

Dynamique moléculaire de disques durs

Samuel Diebolt

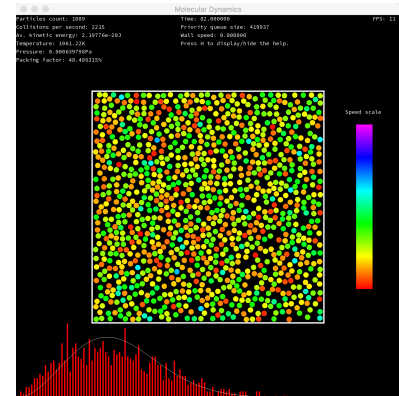
18 mai 2018

Le but de cet exercice est de réaliser une simulation de dynamique moléculaire : on considère un ensemble de particules à l'intérieur d'une boîte à deux dimensions où l'énergie cinétique est conservée. Ces particules répondent donc aux lois des chocs élastiques.

Description succincte du programme

Ci-dessous sont décrites les classes, fonctions et variables les plus importantes du programme, afin d'en comprendre le fonctionnement général. L'intégralité du code est par ailleurs commenté afin d'en faciliter la lecture.

- **class CollisionSystem** : cette classe est la description du système de simulation.
 - **CollisionSystem::Simulate** : fonction de simulation principale, comprenant la boucle de simulation, la gestion des événements, l'appel aux fonctions d'affichage, etc.
 - **std::priority_queue pq_** : file de priorité permettant de trier les événements (collisions particule-particule, affichage, etc.).
 - **CollisionSystem::Predict** : fonction de prédiction des événements futurs – collisions entre particules et particule-paroi – pour une particule considérée.
- **class Particle** : cette classe est la description d'une particule : caractéristiques physiques, éléments graphiques, etc.
 - **Particle::TimeToHit...** : ces fonctions renvoient le temps devant s'écouler avant une collision entre la particule considérée et une autre particule ou une des parois.
- **class Event** : cette classe est la description d'un événement : collisions entre particules, entre particules et parois et affichage de l'état du système sur l'interface graphique.
 - **Event::IsValid** : afin d'optimiser les performances¹, chaque événement est traité comme suit :
 1. Le prochain événement dans la file de priorité dont les particules considérées n'ont pas subi de collisions entre cet instant et l'ajout antérieur de l'événement est considéré comme valide et va être traité.
 2. L'état du système est avancé jusqu'au temps de l'événement.



1. L'usage mémoire est toutefois considérable lorsque le nombre de collisions par seconde devient trop grand ou lorsque la simulation est laissée en fonctionnement pendant un long moment !

3. L'évènement est résolu et l'état du système est mis à jour en conséquence : collision, affichage, etc.
4. Les futurs évènements concernant la ou les particules considérées sont calculés et ajoutés à la file de priorité.

Ainsi, la file de priorité n'est jamais vidée et il n'est nécessaire de calculer que deux prédictions par tour de boucle : on passe d'une complexité de $O(N^2)$ à $O(N)$ ².

Implémentations

La simulation actuelle comporte les éléments suivants :

- Dimensionnement réalisé par les macros du fichier main.h.
- Possibilité de rendre le système non-élastique en ajoutant de la friction dans les chocs entre particules.
- Affichage des caractéristiques physiques de la simulation : température, pression, compacité.
- Affichage des sphères d'influence³.
- Affichage de l'histogramme des vitesses – normalisé par rapport à la vitesse la plus représentée – et de la fonction de densité de probabilité de Maxwell-Boltzmann.
- Changement de la taille de la boîte de simulation à l'aide des touches directionnelles⁴.
- Possibilité d'ajout d'une nouvelle particule de rayon moitié et de masse un quart à une position aléatoire.
- Affichage du chemin d'une particule – mouvement brownien⁵.

Observations et comportements

Loi de Maxwell-Boltzmann

On observe à partir de quelques centaines de particules que la forme de l'histogramme des vitesses est similaire à la fonction de densité de probabilité de Maxwell-Boltzmann et ce très rapidement après le lancement de la simulation, alors que les vitesses sont d'abord générées aléatoirement à l'aide d'une loi uniforme.

Détente adiabatique

Sans friction, en réalisant une compression lente puis une détente rapide sur un nombre conséquent de particules (quelques centaines), on observe la figure typique d'une détente adiabatique, les vitesses des particules les plus éloignées étant les plus hautes.

2. Cette complexité serait encore plus faible en utilisant un arbre pour le traitement des évènements plutôt qu'une file de priorité, ce qui permettrait notamment une suppression des évènements moins coûteuse.

3. Cette effet purement esthétique est basé sur la technique des metaballs (<https://fr.wikipedia.org/wiki/Metaballs>). Il est très coûteux en calculs et doit s'utiliser avec un nombre très limité de particules (5 tout au plus) pour que l'effet soit observé.

4. Cela cause des recouvrement de particules en l'état, du fait d'une mauvaise gestion des collisions sur les parois en mouvement.

5. Retirer l'affichage des particules à l'aide de la touche P rend l'observation du chemin plus aisée.

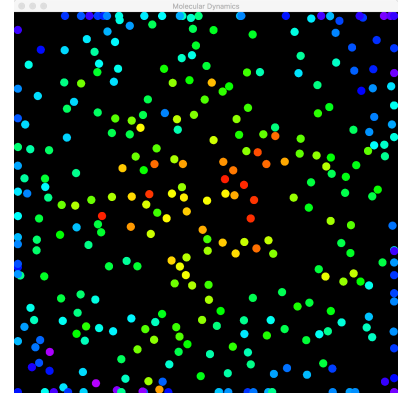


FIGURE 1: Détente adiabatique avec les paramètres 15 15 1

Avec friction en revanche, on observe une figure différente, la détente étant dirigée de façon symétrique selon les faces de la boîte. Cela n'est toutefois pas étonnant, du fait que les particules collées aux parois subissent moins les frictions et sont accélérées par le mouvement des parois.

Toutefois, du fait d'une mauvaise gestion des collisions avec une paroi en mouvement qui n'a pas pu être corrigée par manque de temps, ce phénomène pourrait être causé par un artefact informatique.

Mouvement de rotation

Cette dernière observation est moins commune : avec un coefficient de friction d'environ 0.5 à 0.7, alors que l'énergie cinétique moyenne devient très faible, on observe l'apparition d'un mouvement d'ensemble, semblable à une rotation, tandis que les particules semblent subir une force centrifuge les repoussant sur les parois de la boîte.

De la même manière, ce phénomène pourrait être un artefact informatique. Toutefois, un article de recherche – *A flowing crystal of self-propelled particles*, par Guillaume Briand, Michael Schindler et Olivier Dauchot – décrit un mouvement de rotation similaire avec un système de particules dans une boîte hexagonale.

Suites et recherches

Les différentes possibilités envisageables pour faire évoluer ce programme sont donnés dans le fichier TODO.md. Parmi elles, les plus intéressantes sont :

- Implémenter des collisions multi-particules.
- Implémenter l'expérience du démon de Maxwell.
- Utiliser un arbre bicolore plutôt qu'une file de priorité pour la gestion des événements.
- Paralléliser les phases de calcul.

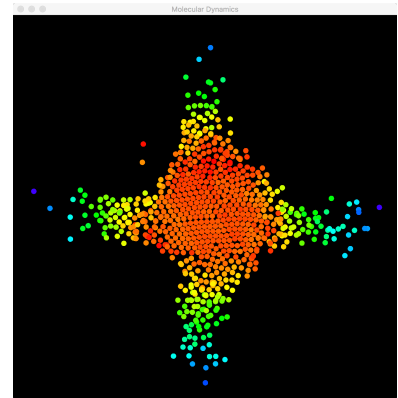


FIGURE 2: Détente adiabatique avec les paramètres 10 10 0.7

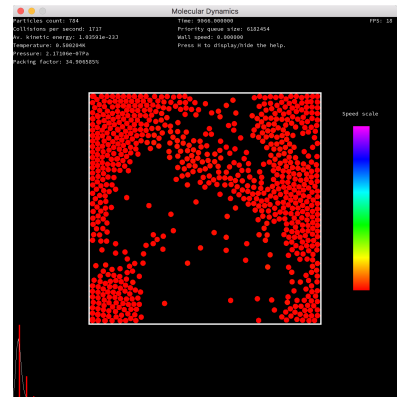


FIGURE 3: Mouvement d'ensemble avec les paramètres 10 10 0.6