

データサイエンス系科目群：シミュレーション実習 第 5 回 講義資料

担当：川崎猛史

名古屋大学大学院理学研究科理学専攻物理科学系・非平衡物理研究室 (R 研)

Last update: May 8, 2022

目次

- 1 講義のスケジュール
 - シラバス
- 2 第 4 回の続き
 - Box Muller method の高速化 (Marsaglia polar 法)
 - 様々な平均量
 - 時間平均
 - アンサンブル平均
- 3 第 4 回自主課題解説
- 4 多粒子系のブラウン運動 (Langevin 熱浴中の多粒子運動)
 - 相互作用ポテンシャルの一例
 - 相互作用力の計算方法
- 5 第 5 回自主課題
- 6 付録 1: スタックメモリと動的 (ヒープ) メモリ確保の比較 (C 言語)
- 7 付録 2: 関数におけるポインタ渡し (C 言語)
- 8 参考文献

目次

1 講義のスケジュール

■ シラバス

2 第4回の続き

- Box Muller method の高速化 (Marsaglia polar 法)
- 様々な平均量
 - 時間平均
 - アンサンブル平均

3 第4回自主課題解説

4 多粒子系のブラウン運動 (Langevin 熱浴中の多粒子運動)

- 相互作用ポテンシャルの一例
- 相互作用力の計算方法

5 第5回自主課題

6 付録1: スタックメモリと動的 (ヒープ) メモリ確保の比較 (C 言語)

7 付録2: 関数におけるポインタ渡し (C 言語)

8 参考文献

1. 講義のスケジュール

- 当実習は春 1 期にて実施する.
- 講義資料は各講義 **予定日当日朝 11 時迄**にアップロードする.
- スケジュール
 - 1** 4/11 : 第 1 回
 - 2** 4/18 : 第 2 回
 - 3** 4/25 : 第 3 回
 - 4** 5/02 : 第 4 回 (中間レポート課題公開)
 - 5** **5/09 : 第 5 回**
 - 6** 5/16 : 第 6 回 (5/15 23:55 中間レポート課題提出期限)
 - 7** 5/23 : **休講**
 - 8** 5/30 : 第 7 回
 - 9** 6/06 : 第 8 回 (期末レポート課題公開)
 - 10** 6/13
 - 11** 6/20 : (期末レポート課題提出期限予定)

1.1. シラバス

当実習では以下の内容を扱う予定である（進捗に合わせ変更する可能性がある）。

1 導入

- C(C++) の使い方 (主に数値計算)
- Python の使い方 (データ解析と作図)
- 数値計算の理念
- 桁落ち
- 科学計算における無次元化

2 常微分方程式の数値解法：減衰振動や調和振動子を例に

- 微分方程式の数値積分
- オイラー法
- 蛙飛び差分法（速度ベルレ法）
- 軌道の安定性と保存則

3 1 粒子系のブラウン運動

- ランジュバン方程式（確率微分方程式）
- 正規乱数の生成法
- オイラー・丸山法
- 時間平均とアンサンブル平均
- 拡散係数の計算

4 多粒子系のブラウン運動

- 相互作用力の計算方法
- 非平衡系のシミュレーション：相分離現象を例に

5 多粒子系の分子動力学シミュレーション

- 位置ベルレ法と速度ベルレ法
- 多粒子系における保存則（運動量・エネルギー・角運動量）

6 モンテカルロ法

- 統計力学の復習
- マルコフ連鎖モンテカルロ法
- メトロポリス判定法

目次

- 1 講義のスケジュール
 - シラバス
- 2 第 4 回の続き
 - Box Muller method の高速化 (Marsaglia polar 法)
 - 様々な平均量
 - 時間平均
 - アンサンブル平均
- 3 第 4 回自主課題解説
- 4 多粒子系のブラウン運動 (Langevin 熱浴中の多粒子運動)
 - 相互作用ポテンシャルの一例
 - 相互作用力の計算方法
- 5 第 5 回自主課題
- 6 付録 1: スタックメモリと動的 (ヒープ) メモリ確保の比較 (C 言語)
- 7 付録 2: 関数におけるポインタ渡し (C 言語)
- 8 参考文献

2.1. Box Muller method の高速化 (Marsaglia polar 法)

2.1 節の目的

ここでは、第4回で説明しきれなかった、Marsaglia polar 法のプログラムについて解説する。

リスト 1: 正規乱数の発生アルゴリズム (Box-Muller 法) 以下のプログラム"BM.h"は以下の GitHub リポジトリより取得可能[\[リンク\]](#)。サブルーチンとして移植するかヘッダとしてincludeして用いるとよい。

```
1 double unif_rand(double left, double right)
2 {
3     return left + (right - left)*rand()/RAND_MAX;
4 }
5 double gaussian_rand(void)
6 {
7     static double iset = 0;
8     static double gset;
9     double fac, rsq, v1, v2;
10
11     if (iset == 0) {
12         do {
13             v1 = unif_rand(-1, 1);
14             v2 = unif_rand(-1, 1);
15             rsq = v1*v1 + v2*v2;
```

2.1. Box Muller method の高速化 (Marsaglia polar 法) (2)

```
16     } while (rsq >= 1.0 || rsq == 0.0);  
17     fac = sqrt(-2.0*log(rsq)/rsq);  
18  
19     gset = v1*fac;  
20     iset = 0.50;  
21     return v2*fac;  
22 } else {  
23     iset = 0;  
24     return gset;  
25 }  
26 }
```


2.2. 様々な平均量

2.2 節の目的

数値計算結果は，統計誤差を含むため，解析の際，各種平均操作を行い，物理的に重要な要素を抜き出す必要がある．よく用いられる 2 つの平均手法を紹介する．

2.2.1. 時間平均

時間平均

定常状態の物理量について、各時刻の値に対して平均化する操作を時間平均という。

- この操作は定常状態における物理量 $X(t)$ に対して行われ、実際に

$$\langle X \rangle_{t_0} = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T dt_0 X(t_0) \quad (1)$$

と計算される。

- 同様に、この操作を 2 時刻相関関数 $C(t, t_0) = X(t + t_0)X(t_0)$ に対して施す場合、

$$C(t) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T dt_0 C(t, t_0) \quad (2)$$

となる。この時、 $C(t) = \langle X(t + t_0)X(t_0) \rangle_{t_0}$ 、あるいは、より単純に $C(t) = \langle X(t)X(0) \rangle$ と書く。

2.2.2. アンサンブル平均

アンサンブル平均

独立かつ同等な試行（実験）をいくつも行いを加算平均することをアンサンブル平均という。

- 熱平衡状態にある α 番目のサンプルの物理量を A_i とするとき、アンサンブル平均は

$$\langle A \rangle_{\text{ens}} = \frac{1}{N_{\text{ens}}} \sum_{\alpha=1}^{N_{\text{ens}}} A_{\alpha} \quad (3)$$

となる。

- このことは以下の統計力学平均と同等である。

$$\langle A \rangle_{\text{ens}} = \frac{\text{Tr} A(\mathbf{q}, \mathbf{p}) e^{-\beta \hat{H}(\mathbf{q}, \mathbf{p})}}{Z} \quad (4)$$

ここで、 Z は分配関数: $Z = \text{Tr} e^{-\beta \hat{H}(\mathbf{q}, \mathbf{p})}$ である。

補足 熱平衡状態では、時間平均とアンサンブル平均が等しくなる。

→ エルゴード仮説

目次

- 1 講義のスケジュール
 - シラバス
- 2 第 4 回の続き
 - Box Muller method の高速化 (Marsaglia polar 法)
 - 様々な平均量
 - 時間平均
 - アンサンブル平均
- 3 第 4 回自主課題解説
- 4 多粒子系のブラウン運動 (Langevin 熱浴中の多粒子運動)
 - 相互作用ポテンシャルの一例
 - 相互作用力の計算方法
- 5 第 5 回自主課題
- 6 付録 1: スタックメモリと動的 (ヒープ) メモリ確保の比較 (C 言語)
- 7 付録 2: 関数におけるポインタ渡し (C 言語)
- 8 参考文献

3. 第 4 回自主課題解説

第 4 回自主課題 1 粒子ブラウン運動の実装

温度 T ，摩擦係数が ζ である 3 次元溶媒中を熱揺動力による駆動される 1 粒子の運動を考える．この粒子の運動は Langevin 方程式 $m\dot{\mathbf{v}}(t) = -\zeta\mathbf{v}(t) + \mathbf{F}_B(t)$ でモデル化できることが広く知られている．この Langevin 方程式を，長さ，時間の単位をそれぞれ a ， $\frac{m}{\zeta}$ として無次元化すると， $T^* = \frac{mk_B T}{a^2 \zeta^2}$ がパラメータとなることを講義で扱った．そこで，この粒子の運動に関する以下の各問いに答えよ．なお，以下の $\langle \cdots \rangle$ は，アンサンブル平均や時間平均を十分とった量であることを表す．

- (1) 平均二乗変位に関する解析解 $\langle \mathbf{r}(t)^2 \rangle = \frac{2dk_B T}{\zeta} \left\{ t + \frac{m}{\zeta} e^{-\zeta t/m} - \frac{m}{\zeta} \right\}$ を無次元化しパラメータ T^* を用いて表せ．
- (2) 任意の T^* に対して，(1) の理論解と数値解が一致することを確認せよ．数値解は半陰 Euler・丸山法で求めよ．
- (3) 粒子の速度相関関数 $C(t) = \langle \mathbf{v}(t) \cdot \mathbf{v}(0) \rangle$ を計算せよ．
- (4) (3) で求めた $C(t)$ を時間積分することにより以下のグリーン・久保公式 $D = \frac{1}{d} \int_0^\infty dt C(t)$ が成立することを数値計算で確認せよ．

補足 (4) の数値積分は矩形や台形の短冊の和の計算をすればよい．

3. 第4回自主課題解説 (2)

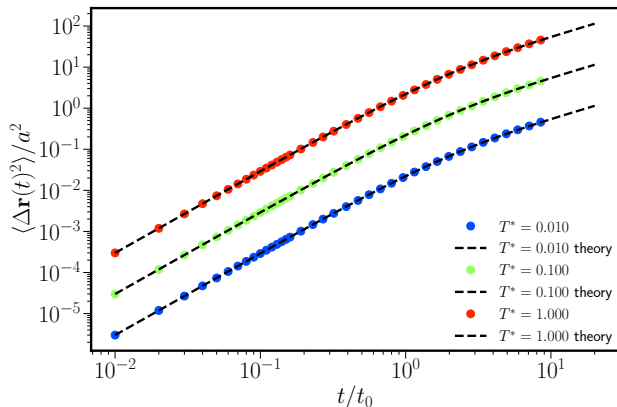


図 1: 平均二乗変位の数値結果. ここでは, パラメータ T^* を 0.01, 0.1, 1.0 と変化させた時の様子を表す. 点線は理論解 [式 (7)] を表す. 数値解とよく合っている様子が見取れる.

3. 第4回自主課題解説 (3)

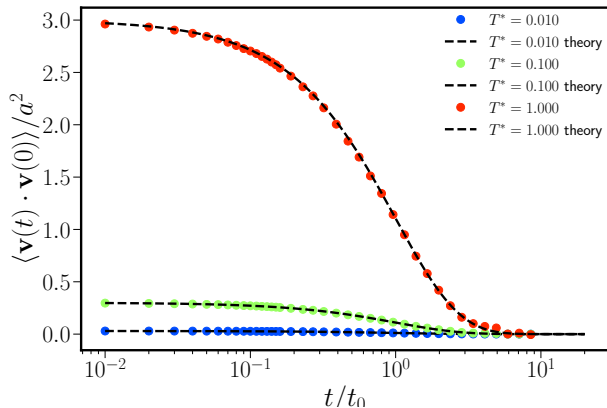


図 2: 速度相関関数の数値結果. パラメータ T^* を 0.01, 0.1, 1.0 と変化した時の様子を表す. 点線は理論解 [式 (13)] を表す. 数値解とそこそこ様子が見て取れる.

3. 第4回自主課題解説 (4)

解説

- (1) 平均二乗変位の解析解 $\langle \Delta \mathbf{r}(t)^2 \rangle = \frac{2dk_B T}{\zeta} \left\{ t + \frac{m}{\zeta} e^{-\zeta t/m} - \frac{m}{\zeta} \right\}$ を無次元化する．長さの単位 a ，時間の単位 $t_0 = m/\zeta$ に注意して，

$$\begin{aligned} a^2 \langle \Delta \tilde{\mathbf{r}}(\tilde{t})^2 \rangle &= \frac{2dk_B T}{\zeta} [t_0 \tilde{t} + t_0 e^{-\tilde{t}} - t_0] \\ &= \frac{2dmk_B T}{\zeta^2} [\tilde{t} + e^{-\tilde{t}} - 1] \end{aligned} \quad (5)$$

しかるに，

$$\begin{aligned} \langle \Delta \tilde{\mathbf{r}}(\tilde{t})^2 \rangle &= \frac{2dmk_B T}{\zeta^2 a^2} [\tilde{t} + e^{-\tilde{t}} - 1] \\ &= 2dT^* [\tilde{t} + e^{-\tilde{t}} - 1] \end{aligned} \quad (6)$$

つまり，3次元 ($d=3$) であれば

$$\boxed{\langle \tilde{\mathbf{r}}(\tilde{t})^2 \rangle = 6T^* [\tilde{t} + e^{-\tilde{t}} - 1]} \quad (7)$$

を得る．

3. 第 4 回自主課題解説 (5)

補足

短時間極限では、 $e^{-\tilde{t}} \sim 1 - \tilde{t} + \frac{1}{2}\tilde{t}^2$ と展開することで、

$$\langle \Delta \tilde{\mathbf{r}}(\tilde{t})^2 \rangle \sim 3T^* \tilde{t}^2 \quad (8)$$

と表される弾道軌道を得る．また長時間極限では、 $e^{-\tilde{t}} - 1$ の項が落ちるので

$$\langle \Delta \tilde{\mathbf{r}}(\tilde{t})^2 \rangle \sim 6T^* \tilde{t} \quad (9)$$

と表される拡散軌道を得る．つまり無次元化された拡散係数は、パラメータ T^* そのものの値となる．

- (2) C 言語による数値計算のコーディング例をリスト 2,3 に示した．運動方程式を解く主計算プログラムは “langevin.cpp”，そこから吐き出される座標と速度のデータを用いて、平均二乗変位と速度相関関数の解析プログラムは “analyze.cpp” とした．平均二乗変位の数値結果を図 1 に示した．ここでは、パラメータ T^* を 0.01, 0.1, 1.0 と変化させた時の様子を表す．点線は理論解 [式 (7)] を表す．よく合っている様子が見て取れる．

3. 第4回自主課題解説 (6)

(3) 第3回講義で説明した通り，ブラウン粒子の速度相関関数は $t \geq 0$ のとき，

$$C(t) = \langle \mathbf{v}(t) \cdot \mathbf{v}(0) \rangle = \frac{dk_B T}{m} e^{-\zeta t/m} \quad (10)$$

となる．これらを無次元化すると

$$\frac{a^2}{t_0^2} \langle \tilde{\mathbf{v}}(\tilde{t}) \cdot \tilde{\mathbf{v}}(0) \rangle = \frac{dk_B T}{m} e^{-\zeta t/m} \quad (11)$$

であるから

$$\langle \tilde{\mathbf{v}}(\tilde{t}) \cdot \tilde{\mathbf{v}}(0) \rangle = \frac{dmk_B T}{\zeta^2 a^2} e^{-\zeta t/m} \quad (12)$$

$$= 3T^* e^{-\tilde{t}} \quad (13)$$

となる．ここで $d = 3$ とした．この理論解と数値解の比較を図2に記した．理論と数値解が平均二乗変位ほどではないがよく合っている様子が見て取れる．

3. 第 4 回自主課題解説 (7)

(4) 速度相関関数の理論式を以下の通り時間積分すると

$$\frac{1}{d} \int_0^\infty C(t) dt = \frac{k_B T}{\zeta} = D \quad (14)$$

となり、拡散係数と速度相関関数に関する Green-久保公式が得られる（第 3 回講義参照）。これらの両辺を無次元化すると

$$\frac{1}{d} \int_0^\infty \frac{a^2}{t_0^2} \tilde{C}(\tilde{t}) t_0 d\tilde{t} = \frac{k_B T}{\zeta} \quad (15)$$

である。しかるに

$$\begin{aligned} \int_0^\infty \tilde{C}(\tilde{t}) t_0 d\tilde{t} &= \frac{dk_B T t_0}{a^2 \zeta} = \frac{mk_B T d}{a^2 \zeta^2} \\ &= T^* d \end{aligned} \quad (16)$$

となる。数値計算で得られた速度相関関数を無次元化された時間に関して積分すれば、 $d = 3$ のとき、面積が $3T^*$ になることを確認すればよい。

3. 第 4 回自主課題解説 (8)

リスト 2: "langevin.cpp" 以下の GitHub リポジトリより取得可能[\[リンク\]](#)

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <math.h>
4 #include <iostream>
5 #include <fstream>
6 #include <cfloat>
7 #include "BM.h"
8
9 #define tmax 10
10 #define dt 0.01
11 #define temp 1.0
12 #define ensemble 10000
13 #define dim 3
14 //using namespace std;
15
16 void ini_phase(double *x, double *v){
17     int i;
18     for(i=0; i<dim; i++){
19         x[i]=0.;
20         v[i]=0.;
21     }
22 }
23
24 void ini_clock(int *j, double *tout){
```

3. 第4回自主課題解説 (9)

```

25     *j=0;
26     *tout=1.e-2;
27 }
28
29 void eom(double *v,double *x){
30     int i;
31     for(i=0;i<dim;i++){
32         v[i]+=-v[i]*dt+sqrt(2.*temp*dt)*gaussian_rand();
33         x[i]+=v[i]*dt;
34     }
35 }
36
37 void output(double *x,double *v,int j){
38     char filename[128];
39     std::ofstream file;
40
41     sprintf(filename,"coord_dt%.3fT%.3f.dat",dt,temp);
42     file.open(filename,std::ios::app); //append
43     file <<j*dt<<"\t"<<x[0]<<"\t"<<x[1]<<"\t"<<x[2]<<std::endl;
44     // std::cout<<j*dt<<"\t"<<x[0]<<"\t"<<x[1]<<"\t"<<x[2]<<std::endl;
45     file.close();
46
47     sprintf(filename,"vel_dt%.3fT%.3f.dat",dt,temp);
48     file.open(filename,std::ios::app); //append
49     file <<j*dt<<"\t"<<v[0]<<"\t"<<v[1]<<"\t"<<v[2]<<std::endl;
50     file.close();

```

3. 第 4 回自主課題解説 (10)

```
51 }
52 }
53
54 int main(){
55     double x[dim],v[dim],t,tout;
56     int i,j;
57     ini_phase(x,v);
58     for(i=0;i<ensemble;i++){
59         ini_clock(&j,&tout);
60         output(x,v,j);
61         while(j*dt < tmax){
62             j++;
63             eom(v,x);
64             if(j*dt >= tout){
65                 output(x,v,j);
66                 tout*=1.2;
67             }
68         }
69     }
70     return 0;
71 }
```

3. 第 4 回自主課題解説 (11)

リスト 3: “analyze.cpp” 以下の GitHub リポジトリより取得可能[\[リンク\]](#)

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <math.h>
4 #include <iostream>
5 #include <fstream>
6 #include <cfloat>
7 #include "BM.h"
8
9 #define temp 0.01
10 #define dt 0.01
11 #define ensemble 10000
12 #define window 39
13 #define dim 3
14 //using namespace std;
15
16 void ini(double *dr2, double *corr){
17     for(int i=0; i<window; i++){
18         dr2[i]=0.0;
19         corr[i]=0.0;
20     }
21 }
22
23 void input(double (*x)[dim], double (*v)[dim], double *t){
24     char filename[128];
```

3. 第4回自主課題解説 (12)

```

25  std::ifstream file;
26  sprintf(filename, "coord_dt%.3fT%.3f.dat", dt, temp);
27  file.open(filename);
28  int asize=ensemble*window;
29  for(int i=0;i<asize;i++){
30      file >> t[i] >> x[i][0] >> x[i][1] >> x[i][2];
31  }
32  file.close();
33
34  sprintf(filename, "vel_dt%.3fT%.3f.dat", dt, temp);
35  file.open(filename);
36  for(int i=0;i<asize;i++){
37      file >> t[i] >> v[i][0] >> v[i][1] >> v[i][2];
38      // std::cout << t[i] <<"\t"<<v[0][i]<<"\t"<<v[1][i]<<"\t"<<v[2][i]
        <<std::endl;
39  }
40  file.close();
41  }
42
43  void output(double *t, double *dr2, double *corr, double integral){
44      char filename[128];
45      std::ofstream file;
46      sprintf(filename, "msd_dt%.3fT%.3f.dat", dt, temp);
47      file.open(filename);
48      for(int i=1;i<window;i++)
49          file<<t[i]-t[0]<<"\t"<<dr2[i]<<std::endl;

```


3. 第4回自主課題解説 (13)

```

50 file.close();
51
52 sprintf(filename, "corr_dt%.3fT%.3f.dat", dt, temp);
53 file.open(filename);
54 for(int i=1; i<window; i++)
55     file<<t[i]-t[0]<<"\t"<<corr[i]<<std::endl;
56 file.close();
57
58 std::cout<<"A result of the integral ="<<integral<<std::endl;
59 }
60
61 void analyze(double (*x)[dim], double (*v)[dim], double *t, double *dr2,
62             double *corr, double *integral){
63     double dx[dim], corr_x[dim];
64     for(int i=0; i<ensemble; i++){
65         for(int j=0; j<window; j++){
66             for(int k=0; k<dim; k++){
67                 dx[k]=(x[j+window*i][k]-x[window*i][k]);
68                 corr_x[k]=v[j+window*i][k]*v[window*i][k];
69                 dr2[j]+=(dx[k]*dx[k])/ensemble;
70                 corr[j]+=(corr_x[k])/ensemble;
71                 if(j!=window-1)
72                     *integral+=0.5*(v[j+window*i][k]*v[window*i][k]+v[j+1+window*
73                     i][k]*v[window*i][k])*(t[j+1]-t[j])/ensemble;
74             }
75         }
76     }
77 }

```

3. 第4回自主課題解説 (14)

```
74 }  
75  
76 int main(){  
77     double t[ensemble*window], dr2[window], corr[window], integral=0.0;  
78     int i, j;  
79     double (*x)[dim] = new double[ensemble*window][dim];  
80     double (*v)[dim] = new double[ensemble*window][dim];  
81  
82     ini(dr2, corr);  
83     input(x, v, t);  
84     analyze(x, v, t, dr2, corr, &integral);  
85     output(t, dr2, corr, integral);  
86     delete[] x;  
87     delete[] v;  
88     return 0;  
89 }
```

目次

- 1 講義のスケジュール
 - シラバス
- 2 第 4 回の続き
 - Box Muller method の高速化 (Marsaglia polar 法)
 - 様々な平均量
 - 時間平均
 - アンサンブル平均
- 3 第 4 回自主課題解説
- 4 多粒子系のブラウン運動 (Langevin 熱浴中の多粒子運動)
 - 相互作用ポテンシャルの一例
 - 相互作用力の計算方法
- 5 第 5 回自主課題
- 6 付録 1: スタックメモリと動的 (ヒープ) メモリ確保の比較 (C 言語)
- 7 付録 2: 関数におけるポインタ渡し (C 言語)
- 8 参考文献

4. 多粒子系のブラウン運動 (Langevin 熱浴中の多粒子運動)

4 章の目的

- これまで Langevin 熱浴中を運動する 1 粒子の運動方程式やその運動 (ブラウン運動) について理論的に議論した [1].
- ここからは, Langevin 熱浴中の多粒子運動を扱う.
- 多粒子が織りなす系の物性に対して, 粒子間相互作用が重要な役割を担う.
 - 例えば, 気液転移は, 相互作用に引力が存在する場合にのみ起こる.
- 今回は分子シミュレーションで用いられる代表的な相互作用ポテンシャルと相互作用力の計算方法を説明する.
- 次回は, 系の体積を定義する上で不可欠な (周期的) 境界条件を導入する.
- 具体的な多体シミュレーションを実装する.

4.1. 相互作用ポテンシャルの一例

相互作用ポテンシャルは実に多様であるが、分子シミュレーションで用いるポテンシャルのごく一例を紹介する。

(1) Lennard-Jones ポテンシャル [2, 3, 4](図 3 参照)

$$U(r_{jk}) = 4\epsilon \left\{ \left(\frac{a_{jk}}{r_{jk}} \right)^{12} - \left(\frac{a_{jk}}{r_{jk}} \right)^6 \right\} + C_{jk} \quad (r_{jk} < a_{\text{cut}} \sim 2.5a) \quad (17)$$

■ 変数の意味：

- $a_{ij} = \frac{a_i + a_j}{2}$ (平均粒径: 粒子接触を特徴づける距離) .
- ϵ は物質固有のエネルギー . (LJ エネルギー単位ともいう)
- C_{jk} はカットオフエネルギーであり、カットオフ長 a_{cut} において、 $U(a_{\text{cut}}) = 0$ になるようにおく .

■ 性質：

- 斥力部分と引力部分から成る .
- モデル分子として広く用いられる (汎用性が高い) .
- 斥力の起源は、原子同士の接近の際、パウリの排他率による生じる反発力 (冪の 12 乗自体には物理的根拠はないが広く実験を再現する) .
- 引力の起源は、原子間の双極子 . 双極子間の相互作用 (これで冪の 6 乗が出る) .

4.1. 相互作用ポテンシャルの一例 (2)

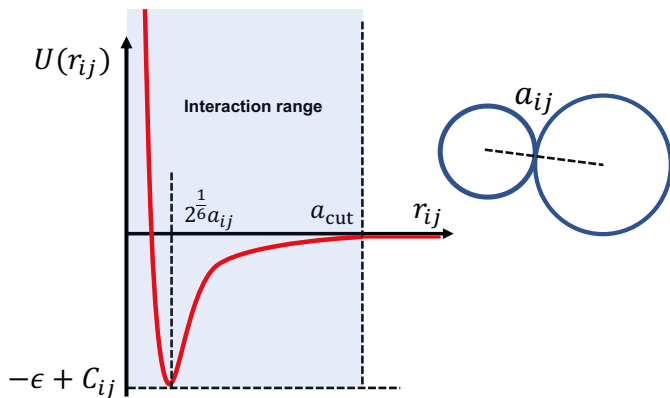


図 3: Lennard-Jones ポテンシャルの概形．粒子接触長 a_{ij} に関する絵解き図．

4.1. 相互作用ポテンシャルの一例 (3)

(2) Coulomb ポテンシャル [3, 4, 5, 6]

$$U(r_{jk}) = \frac{Cq_iq_j}{\epsilon r} \quad (r_{jk} < a_{\text{cut}}) \quad (18)$$

- ϵ : 誘電率, C : エネルギー変換定数.
- $q_{i(j)}$: 粒子 $i(j)$ の電荷.
- 相互作用レンジが長いので, Ewald 法など計算の工夫が必要になる.

4.2. 相互作用力の計算方法

前節で導入したポテンシャルエネルギーの位置微分により、粒子 j にかかる力 \mathbf{F}_j^I が以下のように計算できる。

$$\begin{aligned}
 \mathbf{F}_j^I &= - \sum_{k \neq j} \frac{\partial U(\mathbf{r}_{jk})}{\partial \mathbf{r}_j} = - \sum_{k \neq j} \frac{\partial r_{jk}}{\partial \mathbf{r}_j} \frac{\partial U(\mathbf{r}_{jk})}{\partial r_{jk}} \\
 &= - \sum_{k \neq j} \begin{pmatrix} \frac{\partial r_{jk}}{\partial x_j} \\ \frac{\partial r_{jk}}{\partial y_j} \\ \frac{\partial r_{jk}}{\partial z_j} \end{pmatrix} \frac{\partial U(\mathbf{r}_{jk})}{\partial r_{jk}} \\
 &= - \sum_{k \neq j} \begin{pmatrix} \frac{x_{jk}}{r_{jk}} \\ \frac{y_{jk}}{r_{jk}} \\ \frac{z_{jk}}{r_{jk}} \end{pmatrix} \frac{\partial U(\mathbf{r}_{jk})}{\partial r_{jk}} \\
 &= - \sum_{k \neq j} \frac{\mathbf{r}_{jk}}{r_{jk}} \underbrace{\frac{\partial U(\mathbf{r}_{jk})}{\partial r_{jk}}}
 \end{aligned}$$

この項は解析的に計算

次回は、周期境界条件下における力の計算方法について説明する。

目次

- 1 講義のスケジュール
 - シラバス
- 2 第 4 回の続き
 - Box Muller method の高速化 (Marsaglia polar 法)
 - 様々な平均量
 - 時間平均
 - アンサンブル平均
- 3 第 4 回自主課題解説
- 4 多粒子系のブラウン運動 (Langevin 熱浴中の多粒子運動)
 - 相互作用ポテンシャルの一例
 - 相互作用力の計算方法
- 5 第 5 回自主課題
- 6 付録 1: スタックメモリと動的 (ヒープ) メモリ確保の比較 (C 言語)
- 7 付録 2: 関数におけるポインタ渡し (C 言語)
- 8 参考文献

5. 第 5 回自主課題

第 5 回自主課題 多粒子計算の準備

2 次元平面中に、粒径（直径）1 の円を一辺の長さ $L = 40$ の正方形の空間に 512 個の粒子を均等に配置する．

- (1) 粒子を正方格子で配置し、その結果を図示せよ．
- (2) 粒子を六方格子で配置し、その結果を図示せよ．
- (3) (発展) 正方形境界が周期境界である際、ある粒子 i と他の粒子 j の距離を測るアルゴリズムを考えよ．（力の計算等に必要）
- (4) (発展) (3) の距離が（例えば）5 以下の “粒子番号” を格納する配列を計算するアルゴリズムを考えよ．（ベルレ帳簿法）

目次

- 1 講義のスケジュール
 - シラバス
- 2 第 4 回の続き
 - Box Muller method の高速化（Marsaglia polar 法）
 - 様々な平均量
 - 時間平均
 - アンサンブル平均
- 3 第 4 回自主課題解説
- 4 多粒子系のブラウン運動（Langevin 熱浴中の多粒子運動）
 - 相互作用ポテンシャルの一例
 - 相互作用力の計算方法
- 5 第 5 回自主課題
- 6 付録 1：スタックメモリと動的（ヒープ）メモリ確保の比較（C 言語）
- 7 付録 2：関数におけるポインタ渡し（C 言語）
- 8 参考文献

6. 付録 1: スタックメモリと動的 (ヒープ) メモリ確保の比較 (C 言語)

リスト 4: スタックメモリを用いた変数・配列の確保 (C 言語の例). スタックメモリ (スタック領域におけるメモリ) は, ラップトップ PC では 10MB 程度が上限 (かなり小さい) [7].

```
1 double x,y[10000],z[100000][10];
```

リスト 5: 動的メモリの確保 (C++の例). 変数と配列でメモリの解放のレトリックが異なる. メモリ (ヒープ領域) の上限はラップトップ PC で GB オーダまで確保可能 [7]. C 言語では malloc 関数を用いることにより同等のことができる.

```
1 double *x = new double;  
2 double *y = new double[10000];  
3 double (*z)[10] = new double[10000][10];  
4 // 不要になったらメモリを解放  
5 delete x;  
6 delete [] y;  
7 delete [] z;
```

目次

- 1 講義のスケジュール
 - シラバス
- 2 第 4 回の続き
 - Box Muller method の高速化 (Marsaglia polar 法)
 - 様々な平均量
 - 時間平均
 - アンサンブル平均
- 3 第 4 回自主課題解説
- 4 多粒子系のブラウン運動 (Langevin 熱浴中の多粒子運動)
 - 相互作用ポテンシャルの一例
 - 相互作用力の計算方法
- 5 第 5 回自主課題
- 6 付録 1：スタックメモリと動的（ヒープ）メモリ確保の比較 (C 言語)
- 7 付録 2：関数におけるポインタ渡し (C 言語)
- 8 参考文献

7. 付録 2：関数におけるポインタ渡し (C 言語)

リスト 6: C 言語におけるポインタ渡しのレトリック:以下の型を常に参考にするとうい。

```
1
2 void function(double *x, double *y, double (*z)[10]){
3     *x=1.0;
4     for(int i=0;i<10000;i++)
5         y[i]=i;
6
7     for(int i=0;i<10000;i++)
8         for(int j=0;j<10;j++)
9             z[i][j]=i*j;
10 }
11
12 int main(){
13     double x, y[10000], z[10000][10];
14     function(&x,y,z);
15     return 0;
16 }
```

目次

- 1 講義のスケジュール
 - シラバス
- 2 第4回の続き
 - Box Muller method の高速化 (Marsaglia polar 法)
 - 様々な平均量
 - 時間平均
 - アンサンブル平均
- 3 第4回自主課題解説
- 4 多粒子系のブラウン運動 (Langevin 熱浴中の多粒子運動)
 - 相互作用ポテンシャルの一例
 - 相互作用力の計算方法
- 5 第5回自主課題
- 6 付録1: スタックメモリと動的 (ヒープ) メモリ確保の比較 (C 言語)
- 7 付録2: 関数におけるポインタ渡し (C 言語)
- 8 参考文献

参考文献・ウェブサイト

- [1] Zwanzig R (2001) Nonequilibrium Statistical Mechanics.
(Oxford University Press, Oxford ; New York).
- [2] Verlet L (1967) Computer "Experiments" on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules.
Physical Review 159(1):98–103.
- [3] Allen MP, Tildesley DJ (2017) Computer Simulation of Liquids: Second Edition.
(Oxford University Press).
- [4] Frenkel D, Smit B (2001) Understanding Molecular Simulation: From Algorithms to Applications.
(Elsevier).
- [5] Kolafa J, Perram JW (1992) Cutoff Errors in the Ewald Summation Formulae for Point Charge Systems.
Molecular Simulation 9(5):351–368.
- [6] Di Pierro M, Elber R, Leimkuhler B (2015) A Stochastic Algorithm for the Isobaric–Isothermal Ensemble with Ewald Summations for All Long Range Forces.
Journal of Chemical Theory and Computation 11(12):5624–5637.

参考文献・ウェブサイト (2)

- [7] Lemniscater N (year?) C++のスタックメモリと動的メモリの上限値調査
(<https://qiita.com/LemniscaterN/items/a3abfa143612cb928bde>).