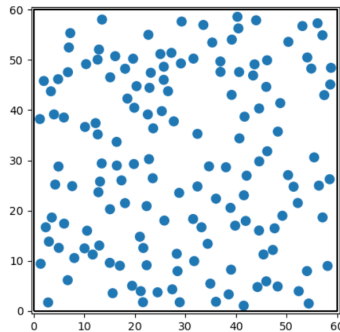


Simulations numériques : Thermal Conduction

CELIK Baran

Nous avons réalisé au cours des TP de simulations numériques un système de N particules assimilées à des sphères dures qui rebondissent entre elles et contre les murs de l'enceinte carrée que nous avons construite. Elles ont une vitesse initiale fixée, et une orientation initiale aléatoire.



Nous allons dorénavant chercher à améliorer ce code en modélisant une conduction thermique entre:
-le mur de gauche, dorénavant chaud
-le mur de droite, dorénavant froid

1 Modélisation des murs chauds et froids

Un chapitre du livre de Werner Krauth "Algorithms and Computations", chapitre "Hard disks and spheres", partie "Pressure and the Boltzmann distribution", va nous aider pour modéliser l'énergie à donner à une particule qui rebondit sur un mur à une température donnée. Une formule en particulier nous intéresse: $v_x = \sqrt{\frac{-2\log[random(0,1)]}{\beta}}$.

β étant égal à $\frac{1}{k_b T}$, et en prenant k_b égal à 1, on a donc la formule à implémenter dans le code:

$$v_x = \sqrt{-2\log[random(0,1)] * T} \quad (1)$$

Nous gardons donc les mêmes rebonds qu'avant sur les bords supérieurs et inférieurs de la boîte (si direction == 1), sinon si la sphère touche le bord gauche elle rebondit sur un mur chaud (si x.velocity < 0) et sinon elle rebondit sur le mur froid de droite.

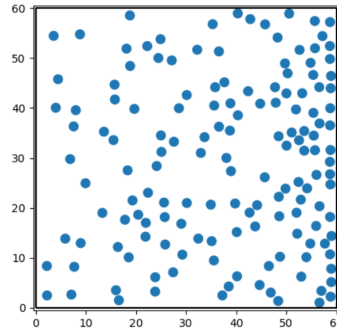
```
if direction == 1:
    self._velocity[particle,direction] = -self._velocity[particle,direction]
elif self._velocity[particle,direction] < 0:
    self._velocity[particle,direction] = np.sqrt(-2*np.log(r)*self.Tc)
else:
    self._velocity[particle,direction] = -np.sqrt(-2*np.log(r)*self.Tf)
```

Nous voilà fin prêts à étudier l'impact de la température des murs sur le système.

2 Mesure de la conductivité thermique

2.1 Mise en place des murs

Mettre en place le code précédent nous apporte plusieurs informations. Tout d'abord, la température n'a pas d'unité concrète ($T = 10$ correspond déjà à un mur chaud). Ensuite, nous observons avec les deux murs de températures différentes un gradient de densité: les boules semblent se serrer du côté froid et être très espacées du côté chaud. Cela paraît finalement instinctif, car le mur chaud leur donne de la vitesse pour repartir tandis que le mur froid absorbe leur vitesse.



2.2 Retrouver λ conductivité thermique

Nous nous plaçons dans une boîte carrée de largeur $L=60\text{ua}$. Notre idée pour mesurer un flux de chaleur est de placer un mur virtuel au centre de la boîte (donc en $x=30\text{ua}$), et de mesurer l'énergie cinétique des particules qui franchissent ce mur. En soustrayant l' E_c des particules qui franchissent le mur de droite à gauche (froid vers chaud) à celle des particules qui le franchissent de gauche à droite, on se retrouve avec un ΔE_c normalement positif. Celui-ci est une énergie que nous allons convertir en j flux de diffusion en J/S/m^2 en prenant $S=L*L$ (boîte cubique) et $t=0,2\text{ua}$ (variable simultime). Enfin, nous avons la loi de Fourier: $j = -\lambda \vec{\text{grad}}T$ avec $\vec{\text{grad}}T$ égal à $\frac{T_c - T_f}{L}$.

Nous pouvons enfin en extraire $\lambda = -\frac{E_c}{tL(T_f - T_c)}$.

2.3 Placer un mur virtuel en $x=30\text{ua}$

Pour implémenter ce mur virtuel nous décidons à chaque mdstep de noter une position antérieure, puis de déplacer les particules selon le code normal, puis de noter les particules qui ont franchi $x=30\text{ua}$ pendant ce déplacement en comparant leur position actuelle à leur position antérieure.

Les particules sont notées dans un tableau coldtohot avec un 1 si elles ont franchi le mur de droite à gauche pendant ce mdstep, et un 0 sinon. De même elles sont notées dans un tableau hottocold avec un 1 si elles ont franchi le mur de gauche à droite. Une ligne permet enfin de calculer l'énergie cinétique de ces particules et de les additionner/soustraire à la somme totale de transfert énergétique, et une autre garde l'évolution de l' E_c au cours du temps pour en tracer un graphe ultérieurement.

En plus de la ligne de code pour noter la position antérieure dans la variable "previousposition", le code prend donc cette allure-ci (nous prenons $m = 1$ pour simplifier):

```
hot_to_cold = np.where((previous_position[:, 0] < 30) & (self.position[:, 0] > 30), 1, 0)
cold_to_hot = np.where((previous_position[:, 0] > 30) & (self.position[:, 0] < 30), 1, 0)
self.E_cin += hot_to_cold*0.5*self._velocity[:, 0]**2 - cold_to_hot*0.5*self._velocity[:, 0]**2
self.graph_E += [sum(self.E_cin)]
```

2.4 Recherche des bonnes températures

Nous avons souhaité, en prenant comme référence $T_f = 1$, observer à partir de quel T_c un réel transfert d'énergie est observable. Pour cela nous prenons une simulation plus longue (nframes = 10 000) et

nous prenons des T_c proches de T_f . Nous constatons que, pour cette durée non-infinie de simulation, un réel transfert thermique est observable dès $T_c = 1.3$.

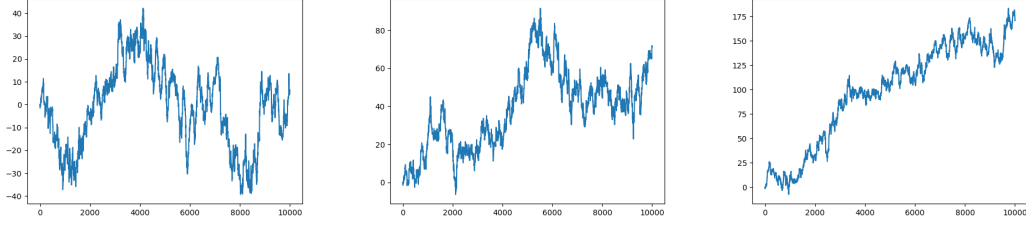


Figure 1: $T_c = 1.1, 1.2$ et 1.3

Nous décidons pour la suite des simulations de prendre $T_f = 1$ et $T_c = 10$ pour observer un réel transfert thermique dès 2000 nframes et un gradient de densité de particules visible.

2.5 Conductivité thermique comme fonction du nombre de particules

Nous noterons pour la suite que ΔE_c est égal à λ à une constante près. On va faire varier N le nombre de particules, faire des simulations pour chaque N et donc avoir l'évolution de E_c en fonction de N . On prend une taille des particules $\sigma = 1$, ainsi donc que 2000 nframes.

Voici des exemples de simulations pour $N=2, 10, 20, 200$

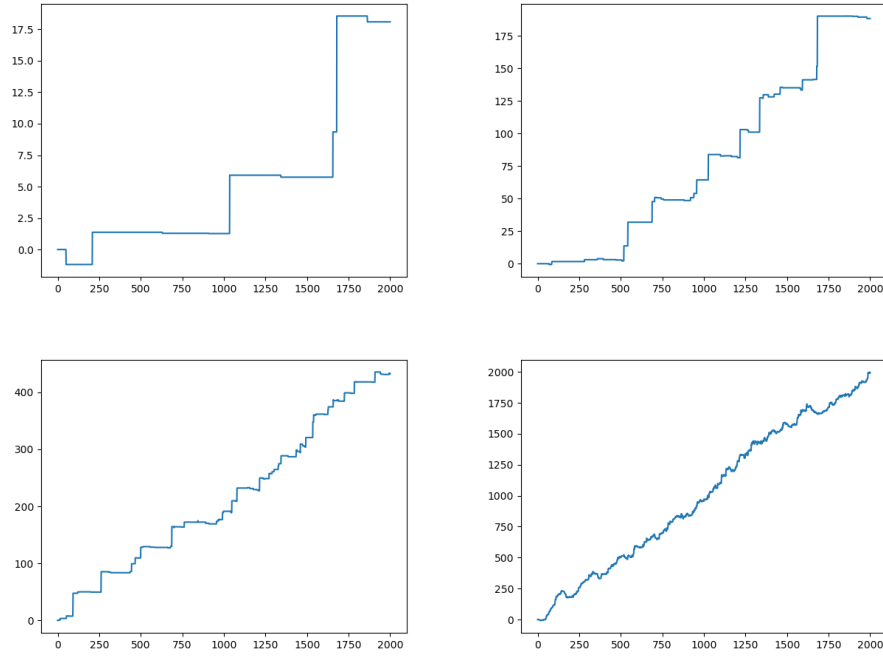
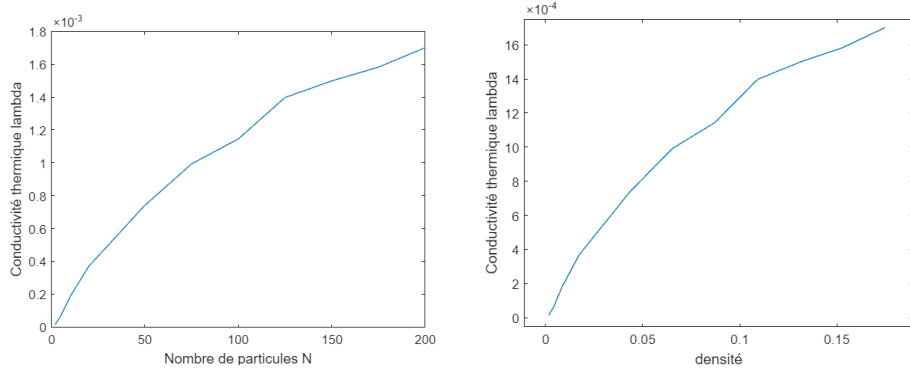


Figure 2: Evolution de E_c en fonction de N

On constate que plus N est grand plus la courbe est lisse, ce qui est logique car un nombre bien plus grands de particules franchissent notre mur virtuel. On extrait de ces graphes une pente égale à ϕ **flux thermique**, ensuite on a $\lambda = -\frac{\phi}{S \text{grad} T}$. On peut pour finir tracer λ en fonction de la densité ($= N * \pi \sigma^2$):

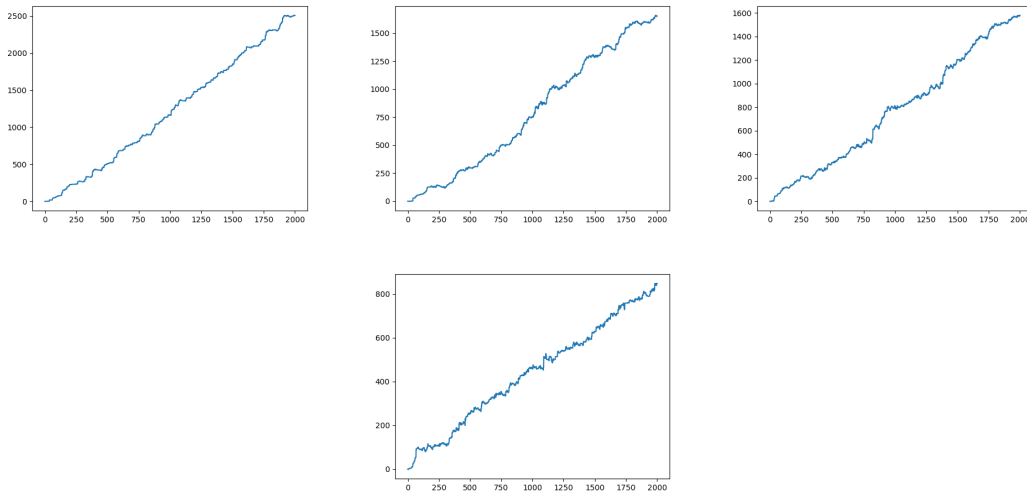


Le tracé ne semble pas linéaire néanmoins il est compliqué de monter vers des N plus grands car cela est limité par les capacités de simulation de l'ordinateur.

2.6 Conductivité thermique comme fonction du rayon des particules

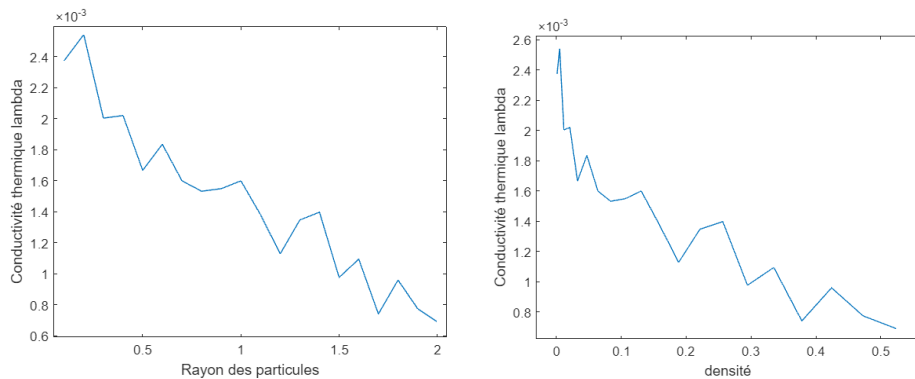
Cette fois nous faisons varier le rayon sigma des particules en gardant les mêmes données que dans la partie précédente et en fixant $N=150$ particules.

Nous traçons alors de nouveau E_c en fonction du temps pour $\sigma = 0.2, 0.8, 1.4, 2$:



Les tracés se ressemblent tous mais l'échelle verticale n'y est pas la même; il y a beaucoup moins de collisions lorsque les particules sont petites donc beaucoup plus d'énergie transférée facilement.

On peut enfin tracer $\lambda = f(R)$ et $\lambda = f(d)$.



Cette fois-ci la conductivité thermique décroît lorsque la densité augmente. Cela est normal sachant que les particules dans notre programme gardent une masse indépendante de leur taille, elle n'entre pas ici en jeu, et donc augmenter la taille des particules gêne ici simplement leur déplacement.

3 Ajout d'une particule plus large et plus lourde

A présent, on ajoute une unique particule plus large et plus lourde. Les murs n'ont pas changé : à gauche $T_c = 10ua$, à droite $T_f = 0.1ua$. On prend une masse de 1 pour les 150 petites particules de rayon $\sigma = 1$. On prend une particule plus grande de rayon $\sigma_{large} = 10$. Pour la masse de la particule, on simule avec deux valeurs: $m_{large} = 5$ et $m_{large} = 50$.

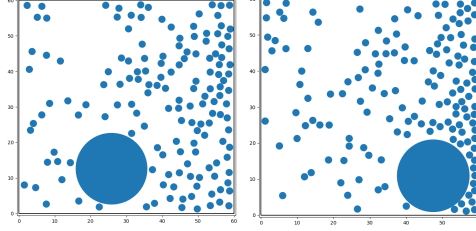


Figure 3: $m_{large}=5$ à gauche et $m_{large}=50$ à droite

On remarque que la particule 50 fois plus lourde que les petites est amenée vers le mur froid et les petites particules qui sont à sa droite ne peuvent pas la pousser vers le mur gauche (chaud). Tandis que pour la particule 5 fois plus lourde, elle peut être déplacée à droite comme à gauche par les petites particules. Si on garde cette particule lourde $m_{large}=50$ et qu'on change le rayon $\sigma_{large} = 3$ puis $\sigma_{large} = 20$:

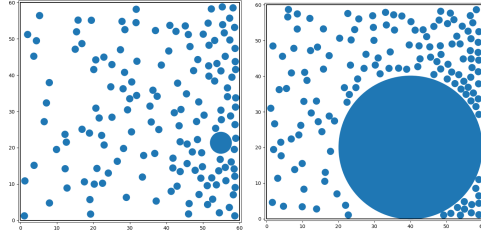


Figure 4: rayon $\sigma_{large} = 3$ à gauche puis $\sigma_{large} = 20$ à droite

Ainsi, pour une masse lourde fixe, le rayon n'influe pas sur le déplacement vers le mur froid de la particule large.

4 Conclusion

Nous avons pu constater que la conductivité thermique augmente avec le nombre de particules et décroît avec la taille des particules. La conductivité thermique ne dépend donc pas uniquement de la densité de particules, elle dépend de comment est répartie cette densité.

Si N augmente, λ augmente.

Pour m fixé, si R augmente, λ diminue.

Lorsqu'on ajoute une plus large et plus lourde particule, les petites particules, qui viennent du mur chaud, amènent la grande particule jusqu'au mur froid. Ainsi cette dernière est collée au mur froid et ne peut prendre de la vitesse pour se déplacer dans la boîte.