



LICENCE PHYTEM

STAGE DE L3

30 MAI - 1 JUILLET 2011

Simulation de la dynamique mol culaire des disques durs

Laboratoire d'accueil :
LABORATOIRE DE PHYSIQUE STATISTIQUE
24 rue Lhomond Paris 5^e

Stagiaire :
Maxim BERMAN

Ma tre de stage :
Werner KRAUTH

1. PRÉSENTATION DU LABORATOIRE ET DE L'ÉQUIPE

Le LPS est le Laboratoire de Physique Statistique de l'ENS. Intégré au sein du département de physique de l'ENS, il se compose de 37 chercheurs et de doctorants. Il regroupe 11 équipes construites autour de thèmes de recherche variés touchant à la physique statistique, tels que la théorie de la matière condensée, la physique non-linéaire et les systèmes biologiques. La présence d'autres laboratoires importants tels que le LKB¹, le LPT-ENS² et le LPA³ dans le bâtiment est propice à la collaboration entre les chercheurs et à l'interaction entre les composantes théoriques et expérimentales de la physique.



FIGURE 1.1 – Le département de physique.



Werner KRAUTH, mon maître de stage, fait partie de l'équipe « Théorie de la matière condensée ». Directeur du département de physique, il m'a proposé ce stage après que j'aie envoyé un email au secrétariat du LPS. Il a sous sa responsabilité deux étudiants de thèse, qui travaillaient dans le même bureau que moi lors de mon stage : Etienne BERNARD et Swann PIATECKI.

Ancien élève de l'ENS Cachan, Etienne BERNARD termine sa dernière année de thèse. Il a travaillé sur un nouvel algorithme de type MONTE-CARLO pour le système de sphères dures à deux dimensions (disques durs), et a pu caractériser les transitions de phase de ce système grâce à cet algorithme.



Swann PIATECKI, élève de l'ENS dans sa première année de thèse, travaille quant à lui sur des simulations de type MONTE-CARLO appliqués à des systèmes de bosons et des condensats de BOSE-EINSTEIN. Il a récemment coécrit un article, en collaboration notamment avec des expérimentateurs du LKB.

L'équipe dispose en outre d'une salle de calcul représentant un total de 128 processeurs toujours en fonctionnement.

1. Laboratoire Kastler-Brossel
2. Laboratoire de Physique Théorique
3. Laboratoire Pierre Aigrain

2. SUJET DU STAGE

Le modèle du système de sphères dures à deux dimensions (ou *disques durs*) est un des modèles fondamentaux de la physique statistique. Il présente une transition de phase analogue à celle observée expérimentalement dans les cristaux liquides, dans des colloïdes, et aux interfaces de solides. La nature de cette transition de phase, étudiée abondamment depuis sa découverte¹ par ALDER et WAINWRIGHT en 1957, a suscité de nombreuses interrogations. Les systèmes à deux dimensions sont caractérisés par des grandes fluctuations, ce qui rend difficile les simulations numériques aux voisinages des transitions de phases. Ce débat a néanmoins été clos récemment par Etienne BERNARD et Werner KRAUTH grâce à un nouvel algorithme de type MONTE-CARLO, plus rapide et permettant thermaliser plus vite le système.²

2.1 Les simulations de type MONTE-CARLO

On cherche à générer des configurations aléatoires de disques durs. Une méthode consisterait à placer aléatoirement les disques et de ne garder que les configurations *valides* où ceux-ci ne se recouvrent pas. Malheureusement cette méthode est impraticable pour des densités raisonnables (trop peu de configurations valides sont générées).

On va alors plutôt partir d'une configuration *valide*, et s'en éloigner petit à petit en tentant à chaque étape de bouger un disque d'un petit déplacement aléatoire. Le mouvement est *accepté* si la configuration obtenue est valide, sinon on remet le disque à sa place. Cette méthode, appelée méthode de MONTE-CARLO par chaînes de MARKOV, génère bien la bonne distribution de probabilités quant à la répartition des sphères. Par le principe d'ergodicité d'un système à l'équilibre, on peut alors en tirer toutes les valeurs moyennes des observables.

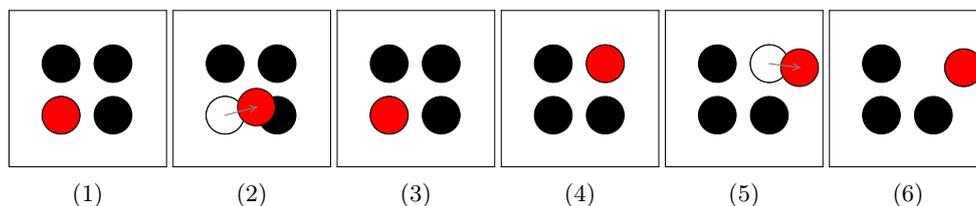


FIGURE 2.1 – (1), (2), (3) : Mouvement rejeté. (4), (5), (6) : Mouvement accepté.

2.2 Les simulations de dynamique moléculaire.

Au lieu de faire appel à la physique statistique pour la simulation, on peut choisir de simuler directement la dynamique newtonienne du système. Si une simulation directe a des qualités séduisantes – notamment les renseignements qu'elle peut nous apporter sur le système hors équilibre – elle paraît de prime abord bien plus coûteuse, voire impossible à mettre en œuvre pour simuler 10^6 particules. Néanmoins, on peut s'en sortir en étant astucieux.

1. BJ ALDER et T.E. WAINWRIGHT. “Phase transition for a hard sphere system”. Dans : *The Journal of Chemical Physics* 27 (1957), p. 1208.

2. E.P. BERNARD et W. KRAUTH. “First-order liquid-hexatic phase transition in hard disks”. Dans : *Arxiv preprint arXiv :1102.4094* (2011).

L'algorithme « événementiel » de ALDER et WAINWRIGHT ³

Pour simuler la dynamique moléculaire, on pourrait résoudre les équations différentielles par discrétisation du temps. Il y a pourtant un moyen simple de traiter le temps comme un paramètre continu. À chaque étape de calcul, on calcule le moment de la prochaine collision qui va avoir lieu (collision particule – particule ou particule – mur), puis on se place à ce temps précis avant d'effectuer la collision. On se déplace ainsi *d'évènement en évènement*.

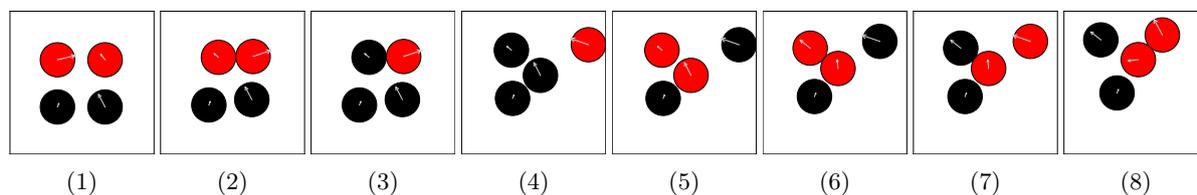


FIGURE 2.2 – (1,2), (5,6), (7,8) : Collisions de particules. (3,4) : Collision avec un mur.

Coût de l'algorithme. À chaque étape :

1. On calcule le temps de la prochaine collision chacune des N particules : cette opération est linéaire en N pour chaque particule (il faut considérer chacune des autres particules avec lesquelles elle pourrait collisionner), donc au total, *quadratique en N* .
2. On calcule le minimum des N temps de collision obtenus : opération linéaire en N .

On obtient donc un algorithme fonctionnant de manière *quadratique en N* à chaque étape : c'est bien plus coûteux que l'algorithme de MONTE-CARLO (qui ne fait qu'une opération à chaque étape), et cet algorithme « naïf » ne convient pas pour des simulations intéressantes (ayant de l'ordre de 10^6 particules).

Implémentation de cases

Pour des densités importantes, on peut se dire que la prochaine collision d'une particule ne se fera qu'avec des particules voisines. On peut exploiter cette observation en divisant la surface en cases. Lors du calcul du prochain temps de collision pour une particule, on n'envisagera que les collisions avec des particules situés dans sa case et dans les cases adjacentes (9 cases en tout).

Coût de l'algorithme. La complexité de l'étape 1. décrite plus haut devient linéaire en N : en effet, pour chacune des N particules, on n'a qu'un nombre constant de particules voisines à envisager pour la prochaine collision. L'algorithme est donc *linéaire en N* à chaque étape.

Utilisation de structures de données efficaces

À chaque étape, on calcule les temps de prochaines collisions pour les N particules, puis on effectue la prochaine collision. La collision entre deux particules est susceptible d'affecter uniquement le temps de prochaine collision des particules voisines de ces particules (situés dans les cases adjacentes). En mémorisant les temps de collision à chaque étape, on n'aura qu'un nombre constant de temps de collision à recalculer.

La seule opération coûteuse est alors le calcul du minimum sur les N temps de collision, dont seule une petite partie change à chaque étape. On peut alors faire le parallèle avec les *listes de priorité* des informaticiens, et tirer parti d'une structure de données en arbre, qui permet de recalculer le minimum de manière *logarithmique en N* .

Coût de l'algorithme. L'algorithme devient de complexité logarithmique en N : ce n'est pas sensiblement plus coûteux que la simulation par méthode de MONTE-CARLO !

3. B.J. ALDER et TE WAINWRIGHT. "Studies in molecular dynamics. I. General method". Dans : *The Journal of Chemical Physics* 31 (1959), p. 459.

3. TRAVAIL PERSONNEL

Mon travail a principalement consisté à programmer un algorithme de dynamique moléculaire de manière à prendre en compte le mieux possible des optimisations décrites plus haut. Mon équipe s'est pour l'instant principalement concentrée sur l'utilisation de méthodes de MONTE-CARLO, et il leur paraissait intéressant que j'aborde le problème par cet autre côté, et que je leur fournisse le code implémentant cet algorithme.

Je connaissais déjà les principales méthodes de simulation numérique grâce aux cours de FORTRAN à l'ENS CACHAN. Mais pour mettre en application et améliorer un algorithme relativement complexe, le FORTRAN devient très vite compliqué. Sous les conseils de M. KRAUTH, j'ai commencé par apprendre, en préparation du stage, le langage PYTHON, un langage « haut niveau » permettant de se focaliser d'avantage sur le principe de l'algorithme que sur les détails techniques du programme, et d'avoir à sa disposition des outils intuitifs permettant de plus vite passer de l'informatique à la physique du système. Je me suis également documenté sur la simulation numérique des systèmes de sphères dures, grâce au livre de mon maître de stage.¹

J'ai passé la première semaine de mon stage à développer l'algorithme « naïf » de dynamique moléculaire, avec des murs, et à créer un petit programme permettant une visualisation simple de la sortie, avec des petites animations (ayant notamment servi à produire la figure 2.2). J'ai également développé une version Python de l'algorithme de type MONTE-CARLO décrit plus haut, pour avoir un outil de comparaison. La figure 3.1 montre la distribution de probabilité des abscisses des sphères obtenues par les deux algorithmes, et le phénomène « d'attraction des bords » bien connu.

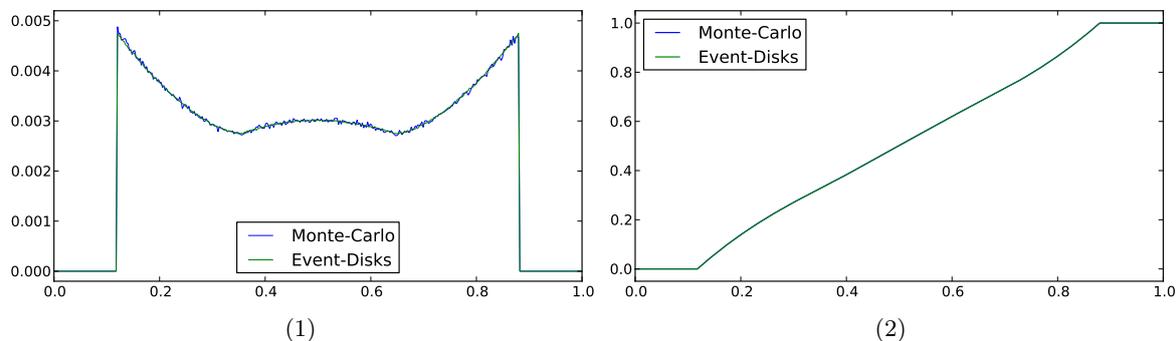


FIGURE 3.1 – Densité (1) et fonction de répartition (2) de la distribution des abscisses de 4 disques de densité 18% pour 10^7 étapes de calcul. Les deux échantillons passent le test de Kolmogorov-Smirnov avec un seuil de 1%.

Je me suis ensuite attaqué à la version optimisée utilisant une structure d'arbre pour calculer le minimum. Le problème consiste à calculer le minimum d'une liste, à changer quelques valeurs de cette liste, et à recalculer le minimum. WERNER KRAUTH m'avait présenté l'idée des « arbres tournoi » : comme dans un tournoi de tennis, les noeuds contiennent l'identité du « vainqueur », donc du temps le plus court parmi ses descendants (voir figure 3.2).

J'ai mis du temps à programmer cet algorithme. J'ai dû réfléchir à ses différents aspects : les structures de données à utiliser, les invariants à maintenir... Comme le gain de l'algorithme « en

1. W. KRAUTH. *Statistical mechanics : algorithms and computations*. T. 13. Oxford University Press, USA, 2006.

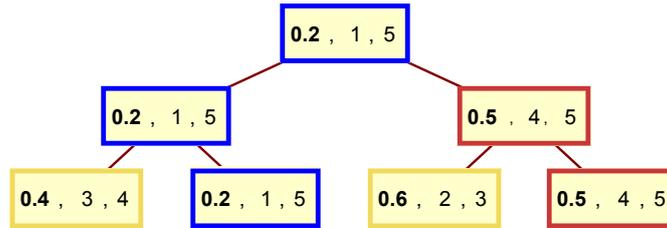


FIGURE 3.2 – Arbre Tournoi : le vainqueur est la collision entre la particule 1 et la particule 5 au temps 0.2. Si on ne change que la feuille bleue, seuls les minimas locaux des nœuds bleus devront être recalculés, laissant la partie droite de l’arbre intacte... Pour N particules cela revient à ne plus faire que $\log(N)$ calculs pour le calcul du minimum.

arbre » est nul en l’absence de cases (toutes les feuilles sont alors remises à jour à chaque étape), j’ai abordé en parallèle l’implémentation des cases. J’ai rencontré deux problèmes principaux :

- Comment ne calculer et ne stocker qu’une fois les collisions $i - j$ et $j - i$?
- Comment faire en sorte que la case auquel un disque appartient soit bien mise à jour au cours de la simulation ?

J’ai d’abord voulu régler le premier problème en ne calculant que les collisions d’une sphère avec les sphères d’abscisse supérieure à la sienne, mais cette condition s’est révélée difficile à maintenir dans le temps ; j’ai donc décidé d’ordonner les sphères en les numérotant, et de n’envisager que les collisions $i - j$ avec $i < j$. Quant au deuxième problème, je l’ai réglé en déclenchant des « événements » lors de la collision d’une sphère avec les bords de sa case, en supplément des événements déclenchés par la collision de deux sphères. Ces événements me permettent de mettre à jour la case associée à la sphère, et à implémenter des limites périodiques en déplaçant une sphère qui sortirait de l’aire de la simulation.

Au bout de deux semaines de programmation de diverses structures de données, j’ai fait le lien entre l’algorithme et les *listes de priorité*, une structure de donnée informatique optimisée pour la recherche de minimum sur les données insérées. Une fois cela compris, j’ai découvert le module `HEAPQ` de `PYTHON`, implémentant des *tas*, une structure d’arbre permettant d’implémenter des listes de priorité de manière efficace. Assez proches des arbres tournois, les tas permettent également des recherches de minimum en temps logarithmique. Au bout de quelques jours, j’avais un algorithme fonctionnel utilisant ce module.

J’ai ensuite tenté quelques améliorations. Il est par exemple inutile de considérer les collisions avec les cases en tant qu’événements si la largeur des cases est grande devant le libre parcours moyen : il suffira de mettre à jour les cases au cours des collisions, dont une se produira avant que la particule sorte du carré formé par ses cases adjacentes. Cela suppose d’étudier la distribution du libre parcours moyen, et j’ai vérifié qu’elle ressemblait à une distribution exponentielle, quelle que soit la densité, conformément à la littérature.² Si ce changement simplifie le code du programme, les essais que j’ai réalisés suggèrent qu’elle rend l’algorithme moins performant : il vaut mieux générer des événements de cases pour pouvoir utiliser des tailles de cases plus petites, donc avoir moins de voisins à considérer.

Enfin, j’ai mis en place une version 1D de l’algorithme, écrite en `PYTHON` de la manière la plus concise possible (moins de 50 lignes), afin de montrer uniquement les concepts importants, et notamment l’utilisation de la liste de priorité. L’image en arrière-plan de la page de garde a d’ailleurs été produite par ce programme : elle représente l’évolution de la position de 10 « segments durs » dans le temps.

2. BJ ALDER et T. EINWOHNER. “Free-Path Distribution for Hard Spheres”. Dans : *The Journal of Chemical Physics* 43 (1965), p. 3399.

4. BILAN PERSONNEL

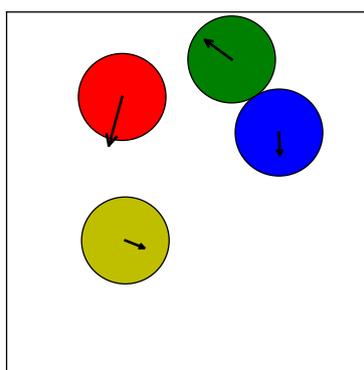
J'ai beaucoup apprécié ce stage. Le sujet était une fusion très intéressante entre un système fondamental de physique et un concept fondamental d'informatique, la liste de priorité. Je regrette que le stage ait été trop court pour aborder le « vrai » travail qu'on pourrait effectuer sur cette simulation, notamment sa comparaison avec l'algorithme « MONTE-CARLO global » développé par l'équipe, son optimisation et son exploitation pour étudier le système physique en profondeur. Néanmoins, je suis content d'en avoir pu aborder quelques aspects fondamentaux.

Bien que les cours de simulation numérique en FORTRAN à l'ENS CACHAN m'aient été utiles, ce stage m'a décidé à ne jamais commencer un projet de simulation par une version FORTRAN. Il vaut mieux commencer par mettre en place une version avec un langage tel que PYTHON, bien plus commode à utiliser, et compréhensible quant aux erreurs qu'il produit. Ces avantages compensent largement la relative lenteur de PYTHON, qui n'est pas significative dans la plupart des utilisations et peut d'ailleurs être corrigée à l'aide d'optimisations.

Le système physique des disques durs m'a en lui-même beaucoup intéressé. Je trouve très étonnante la richesse des comportements qui peuvent émerger de modèles physiques aussi simples en apparence, à l'exemple des transitions de phases dans le système 2D. Je ne pensais pas que je pourrais m'émerveiller devant des collisions de segments durs en 1 dimension. Je trouve aussi intéressant le fait que le comportement d'un système puisse changer radicalement avec la dimension : le système des sphères dures ne possède pas de transition de phase en une dimension,¹ en possède deux rapprochées en 2 dimensions² et en possède une du premier ordre pour les dimensions supérieures.³

J'ai apprécié le cadre de travail au LPS et l'ambiance au sein de l'équipe. Il était intéressant d'avoir un aperçu du travail des deux doctorants, qui ont su répondre à toutes mes questions avec bonne volonté.

Je remercie M. KRAUTH de m'avoir proposé ce stage, et de m'avoir apporté ses explications et son aide. Je remercie également E. BERNARD et S. PIATECKI avec qui j'ai passé un stage agréable. Enfin, je remercie les responsables de la formation PHYTEM pour avoir organisé ces stages de fin d'année.



1. L. van HOVE. “Sur L'intégrale de Configuration Pour Les Systèmes De Particules À Une Dimension”. Dans : *Physica* 16.2 (1950), p. 137 –143. ISSN : 0031-8914.

2. BERNARD et KRAUTH, *op. cit.*

3. Reimar FINKEN, Matthias SCHMIDT et Hartmut LÖWEN. “Freezing transition of hard hyperspheres”. Dans : *Phys. Rev. E* 65.1 (2001), p. 016108. DOI : 10.1103/PhysRevE.65.016108.