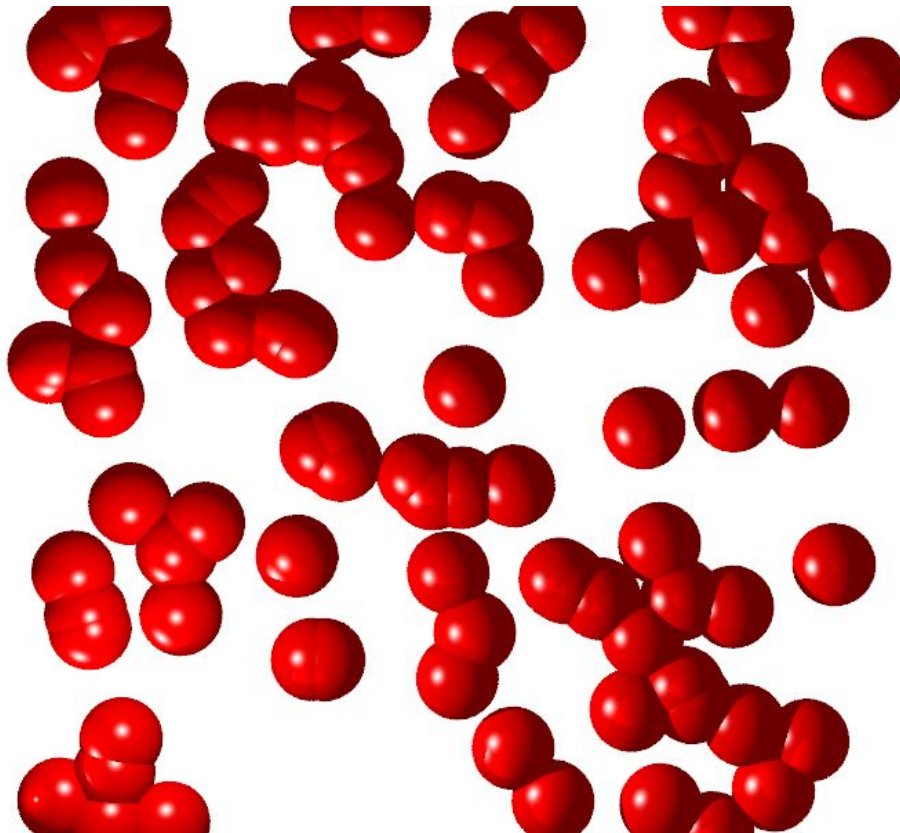


**Projet de Physique P6**

**STPI/P6/2017-045**

# **Dynamique d'un système à N corps**



**Etudiants:**

**Victorine BOUTOILLE**

**Emma ALIX**

**Ivana KOVACEVIC**

**Bozhidar PALASHEV**

**Enseignant-responsable du projet :**

**M. Jérôme YON**



Date de remise du rapport : **19/06/2017**

Référence du projet : **STPI/P6/2017 – 045**

Intitulé du projet : ***Dynamique d'un système à N corps***

Type de projet : ***simulation***

Objectifs du projet :

**Ce projet implique d'appréhender le concept de système à N corps puis de choisir et d'étudier un système en particulier afin de modéliser son fonctionnement.**

Mots-clefs du projet (4 maxi) :

***système à N corps, dynamique, interactions***

---

INSTITUT NATIONAL DES SCIENCES APPLIQUÉES DE ROUEN

Département Sciences et Techniques Pour l'Ingénieur

avenue de l'Université - 76801 Saint Etienne du Rouvray - tél : +33 (0)2 32 95 97 00 - fax : +33 (0)2 32 95 98 60

## TABLE DES MATIÈRES

<b>1. Introduction</b>	<b>4</b>
<b>2. Méthodologie / Organisation du travail</b>	<b>5</b>
<b>3. Travail réalisé et résultats</b>	<b>7</b>
I. Travail réalisé	7
1- Adaptation de modèle du gaz parfait en 2D	7
2- Développement du programme sur Scilab	9
3- Rendu sur Povray	10
II. Résultats	12
1- Premier cas, R petit	12
2- Deuxième cas, R grand	14
3- Comparaison et conclusion	15
<b>4. Conclusions et perspectives d'évolution</b>	<b>16</b>
<b>5. Bibliographie</b>	<b>18</b>
<b>6. Annexes</b>	<b>19</b>
1- Programme sur Scilab	19
2- Explication du programme	21
3- Programme sur Pov-Ray	22
a) Programme	22
b) Fichier texte	23
4- Démonstration des formules utilisées en 2D	24
a) Maxwell-Boltzmann	24
b) Energie cinétique, $\alpha$ et formule de la température	25
c) Loi des gaz parfaits	26
5- Suivi des séances	26

## 1. Introduction

Dans le cadre de notre projet de physique de deuxième année, nous avons choisi d'étudier la dynamique d'un système à  $N$  corps. Il n'a pas été simple de cerner concrètement notre sujet car nous avons tous notre idée sur ce qu'était un système à  $N$  corps et chacun d'entre nous avait son avis sur l'application à étudier. En effet, il existe en réalité de nombreux problèmes à  $N$  corps, par exemple dans l'espace ou encore entre les molécules d'un fluide.

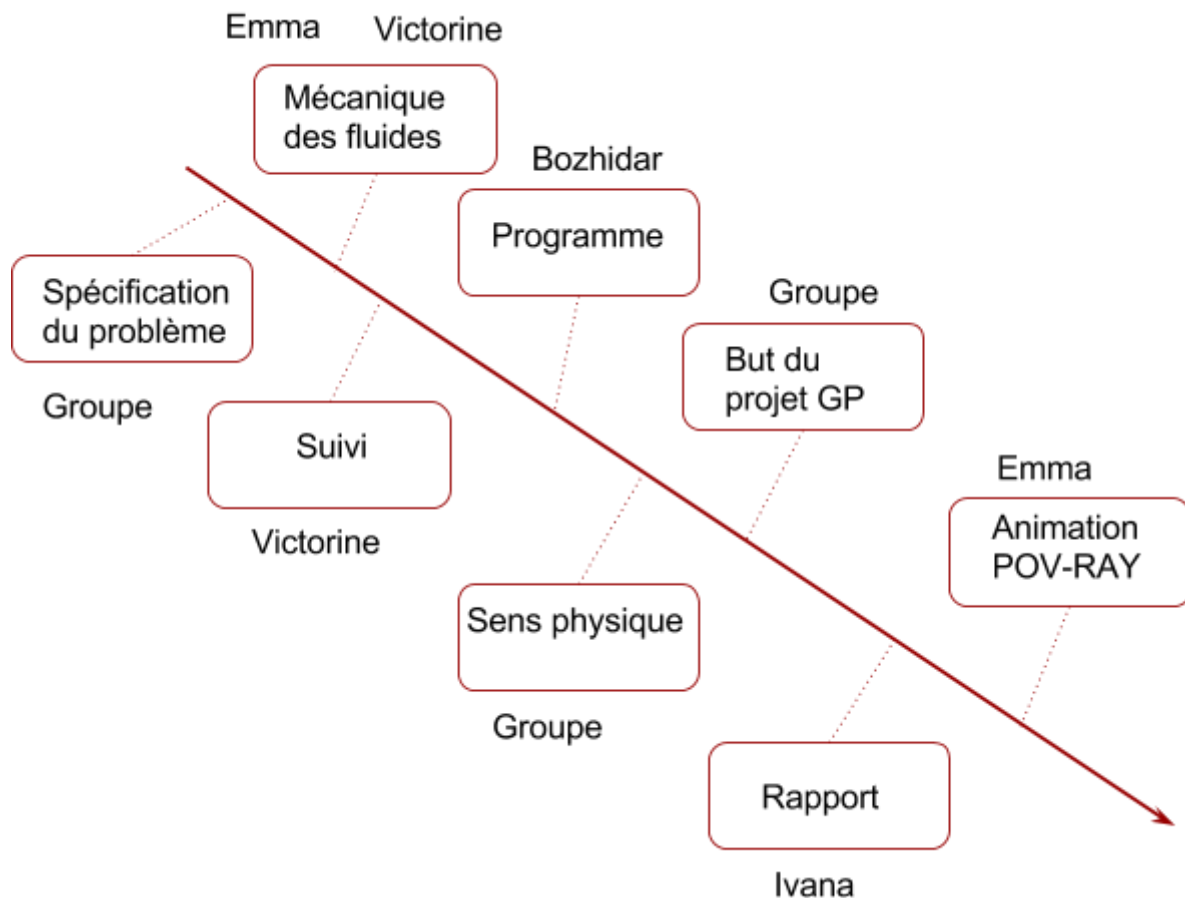
Tout d'abord, l'étude d'un "problème à  $N$  corps" consiste à déterminer le mouvement provoqué par les interactions des forces de  $N$  éléments. Si on ne prend que  $N=2$ , d'après Newton, les deux masses vont décrire des ellipses autour de leur centre de gravité commun. Mais si l'on passe à  $N=3$ , les trajectoires deviennent chaotiques, une légère variation dans les positions et vitesses initiales peut causer d'importantes différences dans la trajectoire des corps. Pour plus de 3 corps, il n'existe pas de solution analytique.

Nous avons déjà eu l'occasion d'étudier analytiquement des problèmes à deux corps en mécanique du point, mais si l'on souhaite s'intéresser à un problème à  $N$  corps, il nous faut utiliser des outils de calcul et de modélisation numérique car ce n'est pas faisable de façon analytique en raison de la quantité de force à prendre en compte. L'analyse numérique peut elle aussi être limitée par la puissance de calcul de l'ordinateur utilisé mais permet tout de même de parvenir à des résultats concrets et intéressants.

Notre projet consiste en l'étude de la dynamique de système à  $N$  corps. C'est un sujet relativement ouvert et nous avons pris le parti dès le départ de nous focaliser sur l'étude approfondie d'un système en particulier. Dans ce but et ayant déjà entendu parler en chimie ou encore en thermodynamique des notions de courbes des potentiels entre deux molécules, nous nous sommes penchés sur ce problème en particulier. Ainsi, dans le cadre de notre étude d'un système à  $N$  corps, nous avons choisi de modéliser numériquement les forces régissant les interactions des molécules au sein d'un gaz.

Tout d'abord nous allons vous présenter la façon dont nous nous sommes organisés et le fonctionnement de notre groupe, dans un second temps nous allons détailler l'évolution de l'étude du système et les résultats auxquels nous sommes parvenus. Enfin nous présenterons les conclusions que nous pouvons en tirer et ce que ce projet nous a apporté.

## 2. Méthodologie / Organisation du travail



Tout d'abord, lorsque nous avons commencé ce projet, il nous a fallu trouver un angle d'approche et choisir dans quelle direction nous souhaitions aller car le problème était assez large et devait être spécifié. Nous avons défini plusieurs possibilités en fonction de nos affinités pour la mécanique ou encore la chimie. Nous nous sommes finalement arrêtés sur l'idée d'interactions de molécules par exemple dans un gaz.

Ayant suivi le cours de théorie cinétique des gazs, Emma et Victorine se sont vite rendues compte que ce projet faisait appel à ces notions, notamment la distribution des vitesses de Maxwell-Boltzmann et se sont donc penchées sur leur emploi.

Notre projet impliquait forcément un programme, en réalité c'était même le principal objectif de cette modélisation, afin de visualiser les interactions. Bozhidar s'est beaucoup investi dans la partie informatique, afin de coder les notions physiques sur lesquelles nous travaillions.

Dans le but de ne pas perdre le fil de nos réflexions, Victorine s'est employée à tenir à jour un document de suivi du projet<sup>1</sup>. Ce fut réellement utile puisque s'y référer permettait de se remettre plus vite dans le projet en début de séance ainsi que de garder une trace de ce que nous avons fait. Cela servait également à définir les objectifs pour la semaine suivante et a facilité l'écriture du rapport par la suite. C'est principalement Ivana qui avançait ce dernier en parallèle, à l'aide du suivi.

Le programme a rapidement pris forme et nous avons pu voir les premiers résultats. L'idée d'utiliser un logiciel de synthèse d'images pour l'affichage de nos résultats est alors venue et Emma a commencé à s'intéresser à l'utilisation de POV-RAY, dans le but d'obtenir un rendu plus esthétique et ludique.

Enfin, une fois le programme bien lancé nous nous sommes rendus compte que certaines incohérences étaient dues à des données choisies au hasard. Nous avons donc dû concrétiser un peu nos idées pour nous rapprocher d'un sens physique. Une fois ce point relativement clarifié nous avons décidé de d'affiner notre étude et de chercher à savoir si notre système était cohérent avec le modèle du gaz parfait.

---

<sup>1</sup> Voir annexe 5-

## 3. Travail réalisé et résultats

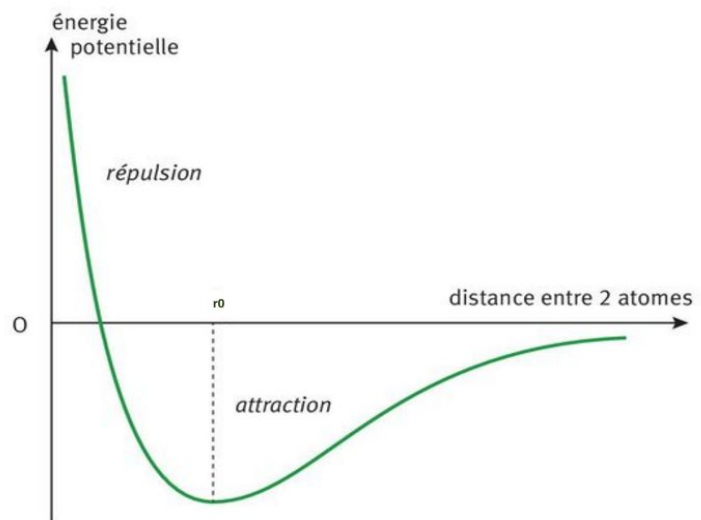
### I. Travail réalisé

Afin d'étudier la dynamique des molécules, nous avons créé un programme en utilisant Scilab, l'objectif étant de calculer et visualiser les interactions de N particules en prenant en compte l'action de chacune d'entre elles sur toutes les autres, au sein d'un domaine fermé. Nous souhaitons ensuite vérifier la cohérence de notre modèle avec celui des gaz parfaits.

#### 1- Adaptation de modèle du gaz parfait en 2D

➔ Tout d'abord, pour illustrer ce problème, nous avons dû définir une loi d'interaction à l'aide d'un modèle de courbe de potentiel. En effet, le nuage d'électrons autour d'une molécule et les forces de répulsion électrique que ce dernier implique vont avoir pour conséquences que deux molécules ne se comportent pas de la même façon selon la distance qui les séparent. La courbe ci contre présente l'allure que nous

souhaitions obtenir, nous y distinguons les domaines où les particules se repoussent puis s'attirent avant finalement de s'ignorer. Nous avons ainsi choisi une fonction mathématique de la forme :  $\frac{A}{r^2} - \frac{B}{r}$



Il a donc été obligatoire de définir dès le début des paramètres A et B que nous avons fixés arbitrairement afin d'obtenir l'allure souhaitée pour commencer. Mais nous avons rapidement compris qu'il était nécessaire d'affiner ce choix afin de palier aux incohérences obtenues. C'est pourquoi nous avons choisi de définir ces paramètres à partir de l'énergie potentielle minimum ( $E_{pmin}$ ) et du rayon  $r_0$ .

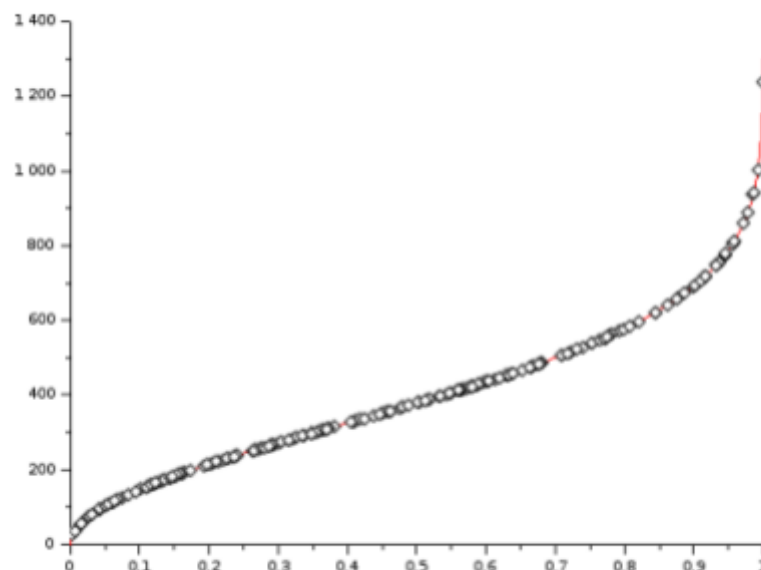
$$A = \frac{-1}{2} \times E_{pmin} \times r_0^2 \quad \text{et} \quad B = -1 \times E_{pmin} \times r_0$$

Cette loi d'interaction permet ainsi de calculer les forces d'interactions entre deux particules.



➔ En outre, nous avons commencé par générer des vitesses de façon aléatoire donc sans réel sens physique. Aussi, toujours dans le but de faire progresser notre programme vers une signification plus réaliste, nous avons utilisé la formule de Maxwell-Boltzmann qui permet de déterminer le nombre probable de particules par unité de volume dont la norme de la vitesse est comprise entre  $v$  et  $v+dv$ , étudiée au cours du semestre. Toutefois il s'est avéré nécessaire de commencer par adapter la loi que nous avons à notre problème en deux dimensions<sup>2</sup>. Nous avons utilisé cette formule pour trouver la probabilité qu'une particule ait une vitesse particulière comprise entre 0 et un  $v_{max}$  fixé arbitrairement à 1300 m/s. Une fois les normes vitesses des différentes particules obtenues, nous leur avons donné une direction aléatoire. De même, la position de départ des particules est elle aussi aléatoire.

➔ *ci contre, les vitesses en fonction de leur probabilité.*



➔ De plus, afin d'étudier ce problème nous avons choisi de distinguer deux cas. Nous aurions pu changer le nombre de particules ou encore la taille du domaine pour simuler un cas liquide et un cas gazeux mais nous avons préféré utiliser un facteur multiplicatif  $R$  qui nous permet d'adapter l'éloignement entre les particules et ainsi de modifier la densité. Plus il est faible, plus les particules sont en interaction et plus il est grand, plus elles sont indépendantes.

Dans le premier cas,  $R$  est plutôt faible, donc on se place dans des conditions d'étude relativement éloignées du modèle des gaz parfaits, puisque l'on va modéliser des particules interagissant rapidement entre elles. C'est donc un cas qui se rapproche plus d'un comportement liquide.

<sup>2</sup> Voir annexe 4-a) pour démonstration

Puis dans le deuxième cas, R est élevé, l'étude est alors plutôt représentative d'un gaz. En effet un R élevé signifie que les particules doivent être proches pour interagir donc cela revient à les éloigner, en quelque sorte. De plus, il est important de préciser que la loi de Maxwell Boltzmann que nous utilisons pour le modèle du gaz parfait repose sur certaines hypothèses comme l'isotropie ou encore l'indépendance des molécules. C'est donc pour coller à ce modèle qu'il nous faut, par exemple des distances importantes entre les molécules et c'est pour cette raison que nous utilisons un facteur R important, cela permet de se placer dans le cas s'approchant du modèle du gaz parfait.

Une fois que nous nous sommes placés dans un cas où dans l'autre, nous pouvons faire varier différents paramètres comme la température. Il nous faut ensuite calculer la pression pour vérifier si notre modélisation est cohérente avec le modèle du gaz parfait.

La loi des gaz parfaits<sup>3</sup>  $P = Nk_B T$  dépend de P et de T. Or la température repose sur la notion d'énergie cinétique<sup>4</sup>. De plus, la pression linéique P est la contribution des forces d'agitation moléculaire agissant sur une surface élémentaire. En effet, ces forces ne sont autre que la résultante de la somme des variations de quantité de mouvement par rapport au temps et à la taille du domaine.<sup>5</sup>

## 2- Développement du programme sur Scilab

Nous avons principalement utilisé Scilab pour résoudre les équations différentielles que nous avons obtenu dans le cadre de notre problème. Nous avons, en effet, un couplage d'équations différentielles car nous évaluons une nouvelle fois à chaque pas de temps les forces présentes pour chaque point du domaine. Ces forces sont alors incorporées à un tableau dans lequel nous pouvons lire chaque interaction d'une particule avec une autre, c'est un balayage de toutes les forces en présence. Par la suite, en se basant sur ce tableau nous allons utiliser le Principe Fondamental de la Dynamique (PFD). En effet, nous connaissons désormais la valeur de la somme des forces extérieures s'appliquant sur une

particule donnée or nous savons que  $\sum f_{ext} = m a = \frac{dv}{dt} m$  (PFD) donc si nous divisons par la masse, nous obtenons l'accélération pour ce pas de temps donc la nouvelle vitesse en nous basant sur celle du pas de temps précédent.

En outre, le programme exigeait que nous définissions d'entrée de jeu un certain nombre de données. Parmi ces conditions initiales à fixer, il y a les vitesses auxquelles nous avons donné un sens plus physique grâce à la loi cumulative, ce qui nous a permis de générer une courbe de distribution bien plus sensée. Mais nous avons également dû fixer un rayon de répulsion électrostatique  $r_0$  avec un ordre de grandeur cohérent ou encore une masse molaire réelle (pour laquelle nous avons choisi le méthane). Il était également obligatoire de définir une énergie potentielle minimale afin de pouvoir calculer les forces d'interaction. De même, la génération des vitesses nécessitait que nous fixions une vitesse

<sup>3</sup> Voir annexe 4-c) pour démonstration

<sup>4</sup> Voir annexe 4-b) pour démonstration

<sup>5</sup> Voir annexe 4-c) pour démonstration

maximale.

Par la suite nous nous sommes retrouvés confrontés à des difficultés concernant la gestion de l'espace. En réalité, nous avons commencé par définir arbitrairement la taille du domaine. Il s'est avéré indispensable de le réévaluer au moyen de la formule  $L = d \times \sqrt{Nbp}$  qui se base sur la taille qu'aurait le domaine si toutes les particules étaient à égale distance  $d$  des bords ainsi que des autres. De plus, on ajoute un facteur  $R = \frac{E_{p,min}}{E_{c,ini}}$  et on exprime la distance moyenne entre les particules avec  $d = R \times r_0$ , où  $r_0$  est la distance interparticulaire de répulsion. On fixe ce domaine de telle sorte dans le but de voir les particules basculer entre répulsion et attraction. De même, les particules sortaient du domaine au bout de quelques instants car les forces qu'elles subissaient les éloignaient. Nous avons donc choisi de fermer le domaine en mettant en place un système de rebond sur les parois. Ces interactions élastiques conservent la vitesse de la particule au moment où elle impacte la paroi. L'angle selon lequel elle repart est égal à l'angle d'incidence. Nous avons aussi changé le signe du vecteur vitesse. Ainsi une boucle évalue à chaque pas de temps si une molécule s'apprête à sortir du domaine et si c'est le cas, la fait rebondir à l'intérieur.

Dans le but d'obtenir des résultats à exploiter, nous pouvons, une fois le facteur multiplicatif fixé, choisir de faire varier la température mais également un des paramètres de départ comme le nombre de particules en présence. En outre lors du calcul nous évaluons à chaque pas de temps l'énergie mécanique, potentielle et cinétique (donc la température, à l'aide de la formule  $T = E_c/kb$ <sup>6</sup>) afin d'estimer la fiabilité de notre modèle. En effet l'énergie mécanique est par exemple sensée rester constante. Cela nous permet également de vérifier qu'il n'y a pas de grosses variations de température, élément de preuve du fait que notre modèle reste physique. En effet, lorsque l'on observe seulement les interactions particulières dans un gaz au repos, la température n'augmente ou ne baisse pas.

Le détail du programme est présenté dans la première annexe.

### 3- Rendu sur Povray

Dans le but d'obtenir des résultats plus visuels et plus ludiques, nous avons choisi d'utiliser le logiciel Pov-Ray, qui permet de créer des images de synthèse à partir de coordonnées. Nous avons donc choisi de montrer les interactions de 100 particules, représentées par des sphères. Pour cela, on utilise le programme réalisé sur Scilab qui nous permet d'exporter les positions des particules en fonction du temps, sous la forme  $(x_1^1, y_1^1, x_2^1, y_2^1, \dots, x_{100}^1, y_{100}^1, x_1^2, y_1^2, x_2^2, y_2^2, \dots, x_{100}^2, y_{100}^2, \dots, x_1^n, y_1^n, \dots, x_{100}^n, y_{100}^n)$  où, dans la forme  $p_a^b$ ,  $p$  est la position en  $x$  ou en  $y$ ,  $a$  le numéro de la molécule et  $b$  le numéro de l'image.<sup>7</sup>

Tout d'abord, le programme exportait à chaque pas de temps les positions des molécules, cependant comme celui-ci n'est pas constant, car il s'adapte à chaque fois à la

<sup>6</sup> Voir annexe 4-b) pour démonstration

<sup>7</sup> Voir exemple de fichier de coordonnées en annexe 3-b)

vitesse maximale des particules, les données récupérées ne permettaient pas d'avoir une représentation claire et continue des positions des différentes molécules. Nous avons alors utilisé un modulo<sup>8</sup> pour que les positions soient exportées à un pas de temps régulier et ainsi avoir une représentation cohérente du mouvement de nos molécules. De plus on normalise l'échelle du domaine d'observation afin que nos résultats soient plus visuels et plus parlants.

Ainsi nous avons écrit un programme sur Pov-Ray<sup>9</sup> afin d'exploiter ces coordonnées. Il a pour but de représenter les 100 molécules et l'évolution de leurs positions à chaque nouvelle image. On pourra ensuite, grâce aux nombreuses images obtenues, créer une vidéo en affichant plus de 25 images par seconde dans le but d'observer le comportement des particules entre elles et lors de leur contact avec les parois. Ici aussi, on étudie deux cas: un premier où  $R=2.5$  (faible) et un second où  $R=1000$  (élevé)

---

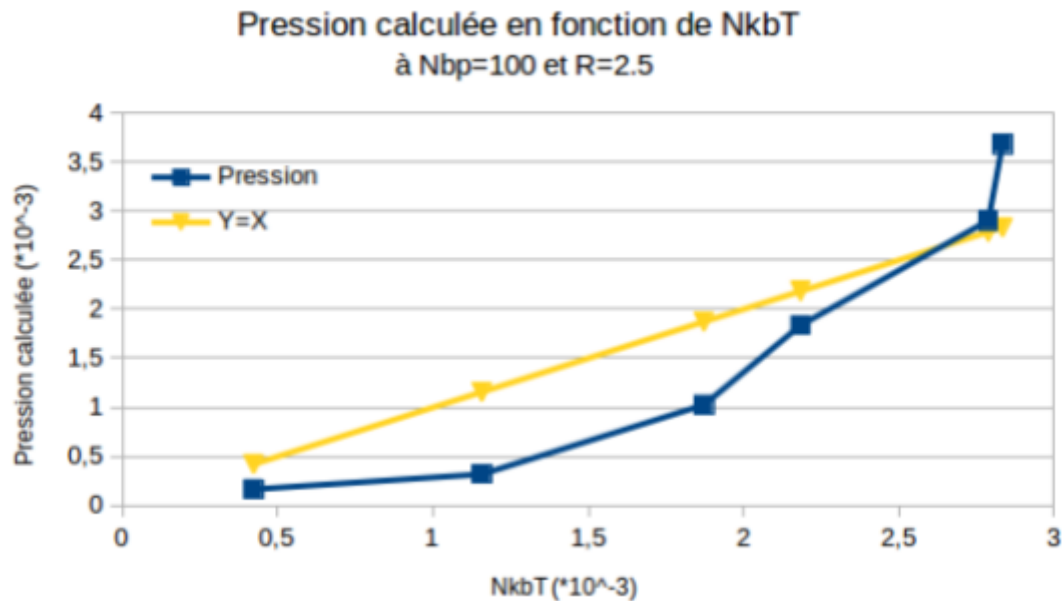
<sup>8</sup> Voir fin du programme scilab en annexe 1-

<sup>9</sup> Voir programme complet en annexe 3-a)

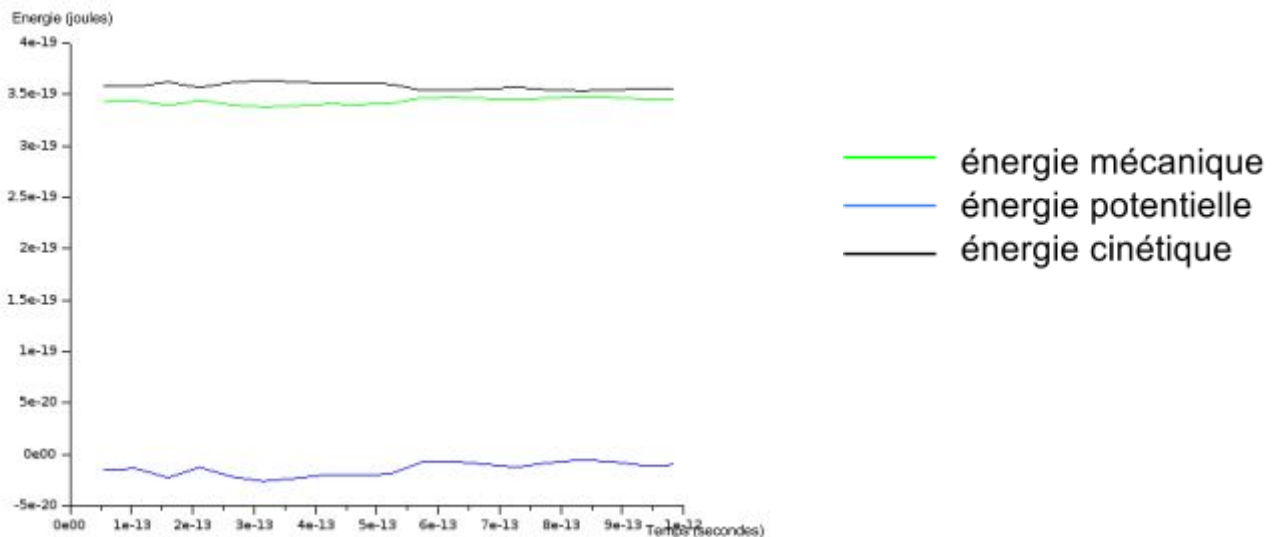
## II. Résultats

Le but de notre modélisation est donc d'abord de représenter les interactions entre les molécules d'un gaz. Nous avons par la suite choisi de nous concentrer sur une idée plus physique en cherchant à savoir, à travers deux cas différents, si notre simulation respectait un comportement que l'on retrouve dans le modèle du gaz parfait.

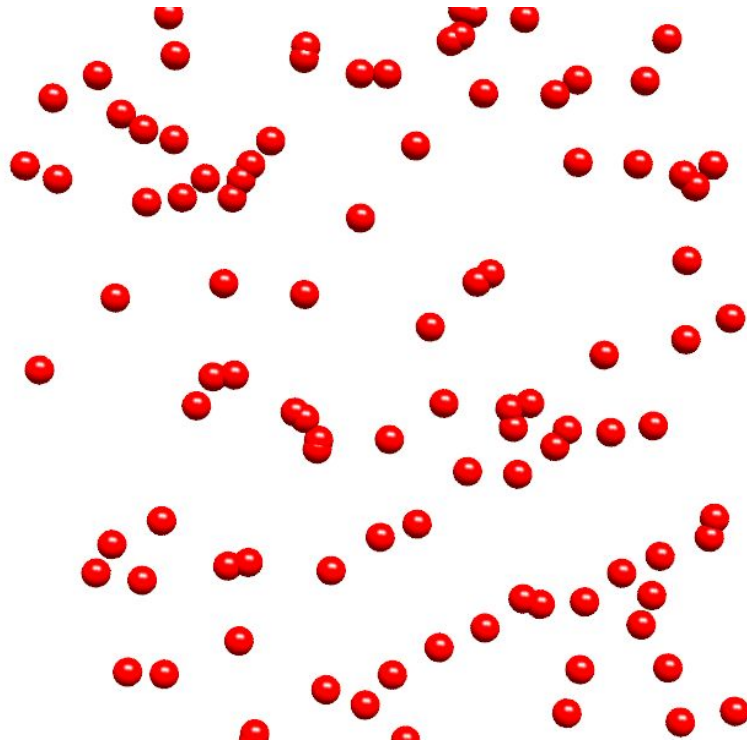
### 1- Premier cas, R petit



Dans ce premier cas, R est faible, on doit donc observer des interactions entre les particules. Il est alors cohérent d'observer une variation non linéaire de la pression. On relève les pressions et la température recalculée sur Scilab puis on trace P en fonction de  $Nk_bT_{\text{recalculé}}$ . Les deux courbes ne se superposent pas, ainsi on peut en déduire que ce cas ne s'apparente pas au modèle des gaz parfait, ce qui était prévisible.

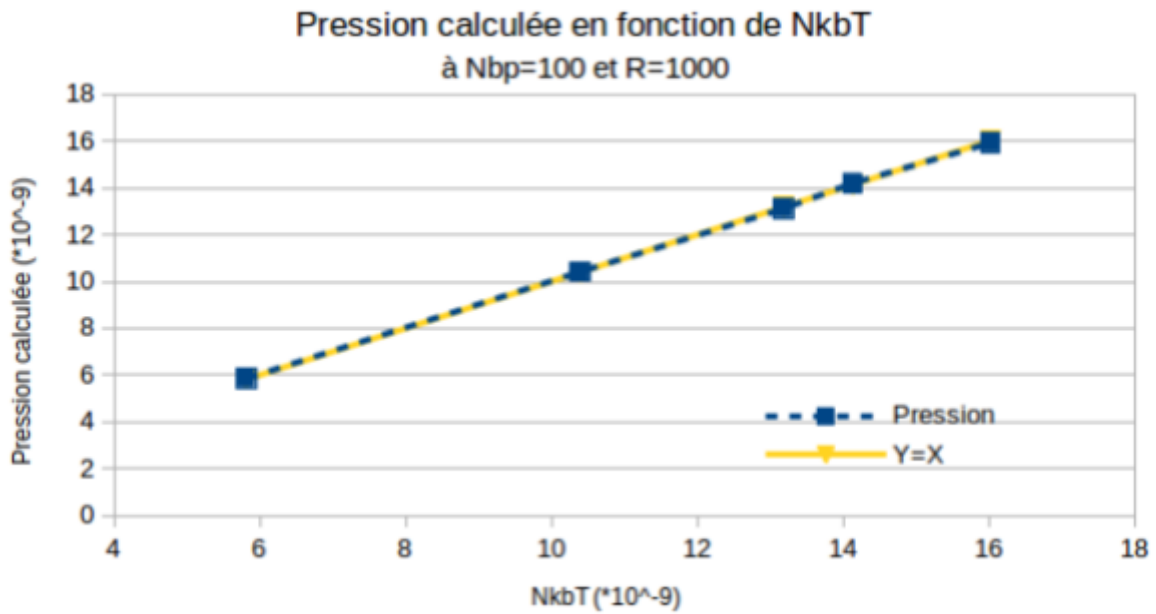


De plus, malgré le fait qu'on observe des petites variations, les trois énergies descriptives du domaine restent plutôt constantes. On peut penser que ces variations sont dues aux pertes d'énergie lors des interactions entre les particules.

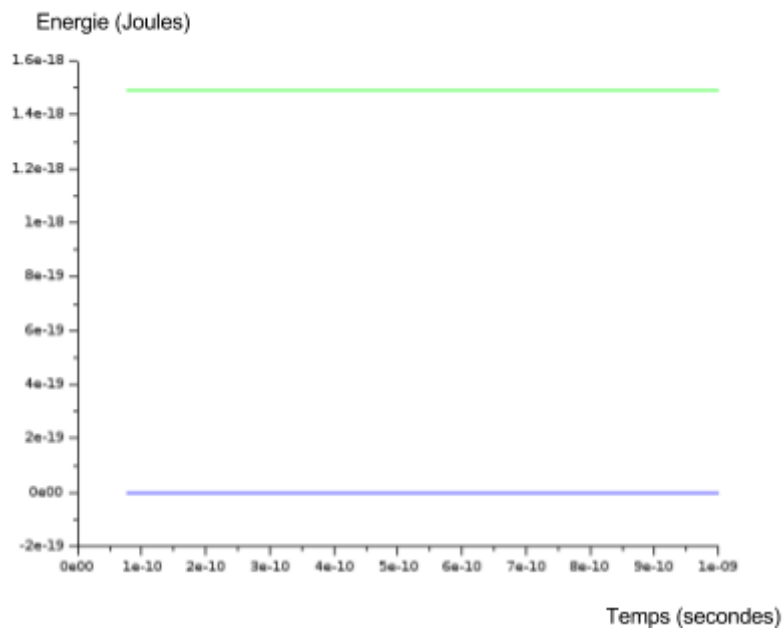


Le résultat observé sur Pov-Ray est également cohérent avec ce que nous souhaitons observer. En effet, la vidéo réalisée nous montre les phases attractives et répulsives des molécules entre elles, en plus des rebonds sur les parois. On observe alors parfaitement le fait qu'elles sont en interaction.

2- Deuxième cas, R grand



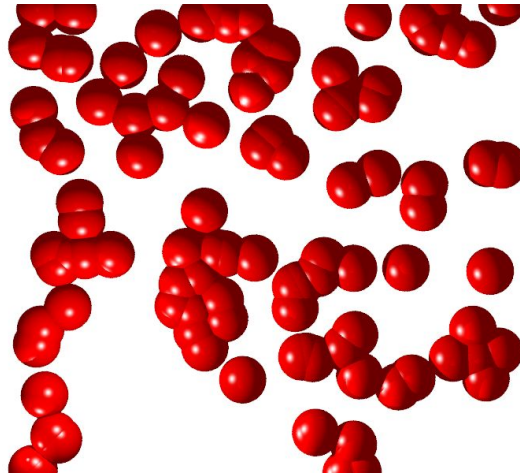
Dans ce cas, la courbe de la pression calculée en fonction de NkBT coïncide avec la bissectrice  $y=x$ . On peut donc en déduire que lors de notre modélisation, on retrouve le modèle théorique définissant les gaz parfaits, c'est à dire la loi  $P = NkBT$ .



- énergie mécanique
- énergie potentielle

*Remarque : L'énergie potentielle est trop petite donc on ne voit pas la courbe de l'énergie cinétique, qui est confondue avec celle de l'énergie mécanique.*

On remarque sur les courbes précédentes que les énergies sont constantes, ce qui est cohérent avec le fait qu'il n'y a pas d'interactions entre les particules. De plus le rebond sur le paroi est considéré comme élastique. On peut donc en déduire qu'il est normal qu'il n'y ait pas de pertes d'énergies.



La représentation obtenue sur Pov-Ray est également témoin du fait que les molécules n'interagissent pas entre elles. En effet on peut observer sur la vidéo qu'elles se croisent sans se repousser, et que les seuls changements de trajectoire qu'elles subissent sont dus au rebonds sur la paroi.

### 3- Comparaison et conclusion

La comparaison des deux cas montre des différences flagrantes. En effet, nous souhaitons retrouver la loi des gaz parfaits et avons pour cela différencié un cas ne correspondant pas à un modèle gazeux et un cas y correspondant.

Or nous pouvons facilement voir que le cas défavorable ne fonctionne pas, en effet le résultat n'est pas linéaire et ne se rapproche pas de la courbe théorique, cela se remarque également sur povray où les particules interagissent fortement. En revanche dans le cas favorable les particules s'ignorent et la courbe obtenue en faisant varier la température est très fortement similaire à la bissectrice.

Toutefois le facteur R multiplicatif choisi est plutôt très élevé, il s'agit donc un cas relativement extrême choisi dans le but de se rapprocher des hypothèses de la loi de Maxwell-Boltzmann.

Pour conclure, l'objectif était ici d'étudier un système à N corps interagissant entre eux. Les interactions de molécules dans un fluide s'inscrivent parfaitement dans cette définition et la modélisation effectuée est plutôt parlante et ludique visuellement. Dans le but constant de se rapprocher d'un modèle le plus concret possible nous avons dû redéfinir plusieurs paramètres comme la distribution des vitesses, le tout dans le but de nous rapprocher d'une loi statistique : la loi des gaz parfaits. Or avec 100 particules et un temps de calcul relativement faible nous obtenons tout de même un résultat plus que satisfaisant par rapport à la loi que nous cherchions à retrouver.



## 4. Conclusions et perspectives d'évolution

Tout d'abord, nous pouvons désormais conclure que notre modélisation des interactions des molécules gazeuse est relativement exacte. En effet, lorsque nous traçons la pression que nous obtenons  $P$  en fonction de  $Nk_B T$ , elle coïncide avec la courbe théorique si nous sommes dans un cas assimilable à un gaz parfait.

Néanmoins, ce projet comporte de nombreuses perspectives d'évolution.

Pour commencer, nous avons choisi dès le départ de travailler en deux dimensions seulement, dans le but de simplifier l'analyse du problème. Toutefois, maintenant que nous avons mieux appréhendé le problème, il serait envisageable d'effectuer une simulation en trois dimensions. Cela impliquerait en premier lieu de rajouter une colonne de coordonnées en  $Z$  à nos tableaux afin de prendre en compte non plus deux composantes mais trois. Il nous faudrait également étendre notre système de rebonds sur les parois à une troisième dimension ou encore effectuer des changements plus subtils comme utiliser les formules de Maxwell Boltzmann en trois dimensions lorsque nous générons aléatoirement les vitesses. De même lorsque que nous associons les normes des vitesses qui viennent d'être générées à des angles, il nous faudrait utiliser deux angles pour tenir compte des trois dimensions et ne plus se limiter au plan. Ainsi nous pourrions étendre notre étude du problème et la rendre plus complète.

En outre, nous avons choisi d'exporter les données pour les utiliser avec povray. Cependant même si les vidéos obtenues montrent bien les interactions des particules, nous aurions pu tout d'abord ajouter une bordure pour délimiter le domaine, mais également améliorer la visualisation de l'évolution du mouvement des particules entre elles en ajoutant le rendu des sphères associées aux images précédentes grâce à un jeu de couleurs et ainsi observer la trajectoire des différentes sphères.

Par ailleurs, nous avons choisi - comme expliqué précédemment - de nous concentrer sur ce problème à  $N$  corps en particulier mais il aurait été envisageable d'étudier un autre système. Par exemple nous aurions pu nous intéresser à un système de masses et de ressorts et essayer de modéliser les interactions au sein d'un tel système. Nous aurions pu également nous intéresser aux galaxies et au mouvement entre différents corps célestes. Nous ne l'avons pas fait principalement par manque de temps et surtout par volonté d'aboutir le modèle étudié mais cela pourrait être réellement intéressant.

Une autre amélioration envisageable serait de réévaluer nos paramètres  $A$  et  $B$ . En effet nous avons rapidement eu besoin d'une courbe de potentiel donc nous avons choisi de les fixer mais il aurait été possible de chercher des courbes de potentiels existants afin de se rapprocher encore d'un modèle physique.

Pour conclure, nous pouvons dire que ce projet nous a beaucoup appris.

En premier lieu, sur le plan scientifique car nous avons pu nous intéresser concrètement aux interactions des molécules et approfondir nos connaissances dans ce domaine. Nous avons également pu réemployer notre cours de P8-1 qui aborde justement la théorie cinétique des gazs. En outre, ce projet nous a donné l'opportunité d'utiliser Scilab et donc d'apprendre à utiliser ce logiciel.

Ensuite cet exercice nous a permis de travailler en groupe, ce qui est intéressant puisque que cela exige une certaine organisation et nécessite une certaine flexibilité afin que tout le monde puisse participer à l'élaboration du projet. Les projets de ce genre sont un vrai plus car ils nous enseignent le travail d'équipe et permettent à chacun de mettre ses compétences au service d'une réalisation commune.

## **5. Bibliographie**

<https://www.drgoulu.com/2008/11/16/le-probleme-a-n-corps/#.WSvhVNwYzcs>

<http://docplayer.fr/4477460-Sequence-8-energie-matiere-et-rayonnement-sommaire.html>

-> (valide à la date du 15/06/2017)

## 6. Annexes

### 1- Programme sur Scilab

```

Model2D-29-05.scl x
1 clear
2 //Fonction qui retourne l'energie potentielle et la norme de la force causée par la particule Pi sur la particule Pj
1 function [Ep,F]=CalculEP(Pi,Pj,r0,Epmin)
2     Vectij=Posi(Pj,:)-Posi(Pi,:); // Le vecteur entre les deux particules
3     r=norm(Vectij); // Calcul de son norme
4     if r/r0>1E-3 then //On verifie si la distance entre les deux particules sur r0 est plus grande de 10^-3
5         Ep=A/r^2-B/r; //Calcul de l'energie potentielle
6         normeF=2*A/r^3-B/r^2; //calcul du norme de la force (la differentiel de l'energie potentielle)
7         F=Vectij/r*normeF //calcul de la force
8     else
9         Ep=0;
10        F=zeros(Vectij);
11    end
12 endfunction
15
16 //Loi de Maxwell-Boltzman
1 function dP=proba(v)
2     dP=s/kb/T*v*exp(-s*v^2/2/kb/T)
3 endfunction
20
21
22 //Initialisation
23 kb=1.3806E-23; // Constante de Boltzman
24 Na=6.022E23; // Nombre d'Avocadro
25 tmax=1E-9; //Durée de la modélisation
26 Masseolaire=16; //CH4 g/mol
27 vmax=1300; //Vitesse max des particules
28 Nbpetoile=400; //Nombre des points pour faire la courbe vitesse/proba
29 r0=10E-10; //rayon de répulsion electrostatique
30 Epmin=-5E-23; //Ecini/100
31
32
33 Nbp=100; //Nombre de particules
34 T=2000; //Temperature du domaine
35 R=1000 //Facteur de Ecini/Ep
36
37
38 s=Masseolaire*1E-3/Na; // Calcul de la masse
39 vetoile=linspace(0,vmax,Nbpetoile)'; // Vecteur ligne de Nbpetoile coordonnées de distances égales entre 0 et vmax
40 for i=1:Nbpetoile
41     Pr(i)=intg(0,vetoile(i),proba) // Integration de la fonction proba de 0 à vetoile pour tout les vetoiles
42 end
43
44 Pt=rand(Nbp,1)
45 d = splin(Pr,vetoile); //Analyse splin de la courbe de proba
46 NormvitesseTab = interp(Pt,Pr,vetoile,d) //Interpolation du d pour trouver la vitesse
47
48 // Calcul de la direction de la vitesse en sachant le norme et un angle aléatoire
49 thetaTab=2*pi*rand(Nbp,1);
50 Vitesse(:,1)= NormvitesseTab.*cos(thetaTab);
51 Vitesse(:,2)= NormvitesseTab.*sin(thetaTab);
52
53 //Calcul des facteurs A et B
54 A=-1/2*Epmin*r0^2;
55 B=-1*Epmin*r0;
56
57 t=0; // Temps initialisé à 0
58 L=sqrt(Nbp)*r0*R; //Largeur du domaine carré
59 Posi=rand(Nbp,2)*L; //Tableau de position initiales des particules (1ère colonne les x et 2ème les y);
60 Vmax=max(NormvitesseTab); //Recalcul de la vitesse max
61 deltatini=(L/Vmax)/1000;

```

```

Model2D-29-05.sci ✕
47
48 // Calcul de la direction de la vitesse en sachant le norme et un angle aléatoire
49 thetaTab=2*pi*rand(Nbp,1);
50 Vitesse(:,1)= NormvitesseTab.*cos(thetaTab);
51 Vitesse(:,2)= NormvitesseTab.*sin(thetaTab);
52
53 //Calcul des facteurs A et B
54 A=-1/2*Epmin*r0^2;
55 B=-1*Epmin*r0;
56
57 t=0; ..... // Temps initialisé à 0
58 L=sqrt(Nbp)*r0*R; ..... // Largeur du domaine carré
59 Posi=rand(Nbp,2)*L; ..... // Tableau de position initiales des particules (1ère colonne les x et 2ème les y)
60 Vmax=max(NormvitesseTab); ..... // Recalcul de la vitesse max
61 deltatini=(L/Vmax)/1000;
62 deltat=deltatini;
63 Ec=[]; ..... // Tableau des énergies cinétiques à chaque affichage
64 Ep=[]; ..... // Tableau des énergies potentielles à chaque affichage
65 Em=[]; ..... // Tableau des énergies mécaniques à chaque affichage
66 Temps=[]; ..... // Tableau des temps de chaque affichage
67 EcMoy=0;
68 EpMoy=0;
69 cpt=0;
70 cpt2=0;
71 Ppart=0; ..... // Tableau des variations qqte mouvement
72 P=0; ..... // La pression
73 Tver=[]; ..... // Tableau des températures recalculées
74 EcM=0; ..... // Initialisation de l'énergie cinétique moyenne
75 //f=fopen('ExportPosi.txt','w')
76 clf
77 while t<tmax
78     TabForce=zeros(Nbp,2); ..... // Tableau des forces initialisé à 0
79     TabEp=zeros(Nbp,1) ..... // Tableau des Ep initialisé à 0
80     cpt=cpt+1;
81     for i=1:Nbp ..... // Boucle pour chaque particule
82         for j=1:Nbp ..... // Boucle pour l'interaction entre particule i et tous les autres
83             [Eptmp,F]=CalculEP(i,j,r0,Epmin);
84             TabForce(j,:)=TabForce(j,:)+F; ..... // Calcul de la force exercée sur la particule i
85             TabEp(j)=TabEp(j)+Eptmp; ..... // Calcul de Ep pour chaque particule i
86         end
87     end
88     Vitesse=Vitesse+TabForce/n*deltat; ..... // Calcul la nouvelle vitesse
89     Posi=Posi+Vitesse*deltat; ..... // Calcul la nouvelle position
90
91     Selectx=(Posi(:,1)>L); ..... // Verification de x>L
92     Posi(Selectx,1)=2*L-Posi(Selectx,1);
93     Ppart=Ppart+sum(2*n*Vitesse(Selectx,1));
94     Vitesse(Selectx,1)=-Vitesse(Selectx,1);
95
96     Selectx=(Posi(:,1)<0); ..... // Verification de x<0
97     Posi(Selectx,1)=-Posi(Selectx,1);
98     Vitesse(Selectx,1)=-Vitesse(Selectx,1);
99     Ppart=Ppart+sum(2*n*Vitesse(Selectx,1));
100
101     Selecty=(Posi(:,2)>L); ..... // Verification de y>L
102     Posi(Selecty,2)=2*L-Posi(Selecty,2);
103     Ppart=Ppart+sum(2*n*Vitesse(Selecty,2));
104     Vitesse(Selecty,2)=-Vitesse(Selecty,2);
105
106     Selecty=(Posi(:,2)<0); ..... // Verification de y<0
107     Posi(Selecty,2)=-Posi(Selecty,2);
108     Vitesse(Selecty,2)=-Vitesse(Selecty,2);

```

```
*Model2D-29-05.sci
87 ... end
88 ... Vitesse=Vitesse+TabForce/n*deltat; //Calcul la nouvelle vitesse
89 ... Posi=Posi+Vitesse*deltat; //Calcul la nouvelle position
90
91 ... Selectx=(Posi(:,1)>L); //Verification de x>L
92 ... Posi(Selectx,1)=2*L-Posi(Selectx,1);
93 ... Ppart=Ppart+sum(2*n*Vitesse(Selectx,1));
94 ... Vitesse(Selectx,1)=-Vitesse(Selectx,1);
95
96 ... Selectx=(Posi(:,1)<0); //Verification de x<0
97 ... Posi(Selectx,1)=-Posi(Selectx,1);
98 ... Vitesse(Selectx,1)=-Vitesse(Selectx,1);
99 ... Ppart=Ppart+sum(2*n*Vitesse(Selectx,1));
100
101 ... Selecty=(Posi(:,2)>L); //Verification de y>L
102 ... Posi(Selecty,2)=2*L-Posi(Selecty,2);
103 ... Ppart=Ppart+sum(2*n*Vitesse(Selecty,2));
104 ... Vitesse(Selecty,2)=-Vitesse(Selecty,2);
105
106 ... Selecty=(Posi(:,2)<0); //Verification de y<0
107 ... Posi(Selecty,2)=-Posi(Selecty,2);
108 ... Vitesse(Selecty,2)=-Vitesse(Selecty,2);
109 ... Ppart=Ppart+sum(2*n*Vitesse(Selecty,2));
110
111 ... NormvitesseTab=sqrt(Vitesse(:,1).^2+Vitesse(:,2).^2);
112 ... Vmax=max(NormvitesseTab); //Recalcul de la vitesse max
113 ... deltat=(L/Vmax)/1000; //Avancement temporel
114 ... t=t+deltat; //Calcul du temps
115
116
117
118 ... if modulo(cnpt,10)==0 then
119 ...     cnpt2=cnpt2+1;
120 ...     clf
121 ...     plot2d(Posi(:,1)/r0,Posi(:,2)/r0,-5,rect=[0,0,L/r0,L/r0]); //affichage du domaine
122 ...     //Remplissage des tableaux des stockage de données
123 ...     Ec(cnpt2)=sum(0.5*n*(NormvitesseTab.^2));
124 ...     Ep(cnpt2)=sum(TabEp);
125 ...     Em(cnpt2)=Ec(cnpt2)+Ep(cnpt2);
126 ...     Temps(cnpt2)=t;
127 //verification de la valeur de la temperature
128 ...     EcM=mean(0.5*n*(NormvitesseTab.^2)); //moyenne des valeurs de mv^2
129 ...     Tver(cnpt2)=EcM/kb;
130 ...     //Export des positions
131 ...     // -if memmodulo<modulo(t,tmax/500)
132 ...     // .....for i=1:Nbp
133 ...     // .....mfprintf(f1,'%f, %f, ',Posi(i,1)/r0,Posi(i,2)/r0);
134 ...     // .....end
135 ...     // .....end
136 ...     // - memmodulo=modulo(t,tmax/500);
137 ... end
138 end
139 P=Ppart/4/L/tmax; // Calcul de la pression
140 Trecalcule=mean(Tver); // Temperature moyenne du domaine
141 clf
142 EcMoy=mean(Ec); //Energie cinétique moyenne du domaine pour le temps tmax
143 EpMoy=mean(Ep); //Energie potentielle moyenne du domaine pour le temps tmax
144 plot2d(Temps,[Ec Ep Em],[1 2 3]); //affichage des courbes des trois énergies descriptives du domaine
145 //mclose(f1);
146
147
```

Ligne 117, Colonne 4.

## 2- Explication du programme

Au début du programme nous déclarons deux fonctions. La première, *CalculEP*, nous permet de calculer l'énergie potentielle des particules et la norme de la force exercée par la particule  $P_i$  sur  $P_j$ . La deuxième, appelée *proba*, est une application de Maxwell-Boltzmann en 2D.

Dans un second temps, on pose les constantes nécessaires à une simulation. On calcule la masse puis grâce à des fonctions telles que *Linspace*, on peut composer un vecteur ligne *vétoile*, de  $N_{bp}$  coordonnées entre 0 et  $v_{max}$ , à des distances égales. La fonction de probabilité est intégrée entre 0 et chaque valeur de *vétoile* afin de trouver la probabilité que la particule ait une vitesse que l'on nomme  $Pr$ . Enfin on réalise une analyse splin de la

courbe de la fonction proba.

On calcule ensuite la vitesse de chaque particule, que l'on va stocker dans NormevitesseTab, en réalisant une interpolation par spline de la fonction proba. On détermine alors l'orientation des vecteurs vitesses en faisant NormevitesseTab\*cos(thetaTab) pour les valeurs en x et avec sin pour celles en y. Puis on trouve A et B utilisés dans la fonction qui représente l'énergie potentielle. On détermine la largeur du domaine d'étude, en fonction du nombre de particules, r0 et R, précédemment initialisés.

On utilise ensuite un boucle while, tant que t<tmax. On initialise deux tableaux: un pour les forces et un pour l'énergie potentielle, que l'on complète pour chaque particule en utilisant la fonction CalculEP. On recalcule les nouvelles vitesses et positions pour les particules, et on vérifie qu'elles ne sont pas sorties du domaine d'étude grâce a la fonction Select. Si jamais des particules sont sorties, on modélise un rebond sur la paroi en supposant qu'il n'y a pas de pertes liées aux collisions. En même temps, on calcule la quantité de mouvement pour les particules ayant rebondi.

On détermine ensuite la vitesse maximale dans le but de déterminer un nouveau pas de temps ( $\Delta T = (L/V_{max})/1000$ ), que l'on ajoute au temps t. L'énergie cinétique, potentielles, mécaniques et le temps t sont relevés tous les 10 calculs. De plus, on crée une autre fenêtre dans laquelle on peut voir l'affichage des particules au cours du temps, pendant qu'on recalcule la température du domaine avec la formule  $T = E_c(\text{moyenne})/k_b$

Une fois la boucle while terminée, le programme nous permet de calculer la pression en fonction de la variation de quantité de mouvement, du périmètre du domaine et du temps maximal. On trace ensuite les valeurs des énergies récupérées (Ec, Ep et Em)

### 3- Programme sur Pov-Ray

#### a) Programme

Ce programme est composé de deux sous-programmes. P6.pov sert à afficher les sphères correspondantes à l'image "frame\_number". Or frame\_number est piloté par le fichier Brownian\_anim.ini qui est un script qui gère l'animation.

Ainsi, il faut mettre à jour les deux fichiers car l'exécution de P6.pov seul ne donne rien, frame\_number reste nul. On règle les paramètres afin qu'ils soient cohérents entre les deux sous programmes (N le nombre de molécules, NbImages et Final\_Frame qui indiquent le nombre d'images souhaitées). Enfin on fait tourner Brownian\_anim.ini dans le but d'obtenir nos images.

```

Messages  Brownian_anim.ini  P6.pov
+W800 +H600;
Antialias=On
Antialias_Threshold=0.3
Antialias_Depth=2
Output_File_Type=S

Input_File_Name=P6.pov
Initial_Frame=1
Final_Frame=1000
Initial_Clock=0
Final_Clock=1

Cyclic_Animation=off
Pause_when_Done=off

```

```

Messages  Brownian_anim.ini  P6.pov
#declare D=600;           //Diametre des particules

#declare Camera_0 = camera {angle 25
                           location <Largeur/2 , Largeur/2 , -3*Largeur> //On déclare et on place une caméra
                           right   x*image_width/image_height           //orientée de façon a ce que l'on
                           look_at <Largeur/2 ,Largeur/2 , 0>}         //puisse observer le domaine d'étude

camera{Camera_0}
light_source{< 1500,2500,-2500> color white}
sky_sphere { pigment { color rgb <1.0,1.0,1.0> } // end of pigment
             } //end of skysphere
light_source { <0, 500, -500> color red 1 green 1 blue 1} //On déclare également une source de lumière

#fopen infos "ExportPosi.txt" read //on ouvre et on lit un fichier
#declare Count=1; //déclaration d'un compteur
#declare i=1;
#declare Sp_x=1;
#declare Sp_y=1;

#while (Count <= NbImages) //Lecture associée à chaque image
#for (i,1,NbImages) //Lecture associée à chaque sphère
#read (infos,Sp_x) //Coordonnée x de la sphère
#read (infos,Sp_y) //Coordonnée y de la sphère
#if (Count=Frame_number) //On n'affiche que les sphères de l'image courante "frame_number",
// pilotée par le fichier "Brownian_anim.ini"
sphere { <Sp_x,Sp_y,0>,D //Ajout de la sphère
        texture {
            pigment {color rgb <1, 0, 0>} //Texture des sphères
            finish {ambient rgbft <0.15, 0.15, 0.15, 0.15, 0.15>
                    diffuse 0.85
                    phong 1
            }
        }
}
#end
#declare Count=Count+1;
#end
#fclose infos //on ferme le fichier

```

## b) Fichier texte

181.189044, 9022.993251, 4288.566020, 4416.313506, 3158.354012, 8333.292726, 3683.683624, 7237.883324, 1465.51777, 5633.511781, 2444.413341, 5522.950142, 8765.267771, 767.674731, 4866.817247, 5369.360367, 3848.570967, 6173.625308, 4812.302244, 9964.380639, 9853.150709, 469.751975, 4474.540744, 9250.756305, 2333.805642, 5.488360, 3498.839608, 9585.780346, 7060.231581, 151.504613, 3595.440999, 2967.735401, 4048.351536, 4686.18194240, 8924.341790, 9521.074638, 3495.735371, 4142.213697, 4746.057907, 4657.215446, 7703.587124, 2218.88374.620767, 396.925398, 9010.649558, 4278.765213, 4429.976356, 3160.595934, 8335.160168, 3686.096183, 7245.2801, 9472.299040, 5643.771510, 2447.879117, 5506.664606, 8774.909476, 765.056238, 4852.020758, 5386.564523, 2401.825605, 6175.655654, 4796.823190, 9964.752444, 9862.281026, 472.058765, 4484.178934, 9260.416426, 220822, 2750.449031, 3505.660964, 9588.078668, 7053.165493, 146.646328, 3591.681275, 2971.119091, 4044.391878727, 6411.757142, 8923.446585, 9506.263104, 3480.196107, 4135.944072, 4746.497264, 4652.891199, 7707.7124930891, 9373.037673, 412.661740, 8998.305876, 4268.964411, 4443.639195, 3162.837854, 8337.027607, 3688.5084011.775664, 9489.746289, 5654.031238, 2451.344895, 5490.379077, 8784.551171, 762.437735, 4837.224277, 5420.681254, 2405.649542, 6177.685996, 4781.344146, 9965.124247, 9871.411335, 474.365559, 4493.817121, 9270.65, 5739.472180, 2735.409713, 3512.482315, 9590.376986, 7046.099410, 141.788052, 3587.921558, 2974.502782, 49816.824929, 6419.320041, 8922.551378, 9491.451586, 3464.656960, 4129.674473, 4746.936618, 4648.566954, 771, 8431.537866, 9371.454585, 428.398061, 8985.962211, 4259.163620, 4457.302018, 3165.079770, 8338.895042, 3627302, 4001.998541, 9507.193515, 5664.290960, 2454.810672, 5474.093560, 8794.192851, 759.819222, 4822.42786, 9108.941045, 2409.473469, 6179.716335, 4765.865118, 9965.496048, 9880.541631, 476.672355, 4503.455303, 9236150, 5739.743539, 2720.370415, 3519.303657, 9592.675297, 7039.033334, 136.929786, 3584.161848, 2977.8864745, 9826.271106, 6426.882933, 8921.656167, 9476.640094, 3449.117900, 4123.404901, 4747.375969, 4644.242714, 5072, 8433.144838, 9369.871503, 444.134349, 8973.618574, 4249.362847, 4470.964813, 3167.321683, 8340.762472145.806291, 3992.221440, 9524.640703, 5674.550667, 2458.276447, 5457.808071, 8803.834505, 757.200703, 4807.28103, 9097.200876, 2413.297385, 6181.746668, 4750.386121, 9965.867846, 9889.671905, 478.979154, 4513.093463570.383821, 5740.014897, 2705.331161, 3526.124979, 9594.973599, 7031.967279, 132.071540, 3580.402154, 29818.197484, 9835.717243, 6434.445804, 8920.760956, 9461.828651, 3433.578932, 4117.135362, 4747.815317, 4639.6908.597242, 8434.751807, 9368.288430, 459.870546, 8961.275008, 4239.562128, 4484.627529, 3169.563582, 83425184, 3156.485217, 3982.444398, 9542.087789, 5684.810322, 2461.742207, 5441.522670, 8813.476100, 754.582186, 8638.569613, 9085.460788, 2417.121276, 6183.776989, 4734.907211, 9966.239641, 9898.802125, 481.285947, 48189797, 3566.631544, 5740.286252, 2690.292095, 3532.946215, 9597.271869, 7024.901312, 127.213361, 3576.6421662, 7660.984918, 9845.163248, 6442.008579, 8919.865752, 9447.017402, 3418.040196, 4110.865915, 4748.25465908349, 6894.889592, 8436.358755, 9366.705381, 475.606093, 8948.931953, 4229.761811, 4498.289684, 3171.8054050.241806, 3167.163702, 3972.667762, 9559.534153, 5695.069563, 2465.207831, 5425.237934, 8823.117291, 7806.329121, 8648.910697, 9073.721195, 2420.945009, 6185.807224, 4719.428937, 9966.611418, 9907.931967, 483.83, 5679.809494, 3562.879382, 5740.557600, 2675.253467, 3539.767251, 9599.570068, 7017.835551, 122.355329, 2194.449985, 7653.772565, 9854.608963, 6449.571134, 8918.970572, 9432.206594, 3402.501945, 4104.596661, 494, 5437.596512, 6881.182352, 8437.965657, 9365.122382, 491.342203, 8936.588457, 4219.961141, 4511.952328, 7.943550, 4055.518614, 3177.842569, 3962.890779, 9576.981140, 5705.329179, 2468.673586, 5408.952609, 8832.7



#### 4- Démonstration des formules utilisées en 2D

##### a) Maxwell-Boltzmann

$$F_o(v) = f(V_x)f(V_y)$$

$$F_o(v) = \left(\frac{\alpha}{\pi}\right)^{\frac{1}{2}} \times e^{(-\alpha V^2 x)} \times \left(\frac{\alpha}{\pi}\right)^{\frac{1}{2}} \times e^{(-\alpha V^2 y)}$$

$$F_o(v) = \left(\frac{\alpha}{\pi}\right) e^{(-\alpha v^2)}$$

$$\langle dN \rangle = NF_o(v)d^2v = N \left(\frac{\alpha}{\pi}\right) e^{(-\alpha v^2)} d^2v \text{ or } d^2v = dv_x dv_y = v dv d\theta$$

$$\text{Donc } d^2N = N \left(\frac{\alpha}{\pi}\right) e^{(-\alpha v^2)} v d\theta dv$$

Intégration en  $\theta$  entre 0 et  $2\pi$

$$dN = N2\pi \left(\frac{\alpha}{\pi}\right) e^{(-\alpha v^2)} v dv$$

b) Energie cinétique,  $\alpha$  et formule de la température

$$\overline{Ec} = \int_0^{\infty} \frac{1}{2} m v^2 dN(\vec{v})$$

$$\overline{Ec} = mN \propto \int_0^{\infty} v^3 e^{(-\alpha v^2)} dv$$

$$\overline{Ec} = \frac{Nm \alpha}{2 \alpha^2}$$

$$\overline{Ec} = \frac{Nm}{2 \alpha}$$

Or par définition de la température,  $\overline{Ec} = \frac{d^o}{2} N k_b T$  (Théorème de l'équipartition)

Donc  $\boxed{\overline{Ec} = \frac{Nm}{2\alpha} = \frac{d^o}{2} N k_b T}$  où  $d^o = 2$  car nous nous plaçons dans un repère 2D lors de notre étude

On en déduit donc  $\boxed{\alpha = \frac{m}{2k_b T}}$

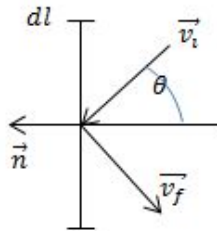
Ainsi  $dN(v) = \frac{2Nm}{2k_b T} \times v e^{\frac{-mv^2}{2k_b T}} dv = \frac{Nm}{k_b T} \times v e^{\frac{-mv^2}{2k_b T}} dv$  et  $\boxed{dP = \frac{dN(v)}{N}}$

Numériquement, on évalue  $\overline{Ec}^{num}$

$$\boxed{T = \frac{\overline{Ec}}{Nk_b} = \frac{\overline{Ec}^{num}}{k_b}} \text{ car } \overline{Ec}^{num} = \frac{\overline{Ec}}{N}$$

### c) Loi des gaz parfaits

Calcul de la pression linéique



$$\Delta_{part} = m\vec{v}_f - m\vec{v}_i = -2mv \cos\theta \vec{n} \text{ avec } n = -\vec{v}_x \text{ ou } \vec{v}_x \text{ ou } -\vec{v}_y \text{ ou } \vec{v}_y \text{ suivant le bord étudié.}$$

Nombre de particules impactant dl pendant dt

$$d^4 Np = d^2 S \times d^2 N(\theta) = v dt dl \cos\theta \times N \left(\frac{\alpha}{\pi}\right) e^{(-\alpha v^2)} v d\theta dv$$

$$d^4 \vec{p}(\theta, v) = \Delta_{part} \times d^4 Np = -2mv \cos\theta \vec{n} \times v dt dl \cos\theta \times N \left(\frac{\alpha}{\pi}\right) e^{(-\alpha v^2)} v d\theta dv$$

$$\left\{ \begin{array}{l} \Delta \vec{P}_{total} = \int_{\theta=0}^{\pi/2} \int_{v=0}^{\infty} d^4 \vec{p}(\theta, v) = -2mN. dt dl. \vec{n} \int_0^{\infty} v^3 e^{(-\alpha v^2)} dv \times \int_{-\pi/2}^{\pi/2} \cos^2 \theta d\theta \\ \Delta \vec{P}_{total} = -2mN. dt dl. \vec{n} \left(\frac{\alpha}{\pi}\right) \times I_3 \times \left[ \frac{\sin 2x}{4} + \frac{x}{2} \right]_{-\pi/2}^{\pi/2} \\ \Delta \vec{P}_{total} = -2mN. dt dl. \vec{n} \left(\frac{\alpha}{\pi}\right) \times \frac{1}{2} \frac{1}{\alpha^2} \times \frac{\pi}{2} \\ \Delta \vec{P}_{total} = -\frac{Nm}{2\alpha} dt dl \vec{n} \\ \Delta \vec{P}_{total} = -\frac{Nm. 2k_b T}{2m} dt dl \vec{n} \\ \boxed{\Delta \vec{P}_{total} = -Nk_b T dt dl \vec{n}} \end{array} \right.$$

$$\text{Et } \boxed{P = -\frac{\Delta \vec{P}_{total} \vec{n}}{dt dl} = Nk_b T} \text{ (Principe d'action réaction)}$$

## 5- Suivi des séances

### **SÉANCE 1 :**

*Contenu de la séance :*

- Spécification du problème qui est encore très général (collision entre deux galaxies, molécules, ressorts...)
- Idée de modélisation du problème: interactions entre des molécules, étude de ces interactions, du potentiel: composantes répulsives ou attractives selon des conditions initiales (température, rayon, masse, vitesse initiale...)
- On cherche à aboutir à une modélisation numérique qui modélise notre problème pour 2, 3, 100, 500 molécules
- Application à la chimie nucléaire ?
- Résolution équation Brownienne et modélisation numérique

*Objectifs pour la suivante :*

- Retrouver et recomprendre interactions vues en P2
- Mécanique des fluides, potentiel d'une molec
- Se renseigner sur équations Browniennes

### **SÉANCE 2 :**

*Contenu de la séance :*

- Écriture d'un programme sur Scilab qui modélise les interactions de N molécules en 2D sur un temps donné à partir de courbes de potentiel définies arbitrairement pour le moment

*Objectifs pour la suivante :*

- Finir de commenter le programme
- Commencer le rapport
- Se renseigner sur les logiciels pour synthétiser nos résultats (Povray)
- Amélioration du programme afin de gérer les sorties des molécules du domaine

### **SÉANCE 3 :**

*Contenu de la séance :*

- amélioration du programme : prise en compte de la fin du domaine : création d'un « rebond » sur la paroi

*Objectifs pour la suivante :*

- Se renseigner sur les logiciels pour synthétiser nos résultats (Povray)
- programme :trouver comment gérer le delta T, essayer de gérer les « bloquages »

### **SÉANCE 4 :**

*Contenu de la séance :*

- insertion d'images écho et prise en compte de leur influence (on a gardé les rebonds sinon disparaissent)

*Objectifs pour la suivante :*

- trouver comment génère vitesses aléatoires a partir gaussienne / MB
- évaluer distance avec A/B
- programme : insérer estimation de l'énergie cinétique
- reprendre démo p8 pour MB

**SÉANCE 5 :**

*Contenu de la séance :*

→ élaboration de MB en 2d, redéfinition du alpha pour l'avoir en fonction de T (en passant par écriture Ec moyenne)

*Objectifs pour la suivante :*

→ intégrer programme modélisant le tirage aléatoire de la vitesse dans gros programme (et commenter)  
→ chercher valeurs physiques (masse molaire, pression température, A, B)

**SÉANCE 6 :**

*Contenu de la séance :*

→ Finalisation du programme générant des vitesse ayant un sens physique (proba)  
→ Réflexion autour du sens de A et B  
→ Première visualisation d'exemples Povray

*Objectifs pour la suivante :*

→ intégrer ce programme (proba) au code de base (model2D)  
→ vérifier la conservation de l'énergie en vérifiant que la température ne varie pas  
→ Avoir une première idée du code Povray à écrire

**SÉANCE 7 :**

*Contenu de la séance :*

→ définition plus précise du but du projet : se pencher sur le modèle du GP en simulant différents états  
→ intégrer ce programme (proba) au code de base (model2D)

*Objectifs pour la suivante :*

→ revenir à un programme ne prenant en compte que les rebonds (plus de simulation autour) MAIS avec des conditions initiales assez cohérentes pour ne pas avoir de soucis de particules « bloquées »  
→ définir un A et un B réaliste par rapport à la taille du domaine donc un R0 réaliste  
→ début de rapport

**SÉANCE 8 :**

*Contenu de la séance :*

→ travail sur programme: n retrait « autres boites » deltat T désormais adaptatif, A et B supprimés pour les remplacer par leur expression dépendant de r0 et Epm (cf feuille calcul)

*Objectifs pour la suivante :*

→ remettre effets de symétries et essayer de mieux comprendre les

**SÉANCE 9 :**

*Contenu de la séance :*

→ modification du programme : normalisation de l'échelle, réflexion sur les différentes énergies, pb : forces trop grandes..

*Objectifs pour la suivante :*

→ envoyer au prof scan page « A&B » et lui envoyer le programme

→emma regarde povray (afficher image à partir des fichier de points sorti de scilab)

## **SÉANCE 10 :**

*Contenu de la séance :*

→ programme repris par le prof fonctionne désormais mais comment calculer P?

*Objectifs pour la suivante :*

→ rapport

## **SÉANCE 11 :**

*Contenu de la séance :*

→ création compteur de collisions + création d'un tableau compilant les variation de qqté de mvmt, somme des ces variations divisée par 4L et par le tmax (si besoin que d'1 valeur de pression alors ok car on a nb de collision et Tmax pour calcul, si besoin de plusieurs valeurs alors il faudra insérer un nb de collision tel qu'on calcul la pression à chaque fois qu'on l'atteint)

=> on a une pression (avec ordre de grandeur cohérent !! vérifie avec la relation  $p = nkbt$ )

→ avancement povray

*Objectifs pour la suivante :*

→ rapport

→ tracer P en fonction de T et sort Kb => doit être retrouvé et change paramètre jusqu'à plus avoir bon kb/plus droite, trace aussi en fct de N ( $Nb/l^2$ )

### **Objectifs pour la suite du projet:**

#### ↘ **programme**

pression = force exercée par particules sur parois : il nous faut compter les collisions puis au bout d'un certain nombre voir combien de temps ça a prit et diviser la somme des variation de qqté de mouvement (expression avec cos, pour évaluer cos il faut regarder sur quelle paroi on est et prendre la composante correspondante) par ça et par la surface (4L) pour obtenir la pression totale

le facteur multiplicatif ligne 54 permet de gérer le 'rayon d'action' des particules les unes par rapport aux autres, il faut en fixer un puis faire plusieurs simulations en changeant la densité ou la T et normalement pour des grands facteurs (ex 1000) on devrait valider  $p = NKbT$  donc trouver qqch de linéaire

→ évaluer P dans le but de tracer le graphique de P en fonction de T (sachant que T dépend de l'énergie cinétique donc des vitesses initiales) sinon possibilité de jouer sur r

→ d'après ce graph, regarder quand on a un comportement linéaire en accord avec le modèle du GP et à partir de quand ce n'est plus le cas => normalement correspond au moment où a plus un gaz mais un liquide

#### ↘ **modélisation**

attention pour l'export de fichier il ne faut pas envoyer des données tous les 20 calculs par ex mais plutôt tous les delta T demandé par povray