

# **R lab simulation seminar**

## Chapter 3: Brownian dynamics simulations for many particles

Takeshi Kawasaki

Department of Physics, Nagoya University

Last-update May 14, 2022

# Agenda

## 1 Inter-particle interaction force

- Two body interaction potentials
- Force calculation
- Periodic boundary condition
- Initial positions of the particles

## 2 Brownian dynamics simulations for many particles

- Non-dimensionallization of Langevin equation
- Discretization of the equation: Euler method
- Homework 3
- Radial distribution function
- Static structure factor
- Intermediate scattering function
- Logarithmic sampling
- Center of mass correction
- Homework 4

## 3 Speedup technique

- Introduction of the speedup technique
- Verlet list method
- Cell list method
- Homework 5

## Section 1

### Inter-particle interaction force

## 1.1 Two body interaction potentials

- In this chapter, we extend the one-particle Brownian dynamics to those of multi-particles.
- Here we deal with  $N$  Brownian particles whose diameters are  $a_j$  ( $j = 1, \dots, N$ ).
- In order to simulate these, we introduce interaction potentials among the particles.
- Typically the two body interaction potential for particle  $j$  and  $k$  is as a function of the **relative vector**  $\mathbf{r}_{jk} = \mathbf{r}_j - \mathbf{r}_k$  (i.e.,  $u(\mathbf{r}_{jk})$ ).
- The total potential energy is represented as

$$U_{\text{tot}} = \sum_{j>k} u(\mathbf{r}_{jk}) \quad (1)$$

- Now typical interaction potentials for molecules, colloids, and grains are introduced as follows.

### (1) Lennard-Jones potential

- The Lennard-Jones potential is most famous inter-particle potential in molecular dynamics simulations.

## 1.1 Two body interaction potentials (2)

- This potential is composed of the attractive and repulsive parts, represented as

$$u(r_{jk}) = 4\epsilon \left\{ \left( \frac{a_{jk}}{r_{jk}} \right)^{12} - \left( \frac{a_{jk}}{r_{jk}} \right)^6 \right\} + C_{jk} \quad (r_{jk} < a_{\text{cut}}).$$

- Typically,  $a_{jk} = \frac{a_j + a_k}{2}$  and  $C_{jk}$  is the cutoff energy, so that  $U(a_{\text{cut}}) = 0$  at the cutoff length  $a_{\text{cut}} \sim 3.0$ .

### (2) Weeks-Chandler-Andersen (WCA) potential

[J. D. Weeks, D. Chandler, and H. C. Andersen, The Journal of Chemical Physics **54**, 5237 (1971)]

- Weeks-Chandler-Andersen (WCA) potential is the truncated Lennard-Jones potential, which mimics a hard sphere.
- This potential is represented as

$$u(r_{jk}) = 4\epsilon \left\{ \left( \frac{a_{jk}}{r_{jk}} \right)^{12} - \left( \frac{a_{jk}}{r_{jk}} \right)^6 + \frac{1}{4} \right\} \quad (r_{jk} < a_{\text{cut}}),$$

- The cut-off length  $a_{\text{cut}} = 2^{1/6} a_{jk}$ . If  $r_{jk} > a_{\text{cut}}$ , then  $U(r_{jk}) = 0$ .
- That is, it becomes a simple repulsive potential.
- At low temperatures, the hard sphere is well approximated.

## 1.1 Two body interaction potentials (3)

### (3) Bernu-Hansen-Hiwatari-Pastore (BHHP) potential

[B. Bernu, J. P. Hansen, Y. Hiwatari, and G. Pastore, Phys. Rev. A **36**, 4891 (1987)]

- BHHP potential is a repulsive potential frequently used in glass simulations, which is represented as

$$u(r_{jk}) = \epsilon \left( \frac{a_{jk}}{r_{jk}} \right)^{12} + C_{jk} \quad (r_{jk} < a_{\text{cut}}),$$

- The cut-off length  $a_{\text{cut}} \sim 4.0a_{jk}$ .
- If  $r_{jk} > a_{\text{cut}}$ , then  $U(r_{jk}) = 0$ .
- That is, it becomes a simple repulsive potential.

### (4) Harmonic sphere potential

- Harmonic sphere potential is a very simple repulsive potential frequently used in jamming simulations, which is represented as

$$u(r_{jk}) = \frac{\epsilon}{2} \left( 1 - \frac{r_{jk}}{a_{jk}} \right)^2 \quad (r_{jk} < a_{jk}),$$

- The cut-off length  $a_{\text{cut}} = a_{jk}$ .
- If  $r_{jk} > a_{\text{cut}}$ , then  $U(r_{jk}) = 0$ .

## 1.1 Two body interaction potentials (4)

- That is, it becomes simply repulsive and the particles can be overlapped.

### (5) Hertzian potential

- A repulsive potential frequently used in jamming simulations.
- The potential is represented as

$$u(r_{jk}) = \frac{\epsilon}{\alpha} \left( 1 - \frac{r_{jk}}{a_{jk}} \right)^\alpha \quad (r_{jk} < a_{jk})$$

- The cut-off length  $a_{\text{cut}} = a_{jk}$  and the exponent  $\alpha$  is 5/2.
- If  $r_{jk} > a_{\text{cut}}$ , then  $U(r_{jk}) = 0$ .
- That is, it becomes simply repulsive and the particles can be overlapped.

## 1.2 Force calculation

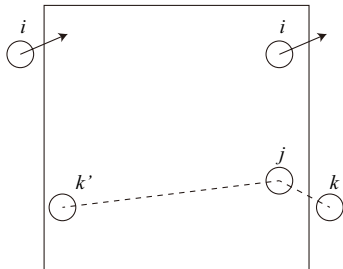
- Using the potential energy, the interaction force can be calculated.
- The force  $\mathbf{F}_j^l$  acting on the particle  $j$  is obtained as

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



## 1.3 Periodic boundary condition

- When dealing with the motion of many particles, **the particle density** plays an important role in determining the physical properties of the system.
- To include density effects, one must set a boundary.
- The most realistic and simple boundary is the hard wall.
- However, when the wall is installed, the dynamics and structure of the particles near the wall are different from those of the bulk.
- **The periodic boundary**, on the other hand, are often used to prevent such effects.



**FIG 1:** Periodic boundary conditions. The particle  $i$  from the boundary in the  $x$  direction emerges from the left wall. It must be determined that the particles  $k$  and  $k'$  are the same particle, but that it is the particle  $k$  which interacts with the particle  $j$ .

Listing 1: The periodic boundary condition for the coordinates.

## 1.3 Periodic boundary condition (2)

```
int p_bound(double* x, double* y, int Np, double L) {
    int k;
    for (k = 0; k < Np; k++) {
        if (x[k] < 0.0) {
            x[k] = x[k] + L;
        }
        if (x[k] > L) {
            x[k] = x[k] - L;
        }
        if (y[k] < 0.0) {
            y[k] = y[k] + L;
        }
        if (y[k] > L) {
            y[k] = y[k] - L;
        }
    }
    return 0;
}
```

## 1.3 Periodic boundary condition (3)

Listing 2: Subroutines for two-body interaction calculation (Harmonic sphere potential)

```
int calc_force_hs(double* x, double* y, double L, int Np, double* a, double* fx,
    double* fy, double* avU) {
    int i, j, k;
    *avU = 0.0;
    double r;
    double t, f;
    double dx, dy;
    double aij;
    double cut;
    for (k = 0; k < Np; k++) {
        fx[k] = 0.0;
        fy[k] = 0.0;
    }
    for (i = 0; i < Np; i++){
        for (j = i + 1; j < Np; j++){
            dx = x[i] - x[j];
            dy = y[i] - y[j];
            if (dx > (0.5 * L))
                dx -= L;
            if (dx < -(0.5 * L))
                dx += L;
            if (dy > (0.5 * L))
                dy -= L;
```

## 1.3 Periodic boundary condition (4)

```
if (dy < -(0.5 * L))
  dy += L;
aij = (a[i] + a[j]) / 2.0;
r = sqrt(dx * dx + dy * dy);
t = r / aij;
cut = aij;
if (r < cut) {
  f = -(1 - t) / aij; //analytical calculation of the 1'st derivative
}
else {
  f = 0.0;
  continue;
}
fx[i] -= f * dx / r;
fx[j] += f * dx / r;
fy[i] -= f * dy / r;
fy[j] += f * dy / r;
*avU += (1 - t) * (1 - t);
}
}
*avU /= double(Np);
return 0;
}
```

## 1.4 Initial positions of the particles

When you start the simulation, you need to specify some particle positions. Here are some typical ways to do this:

- (1) Making crystal.
- (2) Making random (It is difficult in the LJ system. It works for Harmonic sphere and Heltzian potential systems. It has the advantage of being easy to implement.)

```
int square_lattice(double *x,double *y, double L, int Np)
{
    int i,j;
    int N;
    N=(int)sqrt(Np)+1;
    double l = L/N; // lattice constant
    for(i = 0; i < N; i++){
        for(j = 0; j < N; j++){
            k = (i*N+j);
            if(k>Np) break;
            x[k] = l*i;
            y[k] = l*j;
        }
    }
    return 0;
}
```

## Section 2

# Brownian dynamics simulations for many particles

## 2.1 Non-dimensionalization of Langevin equation

- Here we handle the motion of a many- (colloidal) Brownian particles with the constant temperature.
- The motion of the colloidal particle  $j$  is driven by the following Langevin equation:

$$m \frac{d\mathbf{v}_j(t)}{dt} = -\zeta \mathbf{v}_j(t) + \mathbf{F}_j^I(t) + \mathbf{F}_j^B(t).$$

- Here,  $\mathbf{F}_j^I(t)$  is the force acting on the particle  $j$ .
- Now, we make the equation of motions **dimensionless** as follows:

$$\underbrace{m \frac{a_0}{t_0^2} \frac{d\tilde{\mathbf{v}}_j(\tilde{t})}{d\tilde{t}}}_{(1)} = \underbrace{-\zeta \frac{a_0}{t_0} \tilde{\mathbf{v}}_j(\tilde{t})}_{(2)} + \underbrace{\frac{\epsilon}{a_0} \tilde{\mathbf{F}}_j}_{(3)} + \underbrace{\sqrt{2k_B T \zeta} \frac{1}{t_0 \Delta \tilde{t}} \mathbf{R}_G}_{(4)}$$

## 2.1 Non-dimensionalization of Langevin equation (2)

- By dividing both sides of the equation by the coefficient of the term (1), we gain

$$\underbrace{\frac{d\tilde{\mathbf{v}}_j(\tilde{t})}{d\tilde{t}}}_{(1)} = - \underbrace{\frac{\zeta t_0}{m} \tilde{\mathbf{v}}_j(\tilde{t})}_{(2)} + \underbrace{\frac{t_0^2 \epsilon}{ma_0^2} \tilde{\mathbf{F}}_j}_{(3)} + \underbrace{\sqrt{2k_B T \zeta \frac{t_0^3}{m^2 a_0^2 \Delta \tilde{t}}} \mathbf{R}_G}_{(4)}$$

- Here, in order to make the coefficient of the term (3) unity, the time scale is determined as

$$t_0 = \sqrt{\frac{ma_0^2}{\epsilon}}.$$



## 2.1 Non-dimensionalization of Langevin equation (3)

■ Accordingly, we obtain

$$\underbrace{\frac{d\tilde{\mathbf{v}}_j(\tilde{t})}{d\tilde{t}}}_{(1)} = \underbrace{-\zeta^*\tilde{\mathbf{v}}_j(\tilde{t})}_{(2)} + \underbrace{\tilde{\mathbf{F}}_j}_{(3)} + \underbrace{\sqrt{\frac{2\zeta^*T^*}{\Delta\tilde{t}}}\mathbf{R}_G}_{(4)}$$

where  $\zeta^*$  is the dimensionless resistance coefficient  $\zeta^* = \frac{\zeta}{m} \sqrt{\frac{ma_0^2}{\epsilon}}$  and  $T^*$  is the dimensionless temperature  $T^* = \frac{k_B T}{\epsilon}$ .

## 2.2 Discretization of the equation: Euler method

The Langevin equation for the many-particle system introduced in this section is discretized by the Euler method as follows:

$$\tilde{\mathbf{v}}_j(\tilde{t} + \Delta\tilde{t}) = \tilde{\mathbf{v}}_j(\tilde{t}) - \zeta^* \tilde{\mathbf{v}}_j(\tilde{t}) \Delta\tilde{t} + \tilde{\mathbf{F}}_j \Delta\tilde{t} + \sqrt{2\zeta^* T^* \Delta\tilde{t}} \mathbf{R}_G.$$

## 2.3 Homework 3

Consider a two-dimensional binary particle mixture, composed of two species of particles. Set the particle size ratio to be  $a_0 : a_1 = 1 : 1.4$ , the particle number ratio to be  $N_1 : N_2 = 1 : 1$ , the particle number to be  $N = N_1 + N_2 = 300$ , and the particle number density to be  $\rho = \frac{N}{L^2} = 0.8$ . The motion of the particle  $j$  is driven by the following Langevin equation:

$$m \frac{d\mathbf{v}_j(t)}{dt} = -\zeta \mathbf{v}_j(t) + \mathbf{F}_j^I(t) + \mathbf{F}_j^B(t).$$

Here, the thermal fluctuation force acting on a particle satisfies the following fluctuation dissipation theorem :

$$\langle \mathbf{F}_B(t) \mathbf{F}_B(t') \rangle = 2k_B T \zeta \delta(t - t') \mathbf{1}.$$

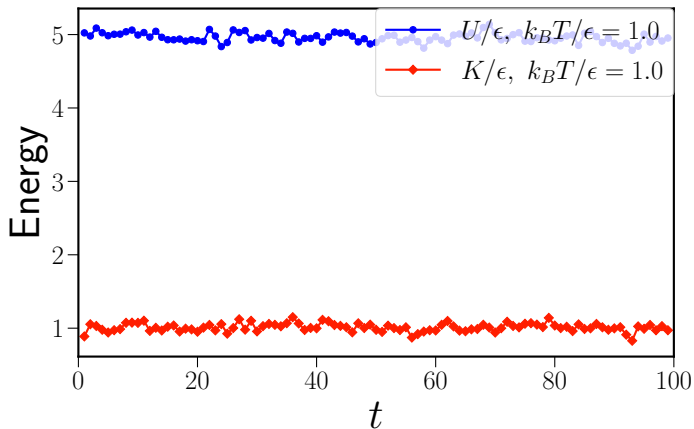
As the interparticle potential, use the following Bernu-Hansen-Hiwatari-Pastore (BHHP) potential,

$$U(r_{jk}) = \epsilon \left( \frac{a_{jk}}{r_{jk}} \right)^{12} + C_{jk} \quad (r_{jk} < a_{\text{cut}}).$$

The cut-off length is set to be  $a_{\text{cut}} = 3.0a_0$ . The interaction force is  $\mathbf{F}_j^I(t) = -\sum_k \frac{\partial U(r_{jk})}{\partial \mathbf{r}_j}$ . Now the time unit is given as  $t_0 = \sqrt{ma_0^2/\epsilon}$ , the length scale as  $a_0$ , and the temperature as  $k_B T/\epsilon = 1$ . Calculate the average kinetic energy

$$\bar{K} = \frac{1}{N} \sum_j \frac{1}{2} m \mathbf{v}_j^2,$$

every 100 steps. Consider also how the value of  $\Delta t$  changes the result.



**FIG 2:** The time evolutions of the potential energy  $U$  and the kinetic energy  $K$  obtained from the HW3 data at  $T = 1.0$ .

## 2.4 Radial distribution function

- The radial distribution function is the most frequently used analytical function for investigating the structure of liquids and solids.
- It is a two bodies density correlation function and is expressed as follows

$$g(\mathbf{r}) = \frac{1}{\rho N} \left\langle \sum_{j=1}^N \sum_{k=1}^N {}' \delta(\mathbf{r} - \mathbf{r}_j + \mathbf{r}_k) \right\rangle$$

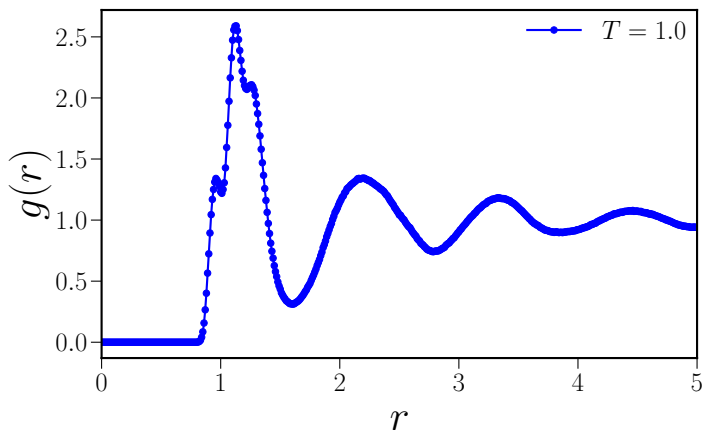
$\sum_{k=1}^N {}'$  means  $k = j$  is excluded.

- Now let us suppose the system is homogeneous and isotropic.
- Here, when  $N_j(r)$  is defined to be the number of other particles contained on a spherical shell at a distance  $r$  from the position of the particle  $j$ , the radial distribution function becomes as follows.

$$g(r) = \frac{1}{N} \left\langle \sum_{j=1}^N \frac{\Delta N_j(r)}{4\pi r^2 \Delta r \rho} \right\rangle$$

where  $\rho$  is the number density  $N/V$ .

## 2.4 Radial distribution function (2)



**FIG 3:** A radial distribution function  $g(r)$  obtained from the HW3 data at  $T = 1.0$ .

## 2.5 Static structure factor

- When examining the static structure of particles by scattering experiments, etc., Fourier transformed quantities are often used.
- The most basic function is the static structure factor

$$S(q) = \frac{1}{N} \langle \tilde{\rho}(q) \tilde{\rho}(-q) \rangle.$$

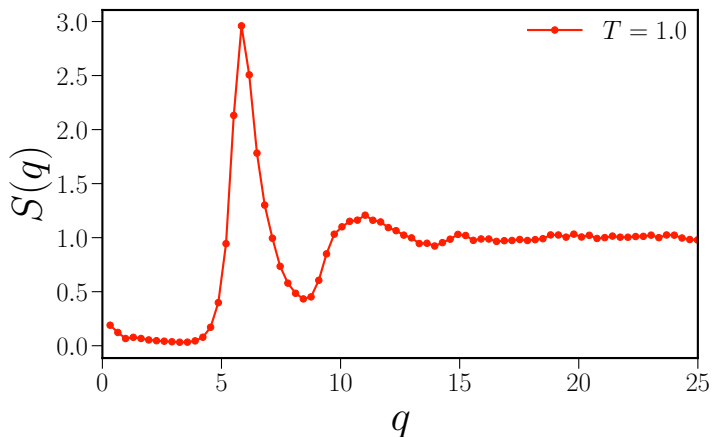
- $S(q)$  is the Fourier transform of  $g(\mathbf{r}) - 1$  in space.
- When calculating  $S(q)$  numerically, Fourier transformation of the density field  $\rho(\mathbf{r}) = \sum_{j=1}^N \delta(\mathbf{r} - \mathbf{r}_j)$  is treated as follows:

$$\tilde{\rho}(q) = \int d\mathbf{r} \rho(\mathbf{r}) e^{-i\mathbf{q} \cdot \mathbf{r}} = \sum_{j=1}^N e^{-i\mathbf{q} \cdot \mathbf{r}_j}$$

- And then we obtain  $S(q)$  as

$$S(q) = \frac{1}{N} \left\langle \sum_{j=1}^N e^{-i\mathbf{q} \cdot \mathbf{r}_j} \sum_{j=1}^N e^{i\mathbf{q} \cdot \mathbf{r}_j} \right\rangle = \frac{1}{N} \left\langle \left( \sum_{j=1}^N \cos(\mathbf{q} \cdot \mathbf{r}_j) \right)^2 + \left( \sum_{j=1}^N \sin(\mathbf{q} \cdot \mathbf{r}_j) \right)^2 \right\rangle$$

## 2.5 Static structure factor (2)



**FIG 4:** A static structure factor  $S(q)$  obtained from the HW3 data at  $T = 1.0$ .



## 2.6 Intermediate scattering function

- Next, I shall introduce the method for measuring the relaxation of the particle structure.
- The density field at time  $t$  is represented as  $\rho(\mathbf{r}) = \sum_{j=1}^N \delta(\mathbf{r} - \mathbf{r}_j)$ .
- As with the static structure factor, if we Fourier-transform the density field in space, we obtain  $\tilde{\rho}(\mathbf{q}, t) = \int d\mathbf{r} \rho(\mathbf{r}) e^{-i\mathbf{q} \cdot \mathbf{r}(t)} = \sum_{j=1}^N e^{-i\mathbf{q} \cdot \mathbf{r}_j(t)}$ .
- Here, the characteristic time in which the correlation of the Fourier transformed density field decreases with time is the structure relaxation time corresponding to the wave number  $q$ .
- Therefore, to measure the structural relaxation time, we introduce the following intermediate scattering function,  $F(q, t)$  such that

$$\begin{aligned}
 F(q, t) &= \frac{1}{N} \left\langle \sum_{j=1}^N e^{-i\mathbf{q} \cdot \mathbf{r}_j(t)} \sum_{j=1}^N e^{i\mathbf{q} \cdot \mathbf{r}_j(0)} \right\rangle \\
 &= \frac{1}{N} \left\langle \sum_{j=1}^N \cos(\mathbf{q} \cdot \mathbf{r}_j(t)) \sum_{j=1}^N \cos(\mathbf{q} \cdot \mathbf{r}_j(0)) + \sum_{j=1}^N \sin(\mathbf{q} \cdot \mathbf{r}_j(t)) \sum_{j=1}^N \sin(\mathbf{q} \cdot \mathbf{r}_j(0)) \right\rangle.
 \end{aligned}$$

- Here as with  $S(q)$ , Fourier transforms can be performed analytically.

## 2.6 Intermediate scattering function (2)

- The following self-intermediate scattering functions, i.e., taking only the self term of the intermediate scattering function, are commonly used:

$$\begin{aligned}F_s(q, t) &= \frac{1}{N} \left\langle \sum_{j=1}^N e^{-i\mathbf{q} \cdot (\mathbf{r}_j(t) - \mathbf{r}_j(0))} \right\rangle \\&= \frac{1}{N} \left\langle \sum_{j=1}^N \cos(\mathbf{q} \cdot \mathbf{r}_j(t)) \cos(\mathbf{q} \cdot \mathbf{r}_j(0)) + \sum_{j=1}^N \sin(\mathbf{q} \cdot \mathbf{r}_j(t)) \sin(\mathbf{q} \cdot \mathbf{r}_j(0)) \right\rangle\end{aligned}$$

- Especially, in the simple liquid in which the cooperativity of particle motion, etc. is not often observed, the result is almost the same as the whole intermediate scattering function. The first peak of the static structure factor  $q = 2\pi/a$  ( $a$  is the diameter of the particle) is usually used as the wavenumber to study the structural relaxation of liquids.

## 2.7 Logarithmic sampling

- When calculating the time correlation function, it is better to store the data by the time interval in geometric series (Useful for analysis over a wide range of time).
- After sampling for a fixed time, the same sampling is repeated again, and the time average can be taken.

Listing 3: logarithmic sampling

```
// add in the time development loop:
if(int(t/dt) == int((sampling_time + time_stamp)/dt)){
    output_t(x,y,avU,avK,Np,t,time_stamp,x_corr,y_corr);
    sampling_time*=pow(10.,0.1);
    sampling_time=int(sampling_time/dt)*dt;
    if(sampling_time > sampling_time_max/pow(10.,0.1)){
        time_stamp=t;
        sampling_time=5.*dt;
    }
}
```

## 2.8 Center of mass correction

- Under the periodic boundary condition of finite size, a motion called Goldstone mode occurs in which **all** particles move together.
- Such motion disappears in the thermodynamic limit of infinite system size.
- It affects the calculation of the density relaxation of the particles.
- Even if the structure does not change much, the particles appear to be relaxed since they move all together.
- By subtracting the center of mass motion, “genuine structural relaxation” can be captured.

Listing 4: Center of mass correction

```
int com_correction(double *x, double *y, double *x_corr, double *y_corr, int Np,
    double L){
    int i;
    double dx, dy;
    static double x0[Npm], y0[Npm];
    static bool IsFirst = true;
    if(IsFirst){
        for(i=0; i<Np; i++){
            x0[i]=x[i];
            y0[i]=y[i];
        }
    }
}
```

## 2.8 Center of mass correction (2)

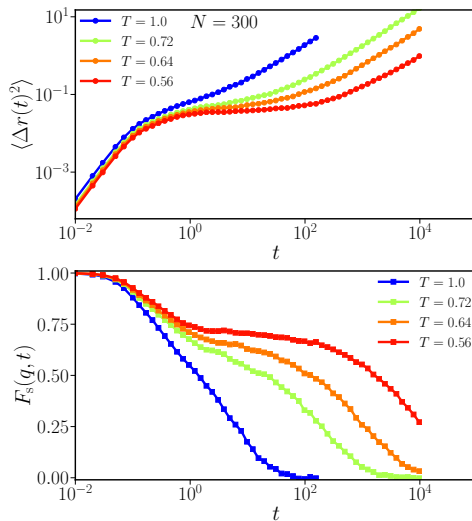
```
    }  
    IsFirst = false;  
}  
  
for(i=0;i<Np;i++){  
    dx=x[i]-x0[i];  
    if(dx>0.5*L) dx-=L;  
    else if(dx<-0.5*L) dx+=L;  
  
    dy=y[i]-y0[i];  
    if(dy>0.5*L) dy-=L;  
    else if(dy<-0.5*L) dy+=L;  
  
    *x_corr+=dx/Np; //center of mass displacement.x  
    *y_corr+=dy/Np;  
  
    x0[i]=x[i];  
    y0[i]=y[i];  
}  
return 0;  
}
```

## 2.9 Homework 4

### Homework 4

By following the setting of Homework 3, evaluate the physical properties when the temperature is changed. In particular compute the radial distribution function, the static structure factor, the mean-square displacement, and the intermediate scattering function at temperatures  $T = 1.0, 0.72, 0.64, 0.56$ .

## 2.9 Homework 4 (2)



## 2.9 Homework 4 (3)

**FIG 5:** The mean-square displacement, and the intermediate scattering function at temperatures  $T = 1.0, 0.72, 0.64, 0.56$ .



## Section 3

### Speedup technique

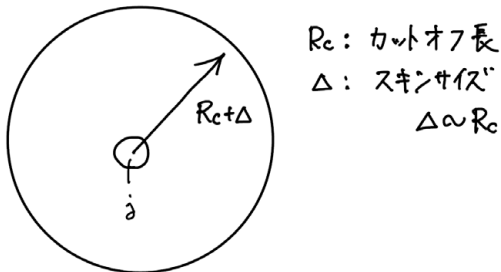
## 3.1 Introduction of the speedup technique

- This chapter describes how to speed up the force calculation, the most computationally expensive part of molecular dynamics.
- In this case, it is sufficient to calculate only the forces in the neighbour of each particle.
- The list method is an algorithm that specifies the neighbour of each particle.

## 3.2 Verlet list method

- Let us construct a list of the nearest neighbor particles  $k$  of particle  $j$  as shown in the Fig. 6.
- The most basic method to achieve this, the Verlet list method, involves first calculating the distances of all particles and then searching for particles  $k$  that lie within  $R_c + \Delta$  from particle  $j$ .
- Here  $R_c$  is the cut-off length and  $\Delta$  is the skin size.
- Once the neighbor list is determined, the same list is used for a while (for  $N_{\text{list}}$  steps) to calculate the interaction.
- In doing so, we need to find the best combination for the relationship between  $\Delta$  and  $N_{\text{list}}$ . In this way the computational cost is reduced roughly from  $O(N^2)$  to  $O(N^{5/3})$ .

## 3.2 Velret list method (2)



**FIG 6:** Make a list for the nearest particle  $k$  of the particle  $j$ , using  $R_c$  as the cutoff length and  $\Delta$  as the skin size.

Here construct following neighbor lists.

## 3.2 Velret list method (3)

### Listing 5: neighbor list

```
list[j][0]  The number of the particle j
list[j][1]  ID of the 1st neighbor of the particle j
list[j][2]  ID of the 2nd neighbor of the particle j

list[j][k]  ID of the kth neighbor of the particle j
```

### Listing 6: Verlet list method

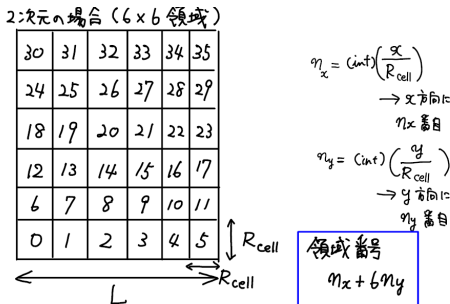
```
int Verlet_list(double *x, double *y, int (*list)[Ndm], double L, int Np)
{
    int i, j;
    int n;
    double r;
    double dx, dy;
    double cut=4.0; // cut off + skin size
    for(i=0; i<Np; i++)
    {
        n=0;
        for(j=0; j<Np; j++)
            if(i>j)
            {
                dx=x[i]-x[j];
                dy=y[i]-y[j];
```

## 3.2 Velret list method (4)

```
if(dx>L/2.)
    dx-=L;
if(dx<-L/2.)
    dx+=L;
if(dy> L/2.)
    dy-=L;
if(dy<-L/2.)
    dy+=L;
r=dx*dx+dy*dy;
if(r<cut*cut)
{
    n++;
    list[i][n]=j;
}
list[i][0]=n;
}
return 0;
}
```

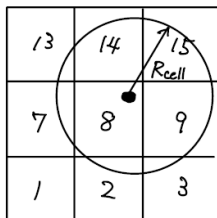
### 3.3 Cell list method

- In the Verlet list method, it was necessary to carry out the adjacency judgment for all particles.
- On the other hand, the cell list method introduced below is more efficient than the Verlet list method because it does not need to refer to all particles when obtaining the list.



**FIG 7:** Step 1) Divide the entire system into small areas. Set the region number as a one-dimensional array.

### 3.3 Cell list method (2)



Verlet list 法の時と  
同様に 配列

$\text{list}[j][0]$

$\text{list}[j][1]$

$\text{list}[j][2]$

$\vdots$

と作成する。

**FIG 8:** (Step 2) The Verlet list is created for the radius  $R_{\text{cell}}$  for the middle region including the first neighbor region.



## 3.3 Cell list method (3)

### Listing 7: Cell list method

```
int f(int i,int M)
{
    int k;

    k=i;

    if(k<0)
        k+=M;
    if(k>=M)
        k-=M;

    return k;
}

void update(double L,int Np,double *x,double *y,int M,double Rcell,int (*list)[Pm])
{
    int i,j,k;
    int nx,ny;
    int l,m;
    double dx,dy,r;

    int (*map)[Npm]=new int[M*M][Npm];
```

### 3.3 Cell list method (4)

```

for(i=0;i<M;i++)
  for(j=0;j<M;j++)
    map[i+M*j][0]=0;

for(i=0;i<Np;i++){
  nx=f((int)(x[i]/Rcell),M);
  ny=f((int)(y[i]/Rcell),M);

  for(m=ny-1;m<=ny+1;m++){
    for(l=nx-1;l<=nx+1;l++){
      map[f(l,M)+M*f(m,M)][map[f(l,M)+M*f(m,M)][0]+1]=i;
      map[f(l,M)+M*f(m,M)][0]++;
      //^^Iprintf("%d\n", map[f(l,M)+M*f(m,M)][0]);
    }
  }
}

for(i=0;i<Np;i++){
  list[i][0]=0;
  nx = f((int)(x[i]/Rcell),M);
  ny = f((int)(y[i]/Rcell),M);

  for (k=1; k<=(map[nx+M*ny][0]); k++){
    j = map[nx+M*ny][k];
    if(j>i){
      dx =x[i] - x[j];

```

### 3.3 Cell list method (5)

```

    dy = y[i] - y[j];

    if(dx < -L/2.0)
        dx += L;
    else if(dx > L/2.0)
        dx -= L;

    if(dy < -L/2.0)
        dy += L;
    else if(dy > L/2.0)
        dy -= L;
    r = dx*dx + dy*dy;

    if(r < Rcell*Rcell){
        list[i][0]++;
        list[i][list[i][0]] = j;
    }
}
}
delete []map;
}

int main(void){
    .....
    double RCHK=4.5;

```

## 3.3 Cell list method (6)

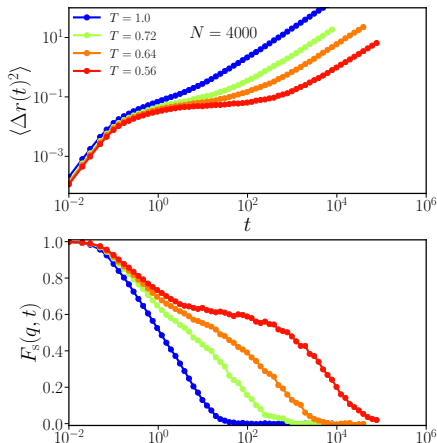
```
double L = sqrt(double(Np) / 0.8);  
int M=(int)(L/RCHK);  
double Rcell = L/(double)M;  
.....  
.....  
    update(L,Np,x,y,M,Rcell,list);  
.....  
}
```

## 3.4 Homework 5

### Homework 5

By following the setting of Homework 3 but the particle number  $N = 4000$ , evaluate the physical properties when the temperature is changed. In particular compute the radial distribution function, the static structure factor, the mean-square displacement, and the intermediate scattering function at temperatures  $T = 1.0, 0.72, 0.64, 0.56$ .

## 3.4 Homework 5 (2)



**FIG 9:** The mean-square displacement, and the intermediate scattering function at temperatures  $T = 1.0, 0.72, 0.64, 0.56$  at  $N = 4000$ .