2023

TP numérique de Physique statistique

Licence 3 Physique & Physique-chimie

Ces TP ont pour but de montrer quelques applications de l'outil numérique pour résoudre des problèmes de physique. En effet, à mesure que les problèmes se complexifient, il est souvent impossible de résoudre les équations analytiquement. On utilise alors un modèle numérique pour pouvoir tout de même faire des prédictions théoriques.

Ces TP contiennent un grand nombre de questions afin de montrer l'étendue des possibilités offertes par cet outil. Certaines de ces questions sont obligatoires (~ 10 points par TP, le barème est donné à titre indicatif et peut évoluer) ; les autres sont **optionnelles**.

Vous devez donc commencer par répondre aux questions obligatoires, puis vous pourrez ensuite **choisir** les autres questions que vous souhaitez développer (suivant vos affinités).

Des points (~ 5) sont aussi alloués sur la clarté du compte rendu et du code.

Les compte rendus individuels doivent rester concis et être rendus au format pdf accompagnés de codes qui se doivent d'être le plus clair possible (avec des commentaires !).

Un fichier de module est fourni pour chaque TP (`ising.py` et `brownian.py`), contenant des fonctions de visualisation et les fonctions de base de l'algorithme. Certaines de ces fonctions devront être modifiées (pensez donc à envoyer ce fichier avec le compte rendu !).

Vous pourrez ensuite utiliser ces fonctions dans des petits scripts ; idéalement 1 par question.

Vous pouvez copier-coller le fichier `TPX.example.py` pour vous servir de base.

Vous devrez déposer sur le serveurs vos fichiers, NOMMÉS AVEC VOS INITIALES (XY):

- Votre compte rendu au format PDF (`CRXY.pdf`)
- Le module `ising(brownian)XY.py` que vous aurez mis à jour
- Vos scripts pour les réponses aux questions, au format `TP1ou2_NM_XY.py` pour la réponse à la question M de la partie N (vous pouvez vous inspirer du fichier `TP1.example.py`).

Vous trouverez page suivante quelques ressources pouvant vous aider à la compréhension des TP.

Il n'est bien entendu pas nécessaire de tout lire, mais vous pouvez vous y référer si besoin.

D'autres ressources, plus complètes, sont disponibles sur le Moodle du TP.

Ressources Python :

Site officiel Python :

<https://docs.python.org/fr/3/>.

Mémento Python3 (2 pages) :

https://perso.limsi.fr/poinal/_media/python:cours:mementopython3.pdf

Résumé de la syntaxe Python (28 pages) :

http://www.xavierdupre.fr/site2013/documents/python/resume_utile.pdf

Référence du module numpy :

<https://docs.scipy.org/doc/numpy-1.16.1/reference/>

Ressources physique statistique :

Résumé de cours (22 pages) :

http://lptms.u-psud.fr/christophe_texier/files/2019/02/resume-cours-ps.pdf

Manuscrit d'étudiante sur le modèle d'Ising (21 pages) :

<https://www.ceremade.dauphine.fr/~viger/Memoire2018Fraval.pdf>

Résumé de physique statistique hors équilibre (pour le mouvement Brownien, 21 pages) :

https://www.lpp.polytechnique.fr/IMG/pdf_Introduction_Cours_Hors_Equilibre_2008.pdf

Les questions du TP1 sont en partie inspirées des documents suivants :

Cours de l'Université d'Orsay :

https://www.lps.u-psud.fr/IMG/pdf_Chapitre-Ising.pdf

TP d'Alexandre Mayer de l'Université de Namur :

<http://perso.fundp.ac.be/~amayer/Cours/ApprocheNumerique/Ising/Ising.pdf>

Simulation du site femto-physique.com :

<https://femto-physique.fr/simulations/ising2D.php>

Les questions du TP2 sont en partie inspirées des documents suivants :

Article de blog sur le mouvement Brownien :

<http://physiquereussite.fr/le-mouvement-brownien/>

TP1 : Modèle d'Ising 2D

I. Présentation du modèle d'Ising

Certains matériaux présentent une aimantation spontanée M lorsque la température T passe en-dessous d'une température critique T_c (dite température de Curie). Afin de décrire la transition de phase (cf TD 1, exercice VI) entre l'état paramagnétique ($M = 0$ pour $T > T_c$) et l'état ferromagnétique ($M \neq 0$ pour $T < T_c$), Ising et Lenz ont introduit dans les années 1920 un modèle où les atomes interagissent *via* les moments magnétiques de spin. Dans une version simplifiée du modèle, les spins sont placés sur les nœuds d'un réseau carré à deux dimensions, de taille $N \times N$. On suppose que chaque spin $S_i = \pm 1$ interagit uniquement avec ses voisins immédiats, l'hamiltonien du système étant :

$$H = -J \sum_{\langle i,j \rangle} S_i S_j . \quad (1)$$

La notation $\langle i, j \rangle$ signifie que les sites i et j sont **plus proches voisins**, chaque paire n'étant comptée **qu'une seule fois**. La constante de couplage $J > 0$ provient des interactions quantiques entre particules identiques.

L'aimantation totale est donnée par :

$$M = \sum_{i=1}^{N_s} S_i , \quad (2)$$

où $N_s = N \times N$ est le nombre de sites.

On va s'intéresser à la *magnétisation* ou aimantation moyenne par site : $m = M/N_s$.

I.1. Faire un schéma. Dans quelle configuration l'énergie (E ou H) est-elle minimale ? (1 point)

I.2. Déterminer (sans calcul) l'aimantation moyenne dans les limites $T \gg T_c$ et $T \ll T_c$. (1 point)

I.3. Comment pourrait on ajouter un champ magnétique extérieur à ce modèle ?

Celui ci doit favoriser l'alignement des spins dans le sens du champ extérieur.

On pourra noter h ce champ magnétique extérieur. (1 points)

I.4. Par quel ensemble statistique peut être décrit ce système ? (microcanonique, canonique, grand-canonique, ...) Justifier. (0.5 point)

Question Bonus :

I.5. Justifier le choix d'une constante de couplage J positive dans le cadre d'une transition

ferromagnétique-paramagnétique. Que pourrait on décrire avec une constante de couplage négative ?

(0.5 points)

II. Algorithme de Metropolis et évolution de l'aimantation

Bien qu'il existe une solution exacte du modèle d'Ising à deux dimensions, les calculs sont extrêmement complexes. Il s'avère donc plus judicieux de développer une approche numérique. On se propose de simuler le modèle d'Ising par la méthode de Monte-Carlo : il s'agit d'une méthode **probabiliste** qui permet d'explorer l'espace des configurations et donc de calculer les propriétés physiques.

Le principe de la méthode est le suivant :

- À $t = 0$ l'algorithme génère une configuration aléatoire de spins sur la grille.
- On itère ensuite un certain nombre de fois (jusqu'à l'obtention d'un certain équilibre) :
 - On choisit un spin $S_{i,j}$ sur la grille (avec une probabilité de choix uniforme)
 - On calcule la différence d'énergie ΔE qui interviendrait si ce spin était retourné
 - o Si ce retournement est favorisé énergétiquement ($\Delta E \leq 0$), on l'accepte ($S_i = -S_i$).
 - o Si il n'est pas favorisé énergétiquement ($\Delta E > 0$), il peut tout de même être accepté avec une probabilité $\propto \exp[-\beta\Delta E]$ où $\beta = \frac{1}{k_B T}$.
Cette dernière option modélise l'effet aléatoire d'une l'agitation thermique (selon une loi de probabilité de Boltzman) sur le retournement d'un spin.

La nouvelle configuration est donc acceptée avec la probabilité $\propto \min(1, e^{-\beta\Delta E})$. Cette méthode permet de mesurer les moyennes de quantités physiques. Plus de détails sont fournis en annexe.

La boucle est codée en Python dans la fonction `move_config` du fichier `ising.py` mis à votre disposition. Pour éviter les effets de corrélation, on effectue un nombre N_{corr} d'itérations entre chaque mesure des propriétés physiques. On impose également des **conditions aux limites périodiques** au système afin de limiter les effets de bord.

Dans le programme, toutes la variables sont **adimensionnées** : les énergies sont exprimées en unité de la constante de couplage J (autrement dit on pose $\tilde{E} = E/J$), et les températures en unité de J/k_B (i.e. on définit $\tilde{T} = k_B T/J$). Dans la suite, **on oubliera systématiquement le tilde**. Les temps sont exprimés en nombre d'itérations `Nmove`. Pour un réseau de taille 64×64 , $J = 1$ et $N_{corr} = 1000$, l'équilibre est souvent atteint pour $N_{move} \sim 3000$. On utilisera donc ces valeurs pour la suite du TP.

II.1. Dans la fonction `move_config`, expliquer l'origine du facteur 2 dans l'expression de ΔE (0.5 point)

II.2. On commence par observer l'évolution temporelle de la structure magnétique du réseau. Décrire le comportement du système lorsque la température est diminuée progressivement de $T = 5$ jusqu'à $T = 1$. Vous pouvez illustrer votre réponse par des cartes 2D. (2 points)

II.3. Utilisez la fonction `calculate_magnetization` pour tracer la *magnétisation* $m(t)$ d'une configuration ou aimantation moyenne par site, en fonction du temps, pour $T = 1$. Commentez. (2 points)

II.4. Comment déterminer le temps de relaxation (temps jusqu'à l'équilibre) ? (1 point)

II.5. Comme en TD, ajoutez un nouveau terme $-hM$ dans la fonction `move_config` modélisant un couplage des spin à un champ extérieur h .

Observez le comportement de $m(t)$ pour $h = 1$ avec $T = 1$ et $T = 4$.

Comparez ce comportement avec $h = 0$. (1 points)

Questions Bonus :

II.6. Compte tenu du nombre de sites du réseau simulé par rapport au nombre d'atomes dans un matériau réel, justifier l'utilisation de conditions aux limites périodiques. (0,5 point)

II.7. Pour déterminer l'état d'équilibre, expliquez l'intérêt de la méthode Monte-Carlo par rapport à l'exploration complète de l'espace des configurations. (0,5 point)

II.8 Ajoutez un nouveau paramètre J dans la fonction `move_config` permettant de modifier la valeur de la constante de couplage. Observez le comportement du système lorsque $J < 0$ de $T = 5$ à $T = 1$. Comparez ce comportement avec le cas des valeurs $J > 0$. (1 points)

III. Propriétés d'équilibre : aimantation et susceptibilité

Atteindre l'équilibre thermodynamique requiert d'effectuer un certain nombre d'itérations afin que le système perde la mémoire de son état initial. On mesurera donc les propriétés physiques à l'équilibre, après avoir effectué une boucle préliminaire au début du programme pour thermaliser le système.

Par valeur moyenne, on entend maintenant moyenne thermodynamique telle que vu dans votre cours, c'est-à-dire moyenne sur l'ensemble des configurations, ou, ici, sur un échantillon (bien choisi grâce à l'algorithme de Metropolis, voir Annexe).

Pour tracer les courbes, on pourra utiliser la fonction `plot_curve` du module `ising`.

III.1. Complétez puis utilisez la fonction `calculate_mean_magnetization` pour tracer valeur moyenne de la valeur absolue de la magnétisation $\langle |m| \rangle$ en fonction de la température T , puis du paramètre $p = 1 - \exp(-2/T)$

Ce paramètre p permet de tracer la gamme de température de 0 à $+\infty$ en échantillonnant p de 0 à 1.

Déterminer graphiquement la température critique T_c et le paramètre critique p_c .

Comparez à la valeur théorique $T_c = \frac{2}{\operatorname{arcsinh}(1)} \simeq 2.269$ et $p_c = 2 - \sqrt{2} \simeq 0.586$. (2,5 points)

III.2. Complétez la fonction `calculate_magnetic_susceptibility` afin de calculer la susceptibilité magnétique

$$\chi_m = \frac{\langle m^2 \rangle - \langle |m| \rangle^2}{T} (= (\frac{\partial m}{\partial h})_T) = -(\frac{\partial^2 F}{\partial h^2})_T \text{ où on a toujours la même définition } m = \frac{1}{N_s} \sum_{i=1}^{N_s} S_i \quad (3)$$

Tracez χ_m en fonction de T . Commentez le comportement de cette fonction. (3 points)

III.3. Modifiez la fonction `calculate_magnetization_theoretical` afin de lui faire calculer la solution théorique

$$m^{th}(T) = \begin{cases} \left[1 - \frac{1}{\sinh^4 2/T} \right]^{1/8} & \text{si } T < T_c \\ 0 & \text{si } T > T_c \end{cases} \quad \text{L. Onsager, Physical Review 65 (1944) (4)}$$

Tracez sur un même graphique la magnétisation moyenne (en valeur absolue) en fonction de la température obtenue numériquement et la solution théorique. Commentez. (2 points)

III.4. Completez la fonction `calculate_heat_capacity` afin de lui faire calculer la capacité calorifique.

On donne

$$C_v(= \frac{d\langle E \rangle}{dT}) = \frac{\langle E^2 \rangle - \langle E \rangle^2}{T^2} . \quad (5)$$

On pourra décomposer le problème en implémentant les fonctions `calculate_energy` et `calculate_mean_energy`, comme ont été définies les fonctions `calculate_magnetization` et `calculate_mean_magnetization` .

Tracez la capacité calorifique à volume constant en fonction de la température.

III.5. Invariance d'échelle - Finite size scaling. (3 points)

Kurt Binder, en utilisant les résultats du groupe de renormalisation (Kennet Wilson, Nobel 1982, mais aussi Michael Fisher et Leo Kadanoff), a remarqué que la quantité

$$K_2(N, T) = \frac{\langle m^2 \rangle}{\langle |m| \rangle^2} \quad (6)$$

ne dépendait pas de N la taille du système lorsque $T = T_c$ (il y a plusieurs propriétés magiques à T_c , en premier lieu l'invariance d'échelle). Cette propriété permet de déterminer précisément T_c .

Vous avez déjà construit tous les outils vous permettant de tracer $K_2(N, T)$ pour des différentes valeurs de taille du système.

Vérifiez que les courbes obtenues pour différentes valeurs de N se croisent bien toutes à T_c (ou p_c).

Annexe : A propos de l'algorithme de Metropolis

Dans l'ensemble canonique, le calcul de la valeur moyenne d'une quantité physique à l'équilibre nécessite de sommer sur toutes les configurations $\{\mu\}$:

$$\langle A \rangle = \frac{\sum A_\mu e^{-\beta E_\mu}}{\sum e^{-\beta E_\mu}} . \quad (7)$$

Or dans ces deux sommes, beaucoup de configurations sont numériquement "inefficaces", car associées à un facteur $e^{-\beta E_\mu}$ exponentiellement faible. Lorsqu'on procède à un échantillonnage non-uniforme de l'espace des configurations avec des probabilités $\{p_\mu\}$, l'estimation de la valeur moyenne s'obtient maintenant en utilisant

$$\langle A \rangle = \frac{\sum_{i=1}^N A_{\mu_i} p_{\mu_i}^{-1} e^{-\beta E_{\mu_i}}}{\sum_{i=1}^N p_{\mu_i}^{-1} e^{-\beta E_{\mu_i}}} \quad (8)$$

où N est la taille de l'échantillonnage. Certaines configuration sont sur(sous)-représentées. L'échantillonnage préférentiel consiste justement à privilégier les états qui ont un fort poids pour la formule (7), en prenant

$$p_\mu \propto e^{-\beta E_\mu} \text{ de sorte que } p_\mu^{-1} e^{-\beta E_\mu} = cste.$$

L'estimation de la valeur moyenne s'écrit alors

$$\langle A \rangle = \frac{1}{N} \sum_{i=1}^N A_{\mu_i} . \quad (9)$$

L'algorithme de Nicholas Metropolis génère un échantillonnage vérifiant cette distribution de probabilité, par l'utilisation de probabilités de transition entre configurations satisfaisant l'équation de *bilan détaillé*

$$\frac{P(\mu \rightarrow \nu)}{P(\nu \rightarrow \mu)} = e^{\beta(E_\mu - E_\nu)} . \quad (10)$$