

3年生学生実験

流体の統計力学とシミュレーション

理論化学研究室 松本 正和

2010年1月18日月曜日

気体の運動論: 剛体球モデルによる計算機実験	3
目的	3
剛体球気体とは	3
2次元気体と3次元気体	4
シミュレーションにとりかかる前に：考察	4
分子の速度	4
平均自由行程	7
圧力	8
状態方程式	9
物性の変化	10
シミュレーションの方法	11
シミュレーションの初期条件	12
プログラムの流れ	13
NodeBoxによるデモ	14
コマンドラインでの実行	14
コマンドラインパラメータ	16
作業と解析	16
 気液平衡の統計力学	 19
目的	19
シミュレーションの方法	19
運動方程式の数値解法	20
相互作用	20
圧力	20
温度について	21
シミュレーションの初期条件	21
プログラムの流れ	22
NodeBoxによるデモ	22
コマンドラインでの実行	23
コマンドラインパラメータ	24
作業と解析	24
TIPS	26
gnuplot	26
スクリプト	26

1. 実験の進め方について

1. 出席はとりません。各自でスケジュールを決めて作業して下さい。逆に、用もないのに実験室に居残って、ゲームをしたり騒いだりしないように。
2. 実験室は飲食喫煙禁止です。

2. レポートの書き方について

1. 演習形式なので、レポートには実験手順や、生のデータを書く必要はありません。課題の解答、データ解析の結果と、自分でプログラムを作成した場合はそのプログラム(と説明など)を提出して下さい。
2. レポートに図や表を入れる場合は、必ず解説と解釈を付けて下さい。解説のないデータは採点できません。
3. 提出されたレポートを見ると、Excelなどを使って、誤ったフィッティングを行っているケースが多数あります(あらかじめわかっている定数まで回帰で求めている、物理的にありえない関数でフィットしている、など)。ご注意ください。
4. レポートの締切は、各グループの四日目の実験日の7日後(1/22, 1/28, 2/3)です。実験室で松本に直接提出するか、高等研究院2階218号室に出して下さい。

3. レジューメの記号

■課題■ この記号は、作業内容とは関係ない設問をあらわしています。事前に解いておくことも可能です。

○作業○ この記号は、作業すべき内容を表しています。

♪解析♪ この記号は、作業で得られたデータをもとにした解析であることを表しています。事前にひととおり目を通して、作業する時に、どんなデータを記録しておく必要があるかを考えておいて下さい。

★発展課題★この記号は、比較的高度な問題を表しています。時間に余裕があれば取り組んで下さい。

気体の運動論: 剛体球モデルによる計算機実験

目的

単純な流体の熱力学的な性質は、統計力学に基いたさまざまな理論、および計算機シミュレーション(分子シミュレーション)で再現することができる。

統計力学の考え方を学ぶための簡単なモデルとして、ここでは2次元剛体球(円板)気体を使って、気体のさまざまな性質を考察し、実際にシミュレーション結果と照合していく。

剛体球気体とは

一般の気体分子の相互作用は、図1(a)のようなポテンシャル関数(Lennard-Jonesポテンシャル関数)でうまく近似できることが知られている。電氣的に中性な、球形分子の場合には、液体状態まで含めて、このポテンシャル関数を使って、様々な物理量をかなり上手く再現することができる。

Lennard-Jones(LJ)ポテンシャルは分子の中心間の距離の関数として表される。

$$v(r) = 4\epsilon \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 \right]$$

ポテンシャル関数の勾配が、分子間力を表す。分子が接近するにつれ、じょじょに引力が働くが、関数形を見ればわかるように、ポテンシャル関数は距離の6乗に反比例するので、遠距離ではほとんど力を感じない。分子が非常に接近する(分子間距離 $< \sigma$)と、分子間に強い斥力が働く。

これを、さらに簡略化した、次の図1(b)のようなポテンシャル関数を考える。このポテンシャルでは、Lennard-Jones相互作用よりも斥力が強く、逆に引力はない。ポテンシャル関数の勾配は、距離1以上で0、距離1で $-\infty$ となる。すなわち、分子は距離1よりもはなれている間は全く互いに力をおよぼさず、距離1になった時に撃力をうけ、完全弾性衝突する。このような相互作用をする分子を剛体球分子と呼ぶ。

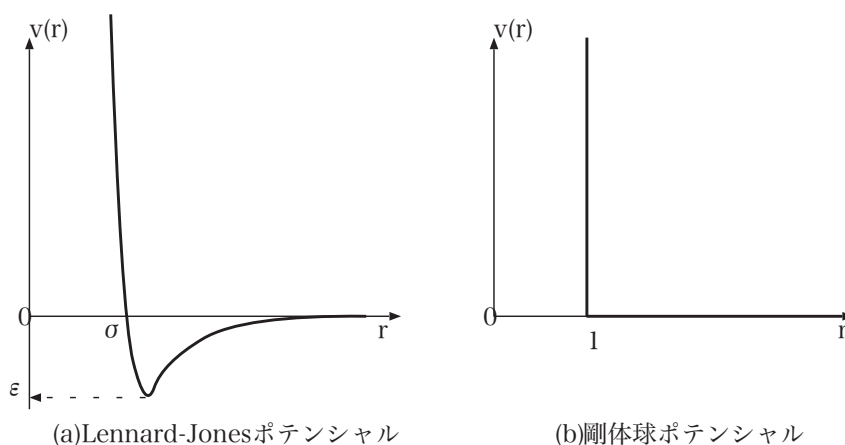


図1 分子間ポテンシャルの関数形

剛体球分子は現実には存在しないし、現実の分子の非常に粗い近似であるが、意外にも実在分子の性質をうまく再現し、計算が容易で、解析的に取り扱い易いので、古くから統計力学の研究対象とされてきた。

2次元気体と3次元気体

現実の気体分子は3次元空間を運動しているが、ここではモデルを単純化し、気体運動論の諸式の意味をよく理解するために、まず2次元モデルを扱うことにする。

2次元の剛体球気体の運動は、ちょうど摩擦がないビリヤード台上のボールの運動に例えられる。はじめにある分子だけにエネルギーを与えると、球同士がお互いに衝突をくりかえすうちに、運動エネルギーはほかの球に伝えられていく。ビリヤードでは、球と台の間に大きな摩擦があるため、球の回転まで考えないとうまく球の軌跡を制御できない。そこがビリヤードの面白さなのだが、ここでは単純に、台との摩擦は考えず、球同士は完全弾性衝突するものと考えよう。現実のビリヤードとは異なり、台との摩擦によってエネルギーを失わないので、球は運動を続け、いずれ運動エネルギーはすべての球に分配される。

シミュレーションにとりかかる前に：考察

実際にシミュレーションにとりかかる前に、気体運動論にもとづいて、2次元剛体球気体の様々な性質を推定する。

2次元気体の統計的性質の多くは、すでに学んできた3次元気体の統計的性質の導出方法を参考にすれば、容易に知ることができる。

3次元の場合の、気体の様々な性質を計算する手順を示すので、どの式が3次元気体特有であるかに十分注意を払ってほしい。

分子の速度

気体分子の、x軸方向の速度の分布はMaxwellの分布式より得られる。x軸方向に速度 u_x の分子の確率は

$$F(u_x)du_x = \left(\frac{M}{2\pi kT}\right)^{1/2} \exp\left(-\frac{Mu_x^2}{2kT}\right) du_x$$

となる。¹

3次元の場合には、各方向(x,y,z)への並進運動は独立だから、3次元での速度分布は

$$F(u_x)F(u_y)F(u_z)du_x du_y du_z = \left(\frac{M}{2\pi kT}\right)^{3/2} \exp\left(-\frac{M(u_x^2 + u_y^2 + u_z^2)}{2kT}\right) du_x du_y du_z$$

である。角度方向に積分し、動径方向の分布関数に変換すると、上の速度分布式は

¹ 平尾・加藤「化学の基礎」講談社サイエンティフィク

$$F(u)du = 4\pi \left(\frac{M}{2\pi kT} \right)^{3/2} \exp \left(-\frac{Mu^2}{2kT} \right) u^2 du$$

となる。計算の過程は省略した。

これより、分子の平均速度は

$$\begin{aligned} \bar{u} &= \int_0^\infty u F(u) du \\ &= \left(\frac{8kT}{M\pi} \right)^{1/2} \end{aligned}$$

と求めることができる。また、速度の2乗の平均値は同様に

$$\begin{aligned} \overline{u^2} &= \int_0^\infty u^2 F(u) du \\ &= \frac{3kT}{M} \end{aligned}$$

となり、1分子あたりの平均運動エネルギーは

$$\frac{1}{2} M \overline{u^2} = \frac{3}{2} kT$$

となる。

2次元の場合にも同様にして速度分布を求めることができる。各方向 (x,y) への並進運動は独立だから、2次元での速度分布は

$$F(u_x)F(u_y)du_x du_y = \left(\frac{M}{2\pi kT} \right) \exp \left(-\frac{M(u_x^2 + u_y^2)}{2kT} \right) du_x du_y$$

である。ここで、積分変数 u_x, u_y を極座標を用いて以下のように書きかえる：

$$\begin{aligned} u^2 &= u_x^2 + u_y^2 \\ u_x &= u \cos \theta \\ u_y &= u \sin \theta \end{aligned}$$

すると、次のような関係が得られる：²

$$du_x du_y = u du d\theta$$

そして角度 θ 方向に積分してしまうと、速度分布は

$$\begin{aligned} F(u) &= \int_0^{2\pi} \left(\frac{M}{2\pi kT} \right) \exp\left(-\frac{Mu^2}{2kT}\right) u du d\theta \\ &= \left(\frac{M}{kT} \right) \exp\left(-\frac{Mu^2}{2kT}\right) u du \end{aligned}$$

という風に求めることができる。

■課題■ 2次元気体の平均速度 $\langle u \rangle$ を求めなさい。また、速度の2乗の平均値を求め、1分子あたりの平均運動エネルギーが $k_B T$ になることを示しなさい。

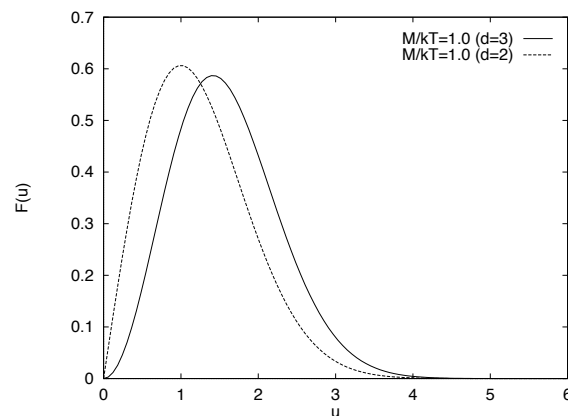


図2 2次元と3次元の速度分布 $F(u)$ のちがい

² 直交座標から極座標への変換は、Jacobianを使うと容易である。2次元の場合には、

$$dx dy = \frac{\partial(x, y)}{\partial(r, \theta)} dr d\theta$$

ただしJacobian $\frac{\partial(x, y)}{\partial(r, \theta)}$ は次のようにして求められる。

$$\begin{aligned} \frac{\partial(x, y)}{\partial(r, \theta)} &= \begin{vmatrix} \frac{\partial x}{\partial r} & \frac{\partial x}{\partial \theta} \\ \frac{\partial y}{\partial r} & \frac{\partial y}{\partial \theta} \end{vmatrix} \\ &= r \end{aligned}$$

3次元の場合のJacobianは同様に

$$\begin{aligned} \frac{\partial(x, y, z)}{\partial(r, \theta, \varphi)} &= \begin{vmatrix} \frac{\partial x}{\partial r} & \frac{\partial x}{\partial \theta} & \frac{\partial x}{\partial \varphi} \\ \frac{\partial y}{\partial r} & \frac{\partial y}{\partial \theta} & \frac{\partial y}{\partial \varphi} \\ \frac{\partial z}{\partial r} & \frac{\partial z}{\partial \theta} & \frac{\partial z}{\partial \varphi} \end{vmatrix} \\ &= r^2 \sin \theta \end{aligned}$$

である。(ランダウ、リフシッツ「統計物理学第3版(上)」岩波書店)

平均自由行程

気体状態でも、分子はひんぱんに衝突を繰り返している。この衝突によって、気体分子は運動エネルギーを交換し、温度勾配がある場合には、熱を高温側から低温側へゆるやかに伝達する。

もし分子同士がほとんど衝突しないとしたら、どうなるだろうか。極端なケースとして、分子がまったく相互作用しない場合には、それぞれの分子は、最初に与えられたエネルギーをだれにも伝えられないまま、いつまでも等速運動を続けることになり、分子の速度分布は初期分布のまま保たれ、平衡には到達しない。

では、分子はどれぐらいの距離を直進することができるのだろうか？3次元の気体の場合にも、簡単な考察でこの距離を求めることができる。

簡単のために、1分子だけが速度 $\langle u \rangle$ で運動していて、ほかの周囲の分子は止まっているものとしよう。時間 Δt の間に、分子は $\langle u \rangle \cdot \Delta t$ だけ進む。分子の進行方向に対して、図3のような円柱を考えると、もし周囲の分子の中心がこの円柱の中に含まれていれば、 Δt の間に衝突するだろう。円柱の体積は $\pi d^2 \langle u \rangle \Delta t$ であり、その中に含まれている分子の個数は $\pi d^2 \langle u \rangle \Delta t \rho$ (ただし $\rho = N/V$)だから、 Δt あたりの平均衝突回数は $\pi d^2 \langle u \rangle \Delta t \rho$ 、つまり、単位時間あたりの平均衝突回数は $\sigma \langle u \rangle \rho$ と求められる。ただし、 ρ は単位体積あたりの分子の個数(数密度)、 $\sigma = \pi d^2$ は衝突断面積と呼ばれている。

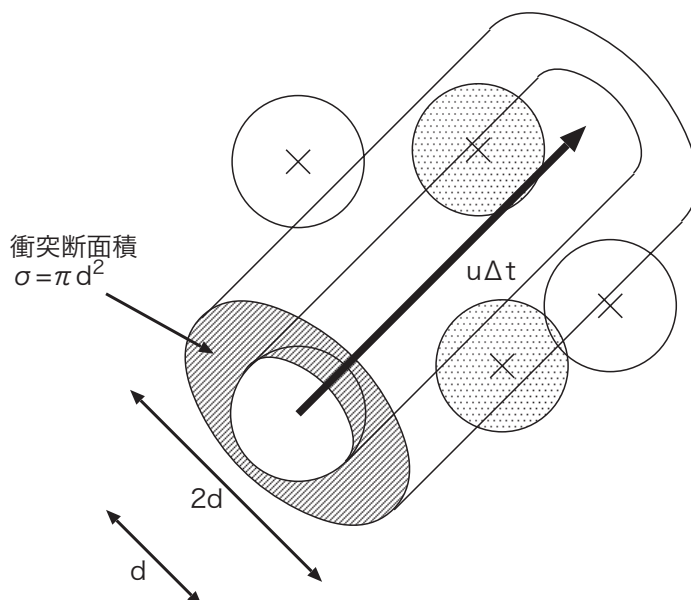


図3 ある分子と、 Δt 以内に衝突する分子。直径 $2d$ 、長さ $\langle u \rangle \cdot \Delta t$ の円筒の内部にある分子(網掛け)は衝突するものとする。

実際には、他の分子も動いているから、上の式の数値 $\langle u \rangle$ は、分子間の相対速度でおきかえなければいけない。分子はあらゆる方向に等確率で運動しているとすると、相対速度は $\sqrt{2} \cdot \langle u \rangle$ となる。

★発展課題★ 相対速度が、分子速度の $\sqrt{2}$ 倍になることを説明しなさい。

1回の衝突までに分子が進む距離は、分子の速度を単位時間あたりの平均衝突回数で割って $\langle u \rangle / (\sqrt{2} \langle u \rangle \rho) = 1/(\sqrt{2} \rho)$ と求められる。これを平均自由行程(Mean free path)と呼び、 λ で表わす。

■課題■ 2次元の場合にも、平均自由行程は同じ形の式になる。 ρ と σ が何に相当するかを考えよ。また、次元解析を行って、平均自由行程が長さの次元を持つことを確かめなさい。

圧力

容器の壁が、容器内部の分子からうける単位面積あたりの力が、気体の圧力である。気体の運動論にもとづいて、3次元での気体の圧力を求めてみよう。

気体分子が壁に衝突すると、壁は気体から $2(\text{分子の質量}) \times (\text{分子の速度の壁に垂直な成分})$ だけ力積を受け、気体分子は完全弾性衝突すると考える。

話を単純にするために、気体分子は一辺 L の立方体のなかにはいつているものとする。図4のように x 軸に垂直な壁に、 x 軸方向の速度が u_x で質量が M の分子が衝突すると、壁に $2Mu_x$ の力積を与える。片方の壁には、単位時間あたり $u_x/2L$ 回衝突するので、壁全体が1個の気体分子から単位時間に受けとる力積は $M u_x^2/L$ である。壁の面積は L^2 なので、壁が単位時間、単位面積あたり、この分子からうける力積(すなわち圧力)は $M u_x^2/L^3$ と求められる。 N 分子について和をとれば、

$$P = \sum_{i=1}^N M u_{xi}^2 / L^3 = MN \overline{u_x^2} / L^3$$

と書くことができる。 $\overline{u^2} = \overline{u_x^2} + \overline{u_y^2} + \overline{u_z^2}$ 、系は等方的だから、 $\overline{u^2} = \overline{u_x^2} \times 3$ が成り立つ。

L^3 を容器の体積 V におきかえれば、

$$\begin{aligned} P &= \frac{1}{3} \frac{NM \overline{u^2}}{V} \\ &= \frac{NkT}{V} \end{aligned}$$

という式にたどりつく。

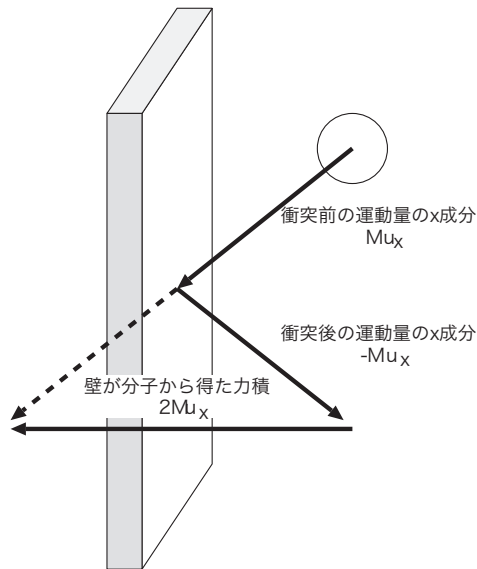


図4 壁に分子が衝突すると、運動量の法線成分の2倍の力積が壁に与えられる。

2次元の場合は、壁の面積が L 、容器の体積が L^2 、平均2乗速度が $\langle u^2 \rangle = \langle u_x^2 \rangle \times 2$ におきかわり、圧力の式は結局

$$P = \frac{1}{2} \frac{NM\overline{u^2}}{V}$$

$$= \frac{NkT}{V}$$

となる。

■課題■ 理想気体の圧力を、全運動エネルギー E_k で表わしてみよ。(簡単)

状態方程式

教科書にのっているような、気体運動論の標準的な導出方法を使うと、最終的に上の式のような理想気体の状態方程式にたどりつく。圧力や平均自由行程を求める過程で、分子はあたかもビリヤードボールのようなものとして取り扱い、その結果、理想気体の状態方程式にたどりついた。では、剛体球気体は本当に理想気体のようにふるまうのだろうか？

一方で、van der Waalsの状態方程式は現実の気体、液体の性質(排除体積効果、分子間力による凝集)をうまく取り込んで、通常の気液相図を定性的に再現する。剛体球気体の場合は、分子間力はないものの、排除体積効果が無視できないので、理想気体の状態方程式には従わないと予想できる。

なにかがおかしい・・・

上に示したさまざまな式の導出の過程で、気体分子を「理想的」にとりあつかっている個所がどこにあるのか単純に予想してみる。

分子の速度分布

Maxwell分布は、並進運動のエネルギー準位を考え、古典近似を導入して求められる。運動量だけに依存し、座標に依存しないので、一般的な気体について成り立つだろう。

平均自由行程

剛体分子の場合にも、密度が低い時は上の式はほぼ正しい。密度が高くなると、分子同士が重なることができない(排除体積効果)ために、衝突頻度は理想気体からの予測値よりも増加し、平均自由行程は短くなるだろう。排除体積を V' と書けば、

• 仮説1：平均自由行程の式の分母には密度 $\rho = N/V$ が含まれている。剛体球の場合には、実効体積は $V - V'$ しかないので、平均自由行程は $\lambda \times (V - V')/V$ となる。

• 仮説2：体積が x 倍になると、長さは $x^{1/3}$ (2次元の場合には $x^{1/2}$) になるから、平均自由行程も $\lambda \times [(V - V')/V]^{1/3}$ になる。

圧力

平均自由行程の場合と同様、排除体積効果のために、実際に分子が動くことができる体積が減ってしまう。従って分子の衝突頻度が増加し、結果として、圧力も理想気体の予測値よりも増えるだろう。van der Waals流に考えれば、 $P = Nk_B T / (V - V')$ になるだろう。

繰り返すが、以上は仮説なので、実際のシミュレーションをふまえて、各自できっちり考えてほしい。

物性の変化

温度変化

例えば、ある瞬間に、容器に含まれる剛体球気体分子の速度をすべて2倍(運動エネルギーは4倍)にしたとする。分子が衝突するまでに必要な時間は半分になるが、衝突のしかたは変化しないし、分子のたどる軌跡もまったく変化しない。この様子をビデオに撮っておいて、半分のスピードで再生すると、もともとの速度での運動と全く同じになってしまう。

つまり、剛体球気体では、温度は系の性質に影響しない。温度を変化させることは時間スケールを変化させることに等しい。

しかし、現実の分子の場合には、速度を2倍にすると、運動エネルギーが4倍、したがって絶対温度を4倍にすることに相当する。 0°C (273K)の氷はいきなり 800°C (約1300K)に熱せられる。たとえ容器の体積が同じであったとしても、 0°C の氷と 800°C の水はビデオで見れば容易にみわけられるだろう。

体積変化

一方、体積を変える(あるいは球の半径を変える)と、系の性質はしだいに変化していく。ビリヤードでつかう三角の枠(ラックと呼ぶそうです)のなかに15個の球を入れると、球は最密充填構造をとり、いくらゆすつてもその構造を壊すことはできない。このことから、密度を高くしていくと、剛体球気体は固体に転移することが予想できる。この転移はAlder転移と呼ばれている。

■課題■ 剛体球系には気相と固相があるが、液相は存在しない。その理由を考察しなさい。

★発展課題★ 体積を固定したまま温度を変えた時に圧力がどのように変化するかを考え、剛体球系の相図(あるいはPVT曲面)のおおよその形を予想しなさい。

シミュレーションの方法

上で考察してきた結果を、コンピュータシミュレーションで再現してみよう。

2年生の計算化学概論の課題ですでに何人かは力学シミュレーションに挑戦して、粒子(星、分子)の運動をシミュレートする方法について学んでいることだろう。Lennard-Jones ポテンシャルのようなスムーズな相互作用ポテンシャルで運動する気体をシミュレートする場合には、ニュートンの運動方程式を数値積分することで、ある時刻 t に気体分子が周囲から受ける力を積分して、直後の時刻 $t+\Delta t$ での粒子の位置を算出し、粒子運動の軌道をつぎつぎに求めていった。

しかし、この方法は、剛体球気体には応用できない。2つの剛体球が徐々に接近していくと、2つの球の中心間距離がちょうど球の直径 d に等しくなった瞬間にポテンシャルが不連続に変化し、しかも ∞ に発散する! (図1) 実際には、ちょうど球が接した時に2つの球は互いに撃力を受け、瞬間的に運動の方向と速度が変化してしまう。球が壁と衝突する場合でも同様である。したがって、ニュートンの運動方程式を解いた場合のように、一定の時間間隔 Δt でちびちび分子を動かしていく方法はいかぬ。

そのかわり、剛体球分子は、接触する瞬間まで直線運動する(衝突以外の相互作用がない)から、2つの球が直進していくと、いつ衝突してしまうか、そしてどちらに反射するかは簡単に計算することができる。衝突までの運動は直線だと決まっているから、時々刻々数値積分を繰り返す必要もなく、非常に精度良く分子の運動を追跡することができるのである。

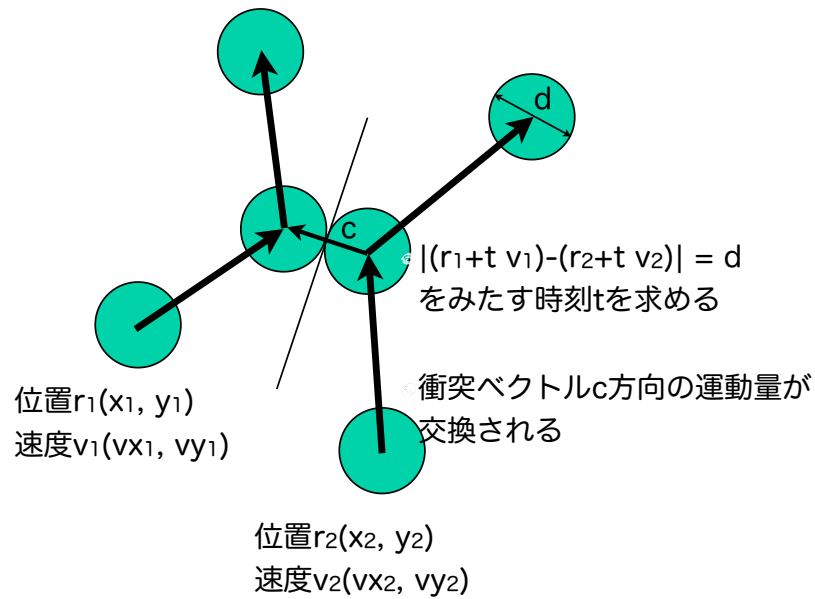
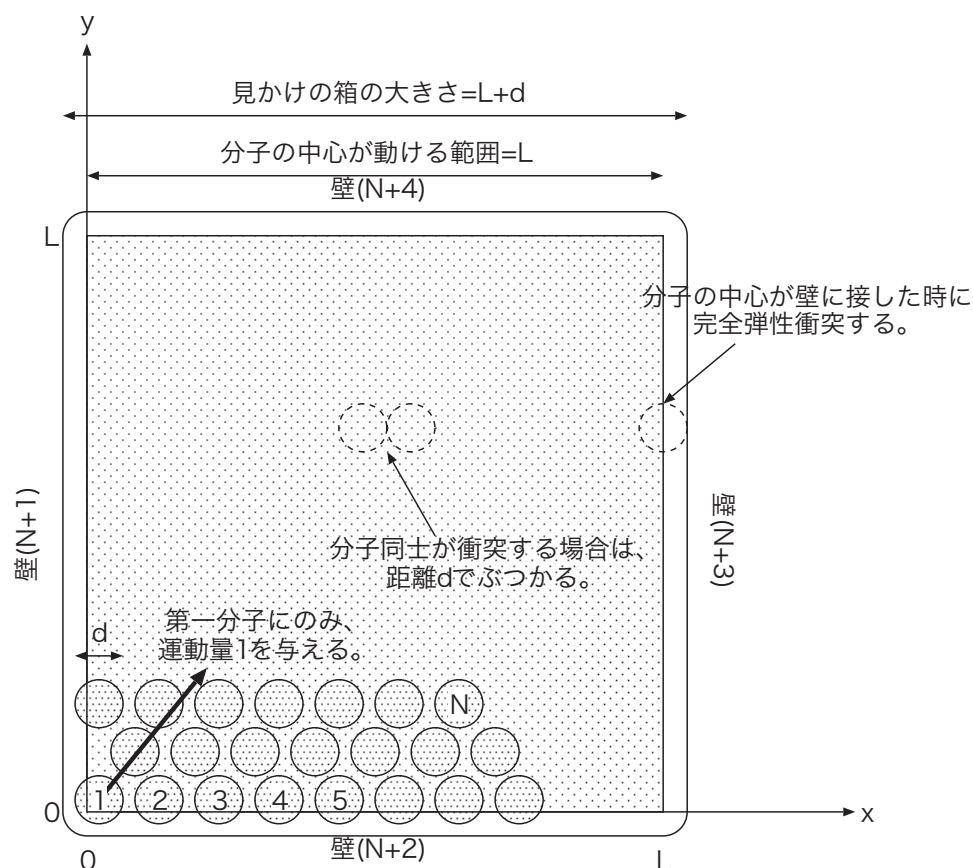


図5: 粒子衝突の図

シミュレーションの初期条件

気体分子は一辺の長さ L の正方形の堅い箱のなかに全部で N 個はいつているものとする。
簡単のため、すべての分子の質量 M は1、直径 d は1とする。

初期状態として、分子は箱の右上隅にかたまって、ほぼ最密充填(三角格子)に並んでおり、1分子だけに運動量を与え、ほかの分子の初速度は0とする。



プログラムの流れ

時間 τ を1ステップと呼び、これをさらに指定ステップ数だけ繰り返す。

■課題■ 直径1の2つの粒子の座標と速度をそれぞれ $(x_i, y_i), (v_{xi}, v_{yi})$ ($i=1,2$)とする。2つの粒子が衝突する時刻を求めよ。この計算は単純な2次方程式の求根問題である。通常、2つの根を得るが、正の根のうち小さいほうが衝突時刻になる。

NodeBoxによるデモ



NodeBoxはpython で手軽にグラフィックスを作成する開発環境である。剛体円板のシミュレーションが実際にどのように進行するかを、アニメーションで見よう。

1. 画面下のDockで、NodeBox(上のアイコン)をクリック。
2. サンプルプログラムhd2nb.pyを開き、メニュー”Python”→”Run”を選択。
3. 粒子が円で、速度の向きと大きさが線で表示されます。初速は1粒子にしか与えていませんが、最初の配置が密集配置になっているので、すぐに隣りの粒子に衝突し、エネルギーが分配され乱雑化していく様子が見えます。
4. 粒子数やセルの大きさはプログラムの以下の行を編集すれば変更できます。てきとうに変更しても構いません。

```
def setup():
    global system
    width = 10.0
    height = 10.0
    zoom = 10.0
    size(width*zoom, height*zoom)

    system=System(step=0, nballs=5, cell=[width,height],
                  gc=GC(zoom=zoom),kT=1.0)
```

コマンドラインでの実行



グラフィックス表示を見ると、何がおこっているのかわかりやすいのだが、表示にかなりの時間をとられるので、実際に統計をとる場合は、コマンドプロンプト(ターミナル)からpythonを直接起動する。

1. ターミナルを起動する。
2. hd2.pyプログラムのパラメータで、セルサイズや粒子数を指定する。例えば、サイズ10x10の2次元の箱の中に、10粒子を入れ、温度を1.0とし、1ステップの時間 τ を1.0

とし、100ステップ計算し、自由行程の統計データを記録し、結果をresult.*ファイルに記録する場合は、以下のようにパラメータを与える。(下線部を自分で入力)³

```
prompt% ./hd2.py --cell=10.0,10.0 --atoms=10 --temp=1.0 --dt=1.0  
--flight 100 result
```

3. しばらくすると、計算が終わり、result.*という名前のファイルがいくつかできる。ファイルの内容は以下の通り。

result.log

温度Tや圧縮因子zなどの熱力学量を、1ステップごとに記録する。統計熱力学量の計算に必要。第1カラムから順に、ステップ数、温度、圧縮因子z、全エネルギー、1ステップの間の粒子衝突の回数(壁との衝突は含まない)

```
0 1.0 1.77434743978 10.0 5  
1 1.0 0.0 10.0 5  
2 1.0 1.74243058245 10.0 2  
3 1.0 1.40554281492 10.0 1  
4 1.0 0.230411213108 10.0 1  
5 1.0 1.65936050429 10.0 6  
6 1.0 1.84022275375 10.0 4  
7 1.0 0.322194530971 10.0 1  
8 1.0 0.0 10.0 2  
9 1.0 3.04384538066 10.0 4
```

result.fli

すべての粒子衝突での、2つの粒子の(衝突直前の)速度と、前回の衝突からの直線飛行時間を記録する。平均自由行程の計算や、速度分布関数の作成に必要。第1カラムが速度、第2カラムが飛行時間。

```
0.556852940081 0.0636938813985  
4.43733194647 0.0636938813985  
0.308273658072 0.274903159039  
0.463737585943 0.338597040437  
0.0515410500751 0.260205996594  
0.460864479834 0.598803037031  
0.00618078732815 0.260401141592  
0.460823031804 0.859204178622  
0.000741687331627 0.260403952259  
0.460822434937 1.11960813088
```

result.hd2

継続して同じ条件でシミュレーションを続けたい場合に、hd2.pyに読みこませるデータ。解析には使用しない。

³ コマンドを入力し、改行を押す前に、行末に"&"を付けておくと、プログラムをバックグラウンドで走らせることができます。ウィンドウの数を増やさず、たくさんの計算を一度に実行するのに便利。


```
10.0 10.0 ←箱の大きさ
10 ←粒子数
5.07454097703 4.22505053688 ←粒子0の座標
0.137268170372 -0.351641303045 ←粒子0の速度

3.37728452774 0.984760325895 ←粒子1の座標
1.63634974442 -0.475144372424 ←粒子1の速度

8.62933995064 9.93664758442
0.929881247531 0.556916128154
....
```

4. 剛体円板の場合、温度を変えても時間尺度が変化するだけで、分子配置には全く影響がないので、密度(箱の大きさ)だけが調節できる。いろいろな密度で統計をとる。このデータはあとの解析で必要になる。

コマンドラインパラメータ

```
使い方: ./hd2.py [-a n|--atoms=n][-d t|--dt=t][-t x|--temp=x][-c x,y|--
cell=x,y,z][-f|--flight] steps outputbase
-a n|--atoms=n      円板の個数を指定。
-f|--flight         自由行程の生情報をoutputbase.fliに記録する。
-d t|--dt=t         1ステップの時間 $\tau$ (default=1.0).
-t x|--temp=x       温度(単位はkT).
-c x,y|--cell=x,y   箱の大きさ。3つ指定すると3次元になる。
steps              シミュレーションのステップ数。
outputbase         出力ファイルの名前(拡張子を除く)
```

作業と解析

○作業○ 様々な密度で、シミュレーションを行え。温度 kT は1.0固定でいい。粒子数 N は30個以上が望ましい。1ステップで圧力がそれなりに正しく計算できるためには、1ステップの時間 τ のあいだに粒子衝突が N 回ぐらひは必要。密度によって τ を調節し、それぞれの密度で($N \times 1000$ 回)以上衝突がおこるように、ステップ数を調節せよ。(衝突は多いほど正確な統計が得られる)。⁴

`average.py`は、`hd2.py`が生成したデータファイルの、あるカラムの値の平均をとるプログラムである。例えば、`result.log`の第3カラム(圧縮因子)の平均を計算する場合は、以下のように使用する。

⁴ ターミナルをいくつも開き、出力ファイル名が重複しないように注意すれば、異なる条件で同時に計算をすることもできます。


```
prompt% ./average.py 3 < result.log  
1.05912198666 1000  
prompt%
```

1つめの数字が平均、2つめの数字がデータ個数を表す。

`result.log`のデータは一定時間間隔に出力されているので、圧力の平均を求める場合は、圧縮因子($=pV/Nk_B T$)の単純平均を求めてから、圧力に換算すればいい。

平均自由行程、平均速度は、`result.fli`に記録された、各粒子の直進時間と速度のデータを平均して得られる。平均自由行程は(速度×飛行時間)の平均値である。平均速度を求める場合は、飛行時間を考慮した重みづけ平均をとる必要がある。

○作業○ 各密度での圧力、平均自由行程、平均速度を求めよ。シミュレーション開始直後の、平衡に達する前のデータ(`result.log`、`result.fli`の最初の100行程度?)は除外する。`average.py`を改造して、平均自由行程と平均速度を計算できるようにする。

`distrib.py`は、`hd2.py`が生成したデータファイルの、あるカラムの値の分布(ヒストグラム)を出力するプログラムである。例えば、`result.log`の第3カラム(圧縮因子)の分布は

```
prompt% ./distrib.py 3 < result.log  
0.0 4.11  
0.1 0.04  
0.2 0.07  
0.3 0.13  
0.4 0.13  
0.5 0.17  
0.6 0.18  
0.7 0.18  
0.8 0.2  
0.9 0.22  
....  
prompt%
```

1つめの数字が区間、2つめの数字が、データがその区間に含まれる確率密度を表す。

速度分布の場合は、`result.fli`に記録された、各粒子の直進時間と速度のデータを利用し、飛行時間で重みづけした分布を求める。

○作業○ データファイルの1カラムの値の分布を計算するプログラム`distrib.py`を参考に、速度分布を求めるプログラムを作れ。

minimum=2.0, binwidth=0.01, maximum=10.0の場合の例

bin[x]のx	0	1	2	3	4	5	...	(800)
計数範囲	2.0～ 2.01	2.01～ 2.02	2.02～ 2.03	2.03～ 2.04	2.04～ 2.05	2.05～ 2.06	...	(10.0)

♪解析♪ シミュレーションで得られた、圧力、平均自由行程、平均速度、速度分布を、理想気体を想定した値と比較してみよ。高密度領域でのずれに注目すること。⁵

♪解析♪ 実在気体の状態方程式は次に示すビリアル方程式で表すことができる：

$$\frac{PV}{NkT} = 1 + B_2(T)/V + B_3(T)/V^2 + \dots$$

通常の気体では、ビリアル係数 B_i は温度の関数だが、剛体球気体の場合には、前述の議論から、これらの係数は温度に依存しない。通常、剛体球気体の状態方程式は上の式の形よりも、もっぱら充填率 $\eta = N\pi d^3 / 6V$ (2次元の場合には $\eta = N\pi d^2 / 4V$)を使って、

$$\frac{PV}{NkT} = 1 + A\eta + B\eta^2 + \dots$$

と表される。⁶シミュレーションの結果から、 A (可能なら B も)を推定しなさい。状態方程式が、排除体積 V' を用いて $P(V-V') = Nk_B T$ と書けるとしたら、上式の A 、 B がそれぞれいくらになるかを考え、それらの係数をシミュレーションの結果と比較しなさい。

直径 d の剛体球の近くに別の剛体球が接近しても、距離 d よりも近付くことができない。このため、剛体球が1つ存在すると、その周辺に、半径 d の他の分子が入りこめない領域ができる。この体積を排除体積と呼ぶ。直径 d の剛体球1つの排除体積は、 $v' = 4\pi d^3 / 3$ (2次元の場合には $v' = \pi d^2$)だが、互いに排除しあう(排除体積を2重に数える)ため、 N 分子の全排除体積は $V' = N v' / 2$ となる。

⁵ 現実のシステムでは、容器の体積が非常に大きく、容器壁があることによる統計量のずれは無視できるが、今回シミュレーションを行ったような非常に小さな系では、無視できなくなってくるので、精度を上げるためには粒子数を増やすことが望ましい。しかし、一方で、粒子数がふえと、シミュレーションにかかる時間が長くなってしまう。正確な統計量を得るためには、粒子数を32、64、128、と変えていって、容器壁によるずれをみつとり、 $N \rightarrow \infty$ の極限を外挿するのが理想的である。時間があれば粒子数を増やしてみよ。

⁶ J.-P Hansen and I.R. McDonald, "Theory of Simple Liquids 2nd Edition", Academic Press.

気液平衡の統計力学

目的

単純な流体の熱力学的な性質は、統計力学に基いたさまざまな理論、および計算機シミュレーション(分子シミュレーション)で再現することができる。

前節では、単純化された剛体球モデルを用いたが、より現実的な(しかしまだ十分単純な)分子間相互作用を持つ流体の分子動力学計算を行い、得られた結果から、統計力学の式に従って、気液平衡近傍の流体の様々な熱力学量を算出することを試みる。また、得られた等温線から、気液平衡、臨界点の存在を確認する。

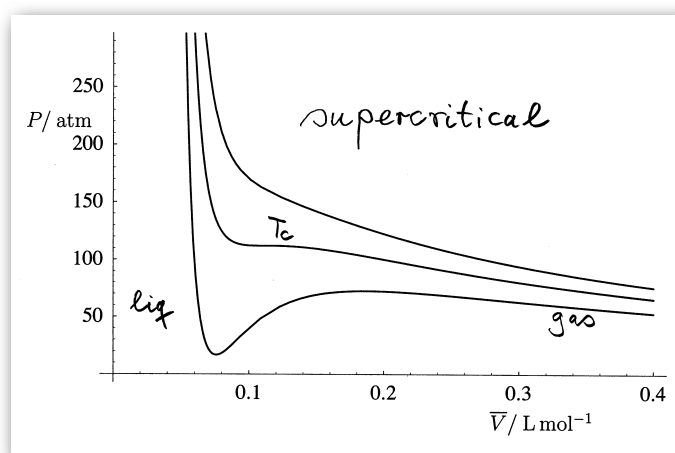


図7 典型的な等温線(体積-圧力曲線)

(<http://faculty.smu.edu/whorsthe/PC1Chap1.pdf>より)

ここでは、原子間相互作用として、Lennard-Jones相互作用を採用する。Arなどの希ガスの分子間相互作用はLennard-Jones相互作用で近似される。⁷

$$v(r) = 4\epsilon \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 \right] \quad (2.0)$$

このような単純な式で、現実のアルゴン、酸素、窒素などの比較的小さな分子の気体・液体の性質をかなり定量的に再現することができる。⁸ただし、前節の剛体球の場合と異なり、遠方の分子との相互作用をきちんと計算する必要があり、運動方程式を逐次解く必要があるため計算量が多いこと、「衝突」を明確に定義できないため、平均自由行程や排除体積といった量を厳密に算出できないことなどの短所もある。

シミュレーションの方法

今回の実験では、実在分子集団の分子動力学シミュレーションを行う。

⁷ Lennard-Jones相互作用する流体を、以後単純にLennard-Jones流体と呼ぶ。

⁸ P.W.Atkins, "Physical Chemistry" 8th edition.

運動方程式の数値解法

運動方程式は、座標の微分方程式

$$\mathbf{F} = m \mathbf{a} \quad (2.1.1)$$

で与えられる。ただし、 \mathbf{F} は物体に加わる力、 m は質量、 \mathbf{a} は加速度(位置の2階微分)。しかし、ごく単純なケースを除き、解析的に解を求めることができない。分子動力学法では、これを差分方程式とみなして数値的に解く。

まず、座標の2回微分である加速度を、速度を介して

$$\partial \mathbf{v}(t) / \partial t = \mathbf{a}(t)$$

$$\partial \mathbf{x}(t) / \partial t = \mathbf{v}(t) \quad (2.1.2)$$

の2つの一階微分方程式に分解する。さらに、微分を差分に書換えると、

$$\mathbf{v}(t+\Delta t) = \mathbf{v}(t) + \mathbf{a}(t) \Delta t$$

$$\mathbf{x}(t+\Delta t) = \mathbf{x}(t) + \mathbf{v}(t) \Delta t \quad (2.1.3)$$

と近似できる。各粒子に加わる力は、(2.0)で示したLennard-Jones相互作用関数で計算し、(2.1.1)で加速度 $\mathbf{a}(t)$ が得られる。これを(2.1.3)に代入し、現在の座標と速度から、 Δt だけ未来の座標と速度を計算する。⁹

相互作用

Lennard-Jones 流体の場合、全分子間の相互作用エネルギーは、

$$\begin{aligned} V(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \dots, \mathbf{r}_N) &= \frac{1}{2} \sum_{i=1}^N \sum_{j=1, j \neq i}^N v(|\mathbf{r}_{ij}|) \\ &= \sum_{i=1}^N \sum_{j>i}^N v(|\mathbf{r}_{ij}|) \end{aligned} \quad (2.2.1)$$

で計算できる。

Lennard-Jones相互作用関数のパラメータは、例えばArの場合には、 $\varepsilon = 119.8\text{K}$ 、 $\sigma = 0.3405\text{nm}$ である。今回のシミュレーションでは、簡単のため $\varepsilon = 1$ 、 $\sigma = 1$ とおく(換算単位系を用いる)。Lennard-Jones相互作用は2体相互作用(2つの原子の相対距離だけで定義される相互作用)である。2体相互作用のみで相互作用すると仮定することで、計算量を減らすことができる。

圧力

圧力を求める方法はいろいろ考えられるが、剛体球以外の分子では、ヴィリアル方程式から圧力を計算するのが一般的である。

⁹ 単純にこれを繰り返すと、誤差がどんどん蓄積してしまうので、それを避けるための様々なアルゴリズムが考案されている。本プログラムでは、Velocity-Verlet法を使用して誤差の蓄積を回避した。(上田顕、「コンピュータシミュレーション」、朝倉書店)

2体相互作用する分子からなる系のヴィリアル方程式は次のような式である：

$$\begin{aligned}\frac{\beta P}{\rho} &= 1 - \frac{\beta}{3N} \left\langle \sum_{i=1}^N \mathbf{r}_i \nabla_i V_N \right\rangle \\ &= 1 + \frac{\beta}{3N} \left\langle \sum_{i=1}^N \mathbf{r}_i \cdot \mathbf{f}_i \right\rangle \quad (2.3.1)\end{aligned}$$

式の、 $\frac{1}{3} \sum_{i=1}^N \mathbf{r}_i \cdot \mathbf{f}_i$ の部分は、内部ヴィリアルと呼ばれ、分子間相互作用に起因する項である。これを以後 W と書く。理想気体の場合にはこの項がなくなり、単に $\beta P/\rho=1$ となる。分子間相互作用が(2.0)式のような2分子間相互作用の和で書ける場合は、内部ヴィリアル W は、総ての分子対に関する、2分子間ヴィリアル関数 $w(r) = r \frac{dv(r)}{dr}$ の総和として次のように書くことができる。

$$W = -\frac{1}{3} \sum_i \sum_{j>i} w(r_{ij}) \quad (2.3.2)$$

シミュレーション中で圧力を計算する場合は、(2.2.1)式に基いて分子間相互作用を計算するついでに、 $w(r)$ も計算しておいて、得られた W をもとに P を求める。

温度について

外界と切りはなされた世界で、力学法則のみに従って粒子が運動する場合、全エネルギーは保存されるが、運動エネルギーは変動し、温度もゆらぐ。今回は等温線を描きたいので、目的の温度になるように、全エネルギーが調節されるよう工夫した。

シミュレーションの初期条件

実際にプログラムを作成するためには、細かな条件を定義する必要がある。

- 粒子数は一定とする(粒子の出入りはない)。容器の体積および温度 $k_B T$ はパラメータとして与える。容器の形状は立方体とする。
- 容器壁周辺では、表面の効果のために、バルクな物質とは物性が異なる場合がある。今回のシミュレーションでは、特に小さなシステムを取り扱うために、容器壁の影響をうけやすい。そこで、周期境界条件¹⁰を採用する。
- 単位はすべて原子サイズ σ 、相互作用パラメータ ϵ で換算してある(つまり、 $\sigma=1$ 、 $\epsilon=1$ とみなす)。分子の初期配置は、剛体円板のプログラムと共通である。

¹⁰ 周期境界条件：容器の上端から出た分子が、容器の下端から入ってくるように、立方体容器の前後、左右、上下をそれぞれ継いでしまう境界条件。外場が加わっていない系では、並進不変性(座標の原点をどこにとっても、系の性質が変化しない)がなりたつので、注目する粒子がいつも容器の中央にくるように、周期境界条件を施して座標系をずらすと、その粒子の周囲にはいつもたくさんの分子があるように見える。つまり、その粒子はバルク系の内部に存在しているかのごとくふるまい、表面の効果が物性に影響を与えない。凝縮系(固体、液体)のシミュレーションでは一般的に使われる手法である。

プログラムの流れ

1. 粒子を2つ選び、相対位置ベクトルを求めて、Lennard-Jones 相互作用ポテンシャルと分子間力、対ヴィリアルを計算する。2粒子それぞれに、加わる力を積算する。¹¹
2. すべての粒子対について、1の計算を行い、各粒子に加わる力を求める。
3. それぞれの粒子について、力から加速度を求め、あらかじめ指定された1ステップの時間 τ の間の速度の増分を計算して速度に加算する。
4. それぞれの粒子について、1ステップの時間 τ の間の位置の増分を速度から計算して座標に加算する。
5. ヴィリアルをもとに圧力を計算し、運動エネルギーから温度を計算する。時計を τ だけ進め、必要ならグラフィックスで粒子配置を表示する。

時間 τ を1ステップと呼び、これをさらに指定ステップ数だけ繰り返す。

NodeBoxによるデモ



NodeBoxはpython で手軽にグラフィックスを作成する開発環境である。剛体円板のシミュレーションが実際にどのように進行するかを、アニメーションで見よう。

1. 画面下のDockで、NodeBox(上のアイコン)をクリック。
2. サンプルプログラム1j2nb.pyを開き、メニュー”Python”→”Run”を選択。
3. 粒子が円で、速度の向きと大きさが線で表示されます。初速は1粒子にしか与えていませんが、最初の配置が密集配置になっているので、すぐに隣りの粒子に衝突し、エネルギーが分配され乱雑化していく様子が見えます。
4. 粒子数やセルの大きさはプログラムの以下の行を編集すれば変更できます。てきとうに変更しても構いません。

```
def setup():
    global system
    width = 10.0
    height = 10.0
    zoom = 10.0
    size(width*zoom, height*zoom)
    system=System(step=0, nballs=64, cell=[width, height],
                  gc=GC(zoom=zoom),logfile=None, kT=1.0)
```

¹¹ 作用反作用の法則により、同じ大きさで逆方向の力がそれぞれの粒子に加わる。相対位置ベクトルを計算する時に周期境界条件を考慮する。

コマンドラインでの実行



グラフィックス表示を見ると、何がおこっているのかわかりやすいのだが、表示にかなりの時間をとられるので、実際に統計をとる場合は、コマンドプロンプト(ターミナル)からlj2(c++版)を直接起動する。

1. ターミナルを起動する。
2. lj2プログラムのパラメータで、セルサイズや粒子数を指定する。例えば、サイズ10x10x10の3次元の箱の中に、10粒子を入れ、温度を1.5とし、1ステップの時間 τ を0.01とし、10000ステップ計算し、速度の統計データを記録し、結果をresult.*ファイルに記録する場合は、以下のようにパラメータを与える。(下線部を自分で入力)¹²

```
prompt% ./lj2 -c 10.0,10.0,10.0 -a 10 -t 1.5 -d 0.01 -v 100 10000 result
```

3. しばらくすると、計算が終わり、result.*という名前のファイルがいくつかできる。ファイルの内容は以下の通り。

result.log

温度Tや圧縮因子zなどの熱力学量を、1ステップごとに記録する。統計熱力学量の計算に必要。第1カラムから順に、ステップ数、温度、圧縮因子z、全エネルギー、全運動エネルギー、全ポテンシャルエネルギー。

```
0 1.44667548062 0.700718922827 10.6960623435 21.7001322093 -11.0040698658
1 1.4347929526 0.621269224477 10.6823699998 21.521894289 -10.8395242892
2 1.41637853495 0.504781702639 10.6690066739 21.2456780243 -10.5766713503
3 1.39132338647 0.350001929742 10.6577464603 20.8698507971 -10.2121043368
4 1.36101941716 0.164269515884 10.6501951221 20.4152912574 -9.76509613534
5 1.32946800359 -0.0274463971265 10.6462567325 19.9420200538 -9.29576332126
6 1.30284022574 -0.181160519215 10.6430674629 19.5426033861 -8.8995359232
7 1.28689101934 -0.250050734771 10.6367075295 19.30336529 -8.66665776056
8 1.28385328558 -0.211591347517 10.6258741781 19.2577992836 -8.63192510556
9 1.2916169396 -0.0789738540793 10.6126206958 19.374254094 -8.76163339824
```

result.vel

-vオプションで指定されたステップごとに、すべての粒子の速度を記録する。速度分布関数の作成に必要。

```
2.288821265
0.757782956849
1.88075060527
2.11801141791
2.37478916948
```

¹² コマンドを入力し、改行を押す前に、行末に"&"を付けておくと、プログラムをバックグラウンドで走らせることができます。ウィンドウの数を増やさず、たくさんの計算を一度に実行するのに便利。

```
2.6626721062
3.03540522723
1.79152332424
1.84315203122
1.0070923247
```

result.lj2

継続して同じ条件でシミュレーションを続けたい場合に、lj2に読みこませるデータ。解析には使用しない。hd2.pyのものと似ているが互換性はない。

```
10.0 10.0 10.0 ←箱の大きさ
10 ←粒子数
5.07454097703 4.22505053688 ←粒子0の座標
0.137268170372 -0.351641303045 ←粒子0の速度

3.37728452774 0.984760325895 ←粒子1の座標
1.63634974442 -0.475144372424 ←粒子1の速度

8.62933995064 9.93664758442
0.929881247531 0.556916128154
....
```

4. 直前の計算結果の*.lj2ファイルのデータを再利用し、温度をすこし下げて継続計算を行う。継続計算の場合は、温度を変えることはできるが、体積は変えられない。下の例では、result.lj2を読みこみ、温度を1.2に変更して10000ステップの継続計算を行う。

```
prompt% ./lj2 -t 1.2 -v 100 10000 result2 < result.lj2
```

コマンドラインパラメータ

使い方: ./lj2 [-a n][-d t][-t x][-c x,y,z][-v i] steps outputbase

- a n 粒子数を指定する.
- v i 粒子速度リストをiステップごとに出力する.
- d t 1ステップの時間(default=0.01).
- t x 目標温度を指定する.
- c x,y,z 箱の大きさを指定する.(3次元のみ)

-c と -a は両方一緒に使用する.

どちらも指定しない場合は、継続データ*.lj2を標準入力からよみこむ.

作業と解析

○作業○ いろんな温度・密度でシミュレーションを実行し、統計データを集めよ。

注意点

- 初期配置(結晶構造)から計算しはじめた直後や、温度設定を変更した直後、しばらくは系が非平衡になるので、統計に利用すべきでない。平衡状態に至ったかどうかは、その都度*.logに記録されている全エネルギーの値をgnuplotで表示するなどして確認する。平衡化が確認できたら、100,000ステップ分程度の継続計算を行って、統計に使用する。
- 分子数は64個以上を推奨する。
- 温度は $kT=\{1.5, 1.2, 1.0\}$ の3点を含む何点かを選ぶ。もっと低い温度を選ぶと、固液転移もおこるはず。
- 密度は $\rho=\{0.015, 0.03, 0.06, 0.09, 0.13, 0.2, 0.3, 0.5, 0.6, 0.8, 1.0\}$ ぐらいの範囲を選ぶ。¹³

○作業○ 前の作業で準備した統計プログラムを再利用し、温度と密度を適当に選んで速度分布を求め、Maxwell-Boltzmann分布からのずれを論じよ。

♪解析♪ 気液共存領域では、等温圧縮率が負になる(見かけ上、圧力を加えると膨張する!)不安定領域が粒子数有限個のシミュレーションでは見られる。体積---圧力のプロットにおいて、気液の平衡を与える密度を決定する方法として、Maxwellの等面積則がある。これは、横軸に体積、縦軸に圧力をとって、等温線を描き、これが極小と極大を持つ場合には、その中間に水平線を引き、水平線と等温線で囲まれた上下2つの領域の面積が等しくなる水平線が、気液平衡の圧力を与えるというものである。これを $T=1.0$ での結果に適用してみよ。

♪解析♪ 気液が平衡になる温度、圧力では、気体と液体の自由エネルギーは等しい。気液平衡が存在する各温度において、気体と液体のエンタルピー、エントロピー差を計算し、その大きさと符号について説明せよ。

★発展課題★ Maxwellの等面積則の原理を説明しなさい。

★発展課題★ 低密度では、ビリアル展開において第二ビリアルまでで圧力は近似される。第二ビリアル係数を計算結果から求め、剛体球の場合と比較せよ。

¹³ 液体と気体の中間の密度で温度を下げると、圧力が負になることがある。例えば注射器に泡が入らないように水を封じこめ、先に栓をして、ピストンを思いっきり引っ張った状態が負の圧力の状態である。植物は負圧によって根の水を葉まで持ち上げている。

TIPS

gnuplot

端末上で、ファイルdataに出力された速度分布関数や各種関数を表示させるには、gnuplotが便利です。

ファイルdataに書かれているデータをプロットする場合

```
prompt% gnuplot  
gnuplot> plot "data" with line
```

データファイルの形式は、数値を含む行が複数行並んだものを与えます。データが1カラムしかない場合は、横軸は行番号、縦軸が第1カラムの数値となります。データが2カラム以上ある場合は、横軸は第1カラムの数値、縦軸が第2カラムの数値となります。例えば、ファイルの内容が

```
1  
2  
4
```

なら、画面には座標(1,1)-(2,2)-(3,4)を結ぶ折線がプロットされます。また、

```
0.1 1  
0.2 2  
0.3 4
```

なら、画面には座標(0.1, 1)-(0.2, 2)-(0.3, 4)を結ぶ折線がプロットされます。

数式をプロットする場合

```
prompt% gnuplot  
gnuplot> plot sin(x)  
gnuplot> replot cos(x)
```

gnuplotを終了する

```
gnuplot> quit
```

gnuplotの中で使えるコマンドの説明は、gnuplotの中でhelpコマンドで見ることができます。

スクリプト

コマンドラインで実行したいコマンドを、ひとつのファイルに書いて、一気に処理させることができます。このようなファイルをスクリプトと呼びます。

```
prompt% sh scriptfile
```