

# PHYS580 Lab12 Report

Yicheng Feng  
PUID: 0030193826

November 17, 2019

**Workflow:** I use the Linux system, and code C++ in terminals. For visualization, I use the C++ package – ROOT (made by CERN) to make plots. At the end, the reports are written in L<sup>A</sup>T<sub>E</sub>X.

The codes for this lab are written as the following files:

- `molecular_dynamics_2d.h` and `molecular_dynamics_2d.cxx` for the class `IsingModel2D` to simulate the 2D molecular dynamics.
- `lab12.cxx` for the main function to make plots.

To each problem of this lab report, I will attach the relevant parts of the code. If you want to check the validation of my code, you need to download the whole code from the link <https://github.com/YichengFeng/phys580/tree/master/lab12>.

- (1) Use the starter programs (or your own equivalent ones) to simulate a system of, say, 25 particles in a square of side length 5 (in units of  $\sigma$ , the Lennard-Jones parameter). Initially, let the particles be at rest but with a relatively small random variation in position from the evenly spaced, square lattice vertices. (Why give the variation?) Then, as the simulation proceeds, produce images of the time evolution of particle positions similar to those displayed in Fig. 9.6 of the textbook. Also reproduce the time series of the total energy, temperature, tagged particle and tagged pair separations. Are the fluctuations in energy and temperature, and the trend of the pair separation as you expect, and why?

### Physics explanation:

We follow the suggested configuration for the simulation: 25 particles in a 2D  $5 \times 5$  box. The maximum initial position variation from the grid is set to be  $0.05\sigma$ . The initial velocity of each particle is 0. The time step is  $\Delta t = 0.005$ , and we record the information every 10 steps.

*Why give the variation?* If the particles are perfectly even placed and at rest at the beginning, the total force on each particle is zero, so they will not move.

Figure 1 shows the evolutions of particle positions in various time ranges: left pad  $0 < t < 0.1$ ; middle pad  $0.2 < t < 4.0$ ; right pad  $13 < t < 16$ . We can see the square grid at the beginning, and triangle grid in equilibrium.

Figure 2 shows the evolutions of total energy  $E$  (left pad) and temperature  $T$  (right pad). They are relatively very stable after  $t = 0$ .

Figure 3 shows the evolutions of tagged particle displacement (left pad) and temperature tagged pair distance (right pad). The displacement of the tagged particle fluctuates while the distance of the tagged pair is stable around 1, so the fluctuation should be more likely a global behavior than individual.

The fluctuations in energy and temperature are as expected. The system is small and particles are close, so it should behave like solid. Therefore, the equilibrium should be quickly achieved (solid transports heat faster than gas), and then the temperature should be stable with small fluctuation after  $t = 0$ . The energy should be constant, but it is not perfectly constant here, because we apply the periodic condition and take the closest distance ( $< 0.5L$ ). Thus, it fluctuates a little bit.

The trends of tagged particle displacement and tagged pair distance are also reasonable. The initial position is not equilibrium, so the tagged particle will quickly go to the equilibrium position at the beginning. Then, it should fluctuates around that position, so the average displacement from the initial position should be obviously nonzero. As the system is small and like solid, the particles should not change position with each other, but instead vibrate around the stable positions, so the distance of the tagged pair is stable around 1.

## Plots:

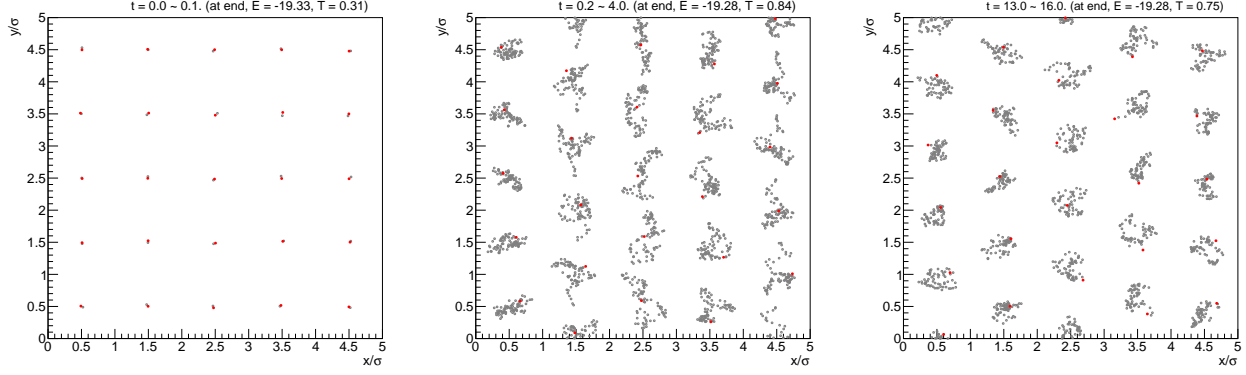


Figure 1: The evolutions of particle positions are shown above in various time ranges: left pad  $0 < t < 0.1$ ; middle pad  $0.2 < t < 4.0$ ; right pad  $13 < t < 16$ . The red dots are the final position of particles of each time range.

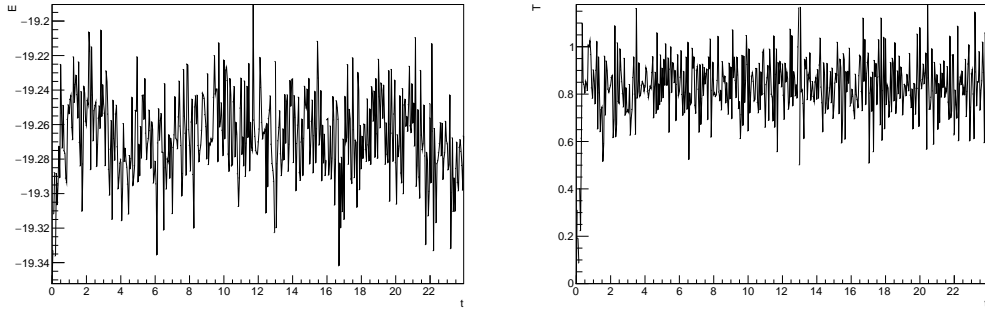


Figure 2: The evolutions of total energy  $E$  (left pad) and temperature  $T$  (right pad) are shown above.

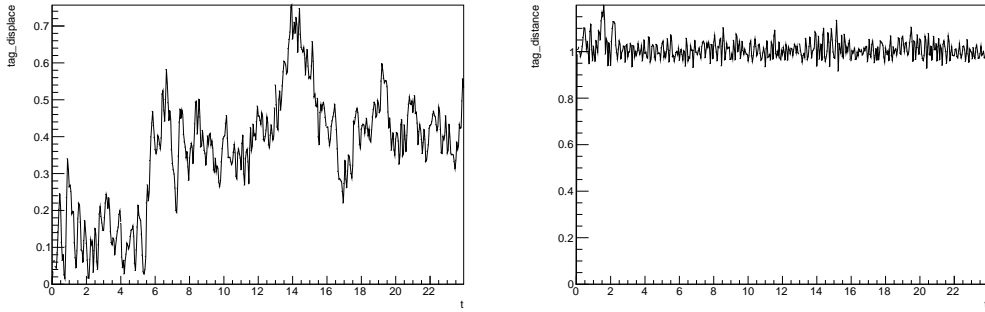


Figure 3: The evolutions of tagged particle displacement (left pad) and temperature tagged pair distance (right pad) are shown above.

## Plots:

For the initialization of the simulation

```
1  //-----//
2
3  MolecularDynamics2D::MolecularDynamics2D(int N, double L, double dt, double dmax, double vmax) {
4
5      _N = N;
6      _rec = 10;
7      _L = L;
8      _dt = dt;
9      _dmax = dmax;
10     _vmax = vmax;
11     _t_now = 0;
12
13     _tag_displace_now = 0;
14     _tag_distance_now = 0;
15
16     _x.clear();
17     _y.clear();
18     _vx.clear();
19     _vy.clear();
20
21     for(int i=0; i<_N; i++) {
22         _x_old.push_back(0);
23         _x_now.push_back(0);
24         _x_new.push_back(0);
25         _y_old.push_back(0);
26         _y_now.push_back(0);
27         _y_new.push_back(0);
28
29         _vx_now.push_back(0);
30         _vy_now.push_back(0);
31     }
32
33     double sqN = sqrt(_N);
34     double grid = sqN==floor(sqN)?_L/sqN:_L/(sqN+1);
35
36     int n = 0;
37     double i = 0;
38     while(i < _L) {
39         double j = 0;
40         while(j < _L) {
41             if(n >= _N) break;
42             _x_now[n] = i + 0.5*grid + _dmax*(1.0*rand()/RAND_MAX-0.5)*grid*sqrt(2);
43             _y_now[n] = j + 0.5*grid + _dmax*(1.0*rand()/RAND_MAX-0.5)*grid*sqrt(2);
44             _vx_now[n] = _vmax*(1.0*rand()/RAND_MAX-0.5)*sqrt(2);
45             _vy_now[n] = _vmax*(1.0*rand()/RAND_MAX-0.5)*sqrt(2);
46             _x_old[n] = _x_now[n] - _vx_now[n]*_dt;
47             _y_old[n] = _y_now[n] - _vy_now[n]*_dt;
48
49             n ++;
50             j += grid;
51         }
52         i += grid;
53     }
54     _x_start = _x_old;
55     _y_start = _y_old;
56
57     cout << "initialization completed" << endl;
```

```

58 }
59
60 //-----//

```

For the simulation process

```

1 //-----//
2
3 void MolecularDynamics2D::cal_once() {
4
5     for(int i=0; i<_N; i++) {
6         double fx = 0;
7         double fy = 0;
8
9         for(int j=0; j<_N; j++) {
10             if(j == i) continue;
11
12             double dx = _x_now[i] - _x_now[j];
13             double dy = _y_now[i] - _y_now[j];
14
15             if(fabs(dx) > 0.5*_L) dx -= dx>0?_L:-_L;
16             if(fabs(dy) > 0.5*_L) dy -= dy>0?_L:-_L;
17
18             double r = sqrt(dx*dx + dy*dy);
19             if(r < 3) {
20                 double fij = 24.0*(2.0/pow(r,13) - 1.0/pow(r,7));
21                 fx += fij*dx/r;
22                 fy += fij*dy/r;
23             }
24         }
25
26         _x_new[i] = 2*_x_now[i] - _x_old[i] + fx*_dt*_dt;
27         _y_new[i] = 2*_y_now[i] - _y_old[i] + fy*_dt*_dt;
28         _vx_new[i] = (_x_new[i] - _x_old[i])/(2.0*_dt);
29         _vy_new[i] = (_y_new[i] - _y_old[i])/(2.0*_dt);
30
31         if(i == 0) {
32             double dx = _x_new[i] - _x_start[i];
33             double dy = _y_new[i] - _y_start[i];
34             if(fabs(dx) > 0.5*_L) dx -= dx>0?_L:-_L;
35             if(fabs(dy) > 0.5*_L) dy -= dy>0?_L:-_L;
36             _tag_displace_now = sqrt(dx*dx + dy*dy);
37         } else if(i == 1) {
38             double dx = _x_new[i] - _x_new[0];
39             double dy = _y_new[i] - _y_new[0];
40             if(fabs(dx) > 0.5*_L) dx -= dx>0?_L:-_L;
41             if(fabs(dy) > 0.5*_L) dy -= dy>0?_L:-_L;
42             _tag_distance_now = sqrt(dx*dx + dy*dy);
43         }
44
45         if(_x_new[i]<0) {
46             _x_new[i] += _L;
47             _x_now[i] += _L;
48         } else if(_x_new[i]>_L) {
49             _x_new[i] -= _L;
50             _x_now[i] -= _L;
51         }
52         if(_y_new[i]<0) {
53             _y_new[i] += _L;
54             _y_now[i] += _L;
55         } else if(_y_new[i]>_L) {

```

```

56         _y_new[i] -= _L;
57         _y_now[i] -= _L;
58     }
59 }
60
61     _x_old = _x_now;
62     _x_new = _x_new;
63     _y_old = _y_now;
64     _y_new = _y_new;
65 }
66
67 //-----//
68
69 void MolecularDynamics2D::cal_ET() {
70
71     double Ekin = 0;
72     double Epot = 0;
73
74     for(int i=0; i<_N; i++) {
75         Ekin += 0.5*(_vx_now[i]*_vx_now[i] + _vy_now[i]*_vy_now[i]);
76
77         for(int j=i+1; j<_N; j++) {
78
79             double dx = _x_old[j] - _x_old[i];
80             double dy = _y_old[j] - _y_old[i];
81
82             if(fabs(dx) > 0.5*_L) dx -= dx>0?_L:-_L;
83             if(fabs(dy) > 0.5*_L) dy -= dy>0?_L:-_L;
84
85             double r = sqrt(dx*dx + dy*dy);
86             if(r < 3) {
87                 double invr6 = 1.0/pow(r,6);
88                 double invr12 = invr6*invr6;
89                 Epot += 4*(invr12 - invr6);
90             }
91         }
92     }
93
94     _E_now = Ekin + Epot;
95     _T_now = Ekin/_N;
96 }
97
98 //-----//
99
100 void MolecularDynamics2D::cal_until(double t_end) {
101
102     if(!check()) {
103         cout << "ERROR: check() not pass!" << endl;
104         return;
105     }
106     cout << "check passed" << endl;
107
108     _t.clear();
109     _x.clear();
110     _y.clear();
111     _vx.clear();
112     _vy.clear();
113     _tag_displace.clear();
114     _tag_distance.clear();
115

```

```

116     _E.clear();
117     _T.clear();
118
119     int n = 0;
120
121     while(_t_now < t_end) {
122         if(n%_rec == 0) {
123             _t.push_back(_t_now);
124             _x.push_back(_x_now);
125             _y.push_back(_y_now);
126             _vx.push_back(_vx_now);
127             _vy.push_back(_vy_now);
128             _tag_displace.push_back(_tag_displace_now);
129             _tag_distance.push_back(_tag_distance_now);
130
131             cal_ET();
132             _E.push_back(_E_now);
133             _T.push_back(_T_now);
134         }
135
136         cal_once();
137
138         n++;
139         _t_now += _dt;
140     }
141 }
142
143 //-----//

```

- (2) How can you speed up the convergence to equilibrium you observed in (1)? Find out a way to do so using the feature of the starter program that allows one to change the kinetic energy of the particles via keyboard input during the simulation. Similarly, when you have attained a stable triangular arrangement of the particles (solid), find a way to melt it by heating it. Demonstrate that you succeeded in melting the crystal by making appropriate plots of the particle arrangements and the time series of various functions.

Note: What happens if the time step is too large (small) or if you raise the temperature too much, and why? Address these in your discussion.

### Physics explanation:

To speed up the convergence to equilibrium, we can decrease the temperature by reducing the velocity. The configuration is the same as that in (1), but we reduce the velocity by 50% at  $t = 4$ .

Figure 4 shows the evolution of particle positions. Before the change of temperature ( $t < 4$ ), the evolution (left pad and middle pad) is almost the same as that in Fig. 1. After the temperature decrease ( $t > 4$ ), the right pad shows more regular position with smaller fluctuation than that in Fig. 1.

Figure 5 shows the evolutions of total energy  $E$  (left pad) and temperature  $T$  (right pad). The temperature fluctuation is smaller after turning down the temperature.

Figure 6 shows the evolutions of tagged particle displacement (left pad) and temperature tagged pair distance (right pad).

To melt down the solid crystal, we can increase the temperature by increasing the velocity. The configuration is the same as that in (1), but we increase the velocity by 400% at  $t = 13$ . (It should have arrived the equilibrium by  $t = 13$ .)

We can see from the right pad of Fig. 7 that the lattice configuration is destroyed. In Fig. 9, the tagged particle is far away from its initial position after increasing the temperature. The value 2.5 is half of the edge  $L = 5$ . The distance of the tagged pair also becomes larger than 1 after  $t = 13$ . It goes to 3 with large fluctuation, which again means the lattice is melt.

*Note: What happens if the time step is too large (small) or if you raise the temperature too much, and why? Address these in your discussion.*

If the time step is too small, it simulation would be slow, but the results would become better.

If the time step is too large or the temperature is raised too much, the infinitely large energy and temperature might happen. In each step, the interaction from molecular potential is calculated from the starting point, so it might be relatively too small to prevent particles going close to each other. With large time step or high velocity, the particles go longer in each step, so the approximation of force becomes worse and there would be more change for two particle to go too close. Once it happens, the energy will be increased dramatically, and then the kinetic energy will also be increased by the huge force. Thus, both the energy and temperature could be increased to an abnormal level.



Plots:

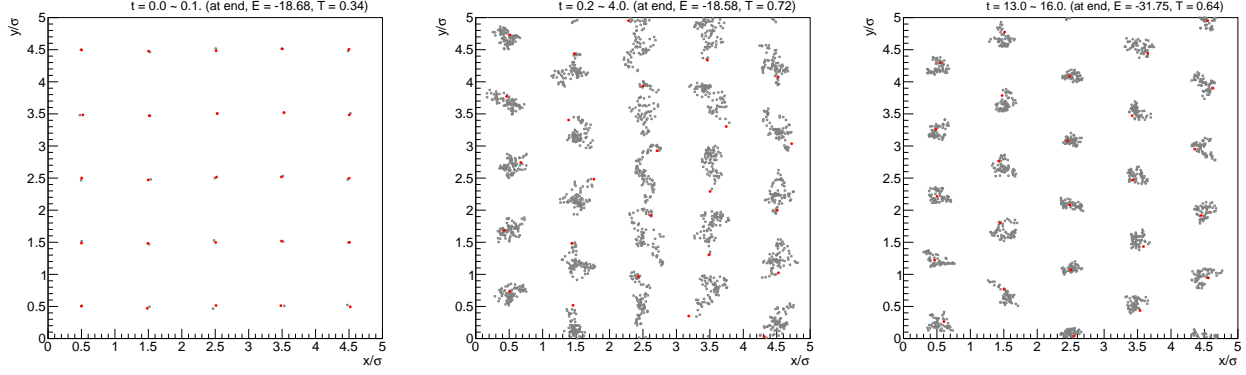


Figure 4: The evolutions of particle positions are shown above in various time ranges: left pad  $0 < t < 0.1$ ; middle pad  $0.2 < t < 4.0$ ; right pad  $13 < t < 16$ . The red dots are the final position of particles of each time range. **The velocity is reduced by 50% at  $t = 4$ .**

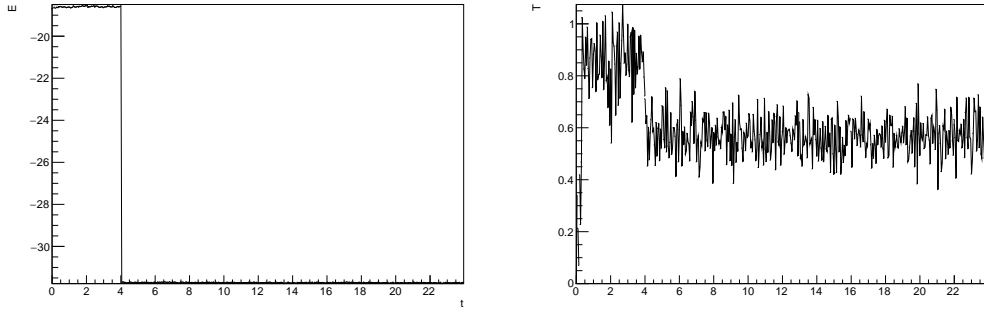


Figure 5: The evolutions of total energy  $E$  (left pad) and temperature  $T$  (right pad) are shown above. **The velocity is reduced by 50% at  $t = 4$ .**

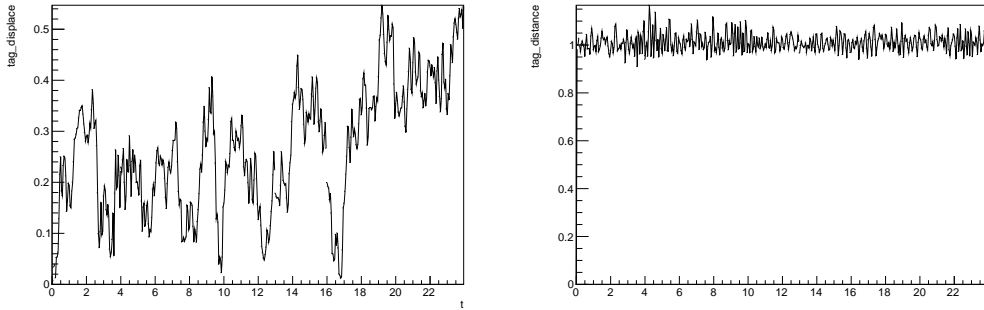


Figure 6: The evolutions of tagged particle displacement (left pad) and temperature tagged pair distance (right pad) are shown above. **The velocity is reduced by 50% at  $t = 4$ .**

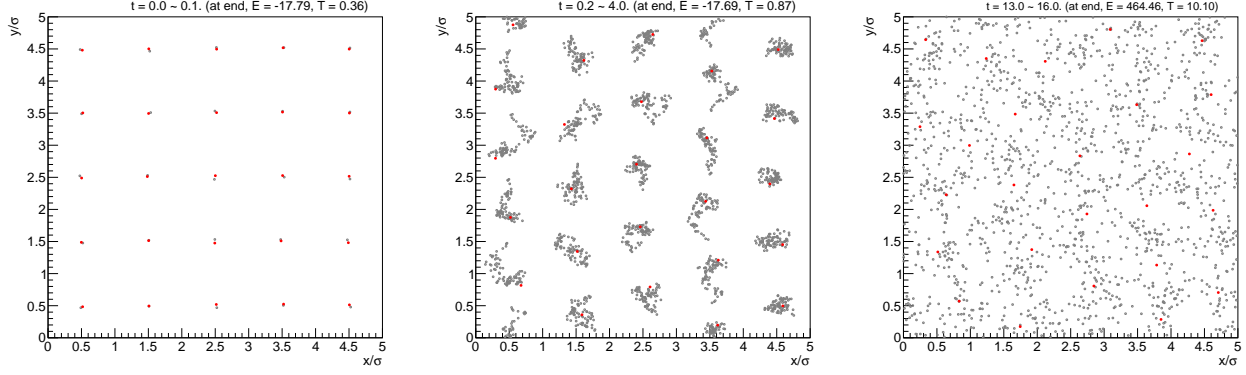


Figure 7: The evolutions of particle positions are shown above in various time ranges: left pad  $0 < t < 0.1$ ; middle pad  $0.2 < t < 4.0$ ; right pad  $13 < t < 16$ . The red dots are the final position of particles of each time range. **The velocity is increased by 400% at  $t = 13$ .**

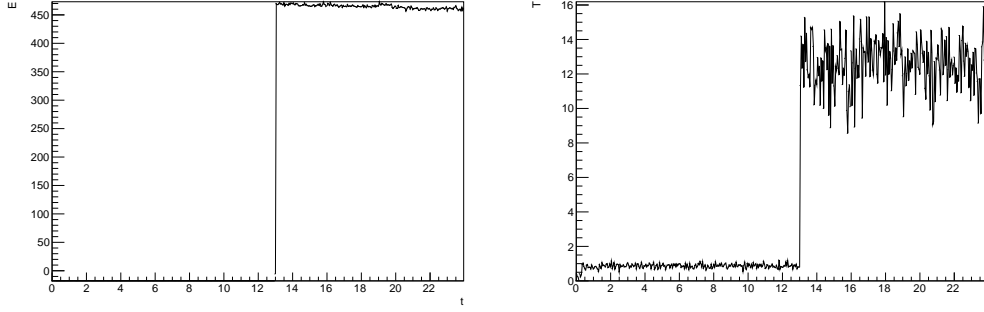


Figure 8: The evolutions of total energy  $E$  (left pad) and temperature  $T$  (right pad) are shown above. **The velocity is increased by 400% at  $t = 13$ .**

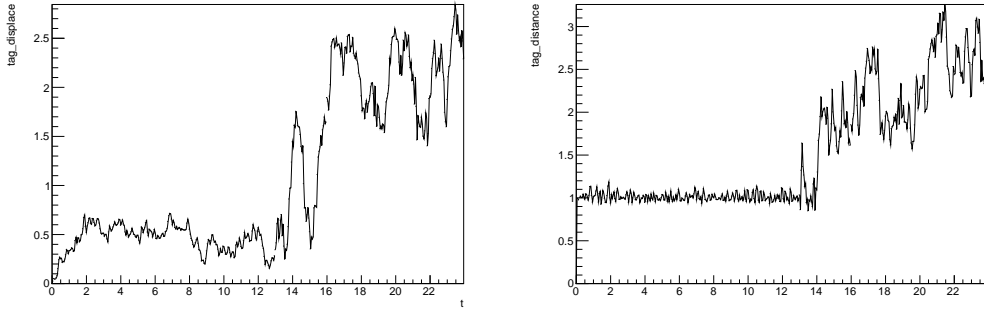


Figure 9: The evolutions of tagged particle displacement (left pad) and temperature tagged pair distance (right pad) are shown above. **The velocity is increased by 400% at  $t = 13$ .**

**Relevant code:**

For the simulation, the code has already shown in problem (1).

For the tuning of velocity (temperature)

```
1  //-----//
2
3  void MolecularDynamics2D::change_velocity(double factor) {
4
5      for(int i=0; i<_N; i++) {
6          _x_old[i] = _x_now[i] - factor*(_x_now[i] - _x_old[i]);
7          _y_old[i] = _y_now[i] - factor*(_y_now[i] - _y_old[i]);
8      }
9  }
10
11 //-----//
```

- (3) Study the effect of varying the density, initial velocities and/or positions on the approach to equilibrium and the nature of the final equilibrium configuration. You do not need to be exhaustive on this. For example, try putting 25 particles in a square of side length 10, and see how their characteristics change as you vary the temperature, substantiating your discussion using various time series graphs.

### Physics explanation:

The effects of the initial velocity and position are somehow equivalent in simulation, so we only explore one of them (position in  $L = 5$ , velocity in  $L = 10$ ). We try 3 different maximum initial displacements:  $0.02\sigma$ ,  $0.05\sigma$  and  $0.10\sigma$  with lattice size  $L = 5$ ; and 3 different maximum initial velocity 2, 5, and 10 with lattice size  $L = 10$ .

With  $L = 5$ , Fig. 10 shows the evolutions of tagged pair distance with various maximum initial displacement: left pad  $0.02\sigma$ ; middle pad  $0.05\sigma$ ; right pad  $0.10\sigma$ ; Fig. 11 shows the evolutions of particle positions with various maximum initial displacement. We can see with larger initial displacement, it takes longer to achieve the equilibrium, or the equilibrium has more fluctuation.

With  $L = 10$ , it becomes a totally different story – the lattice structure disappears, because there is more space and the potential vanishes with large distance. In this case, I think it doesn't have the lattice equilibrium, but gas instead. Due to the large distance, the initial displacement cannot change the temperature much. However, the initial velocity still affects. The temperature increases along with the increasing initial maximum velocity. When the velocity is high (temperature is high), the trajectories of particles cover the space more evenly, and the distance of the tagged pair fluctuates more frequently, which indicates that the equilibrium of gas is achieved more quickly. **The gap equilibrium pattern is opposite to the solid lattice equilibrium.**

### Plots:

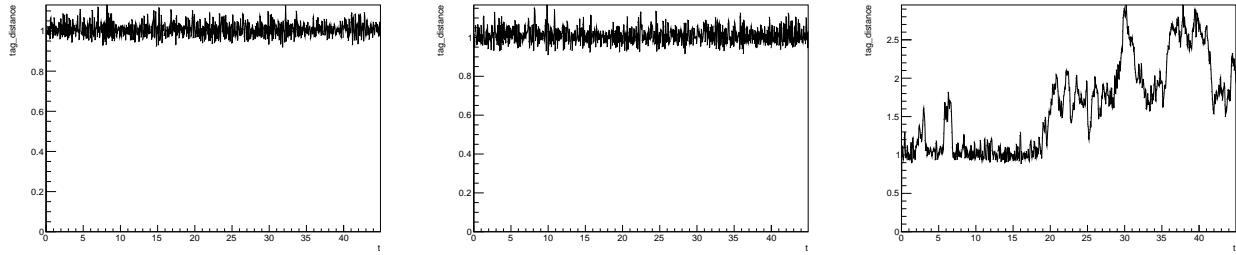


Figure 10: ( $L = 5$ ) The evolutions of tagged pair distance depend on time with various maximum initial displacement: left pad  $0.02\sigma$ ; middle pad  $0.05\sigma$ ; right pad  $0.10\sigma$ .

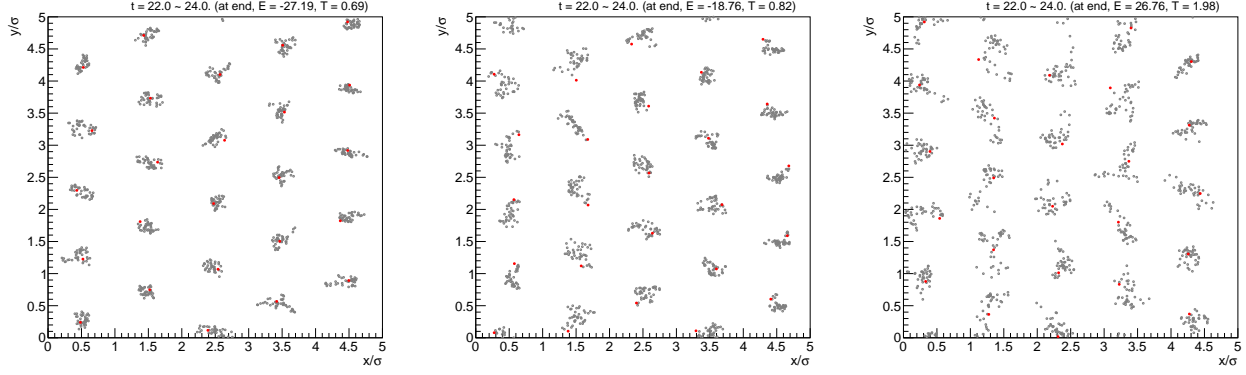


Figure 11: ( $L = 5$ ) The evolutions of tagged pair distance depend on time with various maximum initial displacement: left pad  $0.02\sigma$ ; middle pad  $0.05\sigma$ ; right pad  $0.10\sigma$ . Time range is  $22 < t < 24$ .

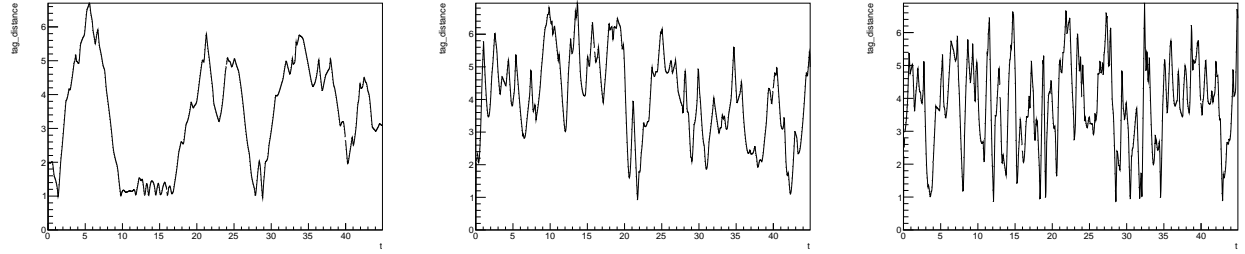


Figure 12: ( $L = 10$ ) The evolutions of tagged pair distance depend on time with various maximum initial velocity: left pad 2; middle pad 5; right pad 10.

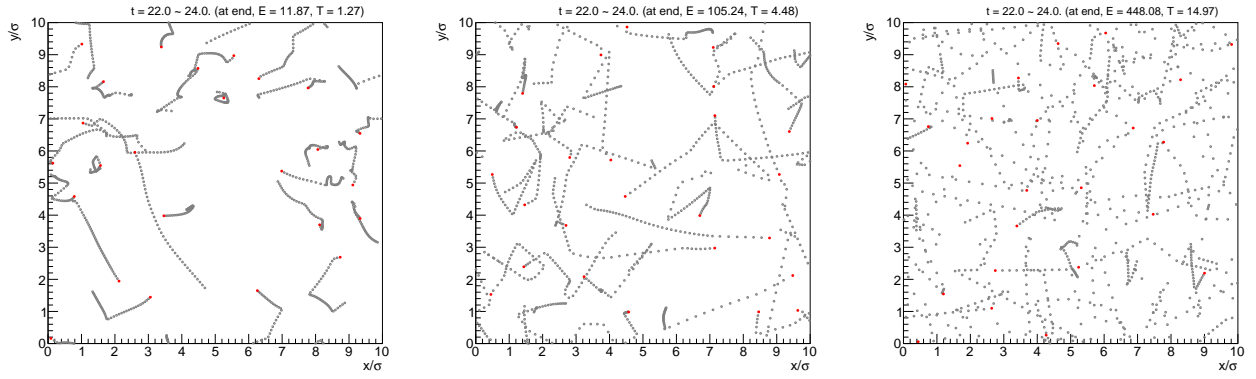


Figure 13: ( $L = 10$ ) The evolutions of tagged pair distance depend on time with various maximum initial displacement: left pad 2; middle pad 5; right pad 10. Time range is  $22 < t < 24$ .

#### Relevant code:

For the simulation, the code has already been shown in problem (1).