PHYS2020 Computational Project

April 2022

Part 1 - Simulating a non-interacting ideal gas

Introduction

An ideal gas is a model of a gas that makes a few key assumptions that allow us to more easily solve problems involving gasses. It is undoubtedly extremely useful, but as with any model it comes with a caveat, which is that real gasses don't behave exactly like ideal gasses, only approximately [1]. However, with a simulation, we can create a gas that is ideal. In this part of the report, I aim to develop a gas simulation to experimentally confirm the ideal gas law.

Background

Ideal gasses

An ideal gasses can be characterised by a few key parameters, these being the temperature T, pressure P, volume V and finally number of particles N. These parameters are related by the ideal gas law, where k is the Boltzmann constant [1]:

$$PV = NkT \tag{1}$$

The simulation will need to either measure or control these parameters. This simulation will be in 2 dimensions, which means that the volume and pressure would be defined differently to our typical understanding in 3 dimensions. We can consider the area as the 2 dimensional analogue of volume. For pressure, we will consider force applied per length, instead of force applied per area.

Simulating particles in a box

For the simulation, particles are represented as points on a square L x L region, where L is the length of the box, which I can change. The volume therefore is L^2. I can control the mass, temperature, and number of particles as well, which allows me to perform experiments on the gas by changing these quantities, and holding the rest constant. At the beginning of an experiment, all particles are given a random x and y velocity sampled from a normal distribution scaled by $\sqrt{kT/m}$.

```python
for t in np.arange(N_time):

        pos +=  vel * dt # numerical integration of velocity to update the position of particles
        J[t] = 2 * m * np.sum(vel[pos >= L]) # Searching for any particles which have hit the right side of the box, then calculating and storing the total impulse from them
        vel = np.where(pos >= L, -vel, vel) # Checks if particle is outside the box on the right and top,if so reversing the direction
        vel = np.where(pos <= 0, -vel, vel) # Checks if the particle is outside the box on the left and bottom , if so reversing the direction
```

As can be seen in the above code, at each time step, the code numerically integrate the velocity to update the position of the particles. I prevent particles from escaping the box by simulating elastic collisions at the boundaries. Upon touching the boundary of the box, the particle's x or y velocity is reversed to prevent the particle escaping. Finally, the pressure is determined through measuring the impulse imparted by particles onto a particular wall, then dividing this impulse by the length of the wall. The impulse over time of the system is shown in the below figure: As can be seen, the graph is linear and increasing over time. This is expected behaviour. With 100000 particles, there are sufficiently collisions per second for the impulse imparted by each collision to appear continuous. On average, particles have the same magnitude of velocity given by $\sqrt{kT/m}$, so over a large number of collisions, the impulse adds linearly.

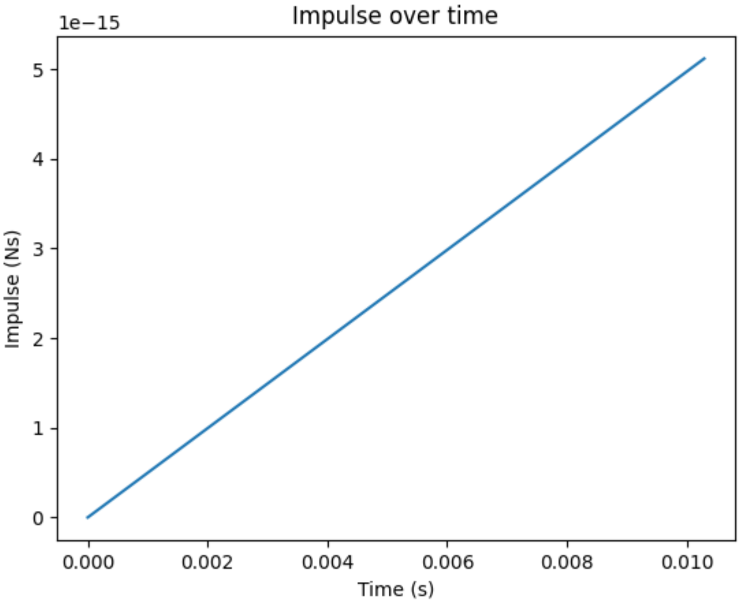

Figure 1: Impulse over time for Trial 1

Theory

With the functionality of the code established, I can now begin to perform some experiments with the code. I will test the ideal gas relationship between pressure and volume, holding N and T constant. To test whether the relationship holds, I will linearise the ideal gas law.

$$PV = NkT$$

$$P = NkT\frac{1}{V}$$

We expect that a graph of pressure against the inverse of volume will give us a directly proportional relationship, with a gradient of NkT. We can use the gradient to determine the error associated with the experimental results by comparing it to the theoretical values. I have selected 5 volumes to test, which are shown in the following table:

Box Length (m)	Box Volume(m^2)	Inverse Volume (m^{-2})	log($1/V$)
0.0001	10^{-8}	10^{8}	8
0.001	10^{-6}	10^{6}	6
0.01	10^{-4}	10^{4}	4
0.1	10^{-2}	10^{2}	2
1	1	1	0

Table 1: Tested box volumes for the experiment

I decided to examine volumes resulting from scaling the length of the box by a factor of 10. To make the data more presentable, I will use a log-log plot of the pressures against inverse volumes, since we are dealing with large magnitudes. I will need to adapt the linearised equation for a log-log context:

$$P = NkT\frac{1}{V}$$

$$\log P = \log\left[NkT\frac{1}{V}\right]$$

$$\log P = \log\left[NkT\right] + \log\left[\frac{1}{V}\right]$$

$$\log P = \log\left[\frac{1}{V}\right] + \log\left[NkT\right] \tag{2}$$

We see that graphing on a log-log plot, the y axis intercept will give the log of NkT, which will make it straightforward to compare the experimental results to the input parameters of the simulation. We also expect the gradient of this graph to be 1, which will provide another way to validate the experimental findings.

Methodology

I ran the simulation for a total of 50 times, to give 10 trials for each selected volume. This provides enough data to be statistically significant, so that I can calculate standard deviation values for error values. Since a simulation is much less difficult to execute than an actual experiment, it is reasonable to take a large sample size. However, I had to be mindful of computational expense. Despite optimising my code as much as I could, a simulation would still take some time. In the experiments, I kept the temperature fixed at 300 K, and set the number of particles to 10^5, which was the most my computer could feasibly handle. One factor that does not remain exactly constant throughout trials is the run time. The code is designed to run until every particle has collided with the walls at least once. On the one hand, this ensures that the data is consistent by allowing all the particles to collide. On the other, it results in the run time being inconsistent. I suspect this won't affect results, but I will address this again after seeing the results.

Raw Data

Pressure in box at different box volumes (Pa)					
Trial Number	Box Volume (m^2)				
	10^{-8}	10^{-6}	10^{-4}	10^{-2}	1
1	2.79E-09	1.09E-11	1.05E-13	2.15E-15	4.70E-18
2	8.16E-09	7.05E-12	4.86E-13	2.63E-15	5.19E-17
3	8.17E-09	1.44E-11	8.47E-14	1.69E-15	3.26E-17
4	1.51E-09	9.84E-12	4.50E-13	4.63E-16	4.48E-18
5	3.21E-11	1.69E-11	8.36E-14	8.65E-16	1.14E-17
6	5.62E-10	7.84E-12	3.79E-14	7.63E-16	1.04E-17
7	8.93E-09	1.34E-12	2.58E-14	4.25E-15	8.56E-18
8	9.94E-10	2.39E-11	8.05E-14	8.86E-16	8.79E-18
9	1.95E-09	1.55E-11	2.17E-13	9.87E-16	1.10E-17
10	9.21E-10	6.27E-11	3.84E-13	2.54E-15	1.91E-17
Average	3.40E-09	1.70E-11	1.95E-13	1.72E-15	1.63E-17
Standard Deviation	3.55E-09	1.72E-11	1.78E-13	1.18E-15	1.50E-17

Table 2: Pressure data collected for each box volume in the experiment

The above table shows the raw data collected from the experiment. As can be seen, it is difficult to interpret due to the magnitude of the values, however, we only take the average value and standard deviation from here to move on. However, we can draw some conclusions even from this raw data. It seems that the pressure decreases by the same magnitude when volume is decreased, which lends credence to a linear relationship between the two parameters. An interesting observation is the high standard deviation values, which indicates that there is a lot of variance throughout the trials. However, the values are almost exactly what we expect. Perhaps this can be explained by not having sufficient particles in the experiment to accurately represent a more realistic gas density at the given volumes.

Data processing

Next, I will process the raw data to make it more presentable. I took the log (base 10) of $1/V$ for all volumes (as shown in the Theory), also also took the log of all pressure values. We also needed to determine the error for the log pressure values. For $\log P$, the error is simply $\Delta P/|P|$. For example, for the first tested volume:

$$\begin{aligned}\Delta \log(P) &= \frac{\Delta P}{P} \\ &= \frac{3.55 \times 10^{-9}}{3.40 \times 10^{-9}} \\ &= \approx 1.04\end{aligned}$$

There is no uncertainty associated with the volumes since they are controlled within the simulation. Applying the above processing to all data, we have the following results shown in the below table:

$\log(1/V)$	$\log(P)$	$\Delta\log(P)$
8.00	-8.47	1.04
6.00	-10.77	1.01
4.00	-12.71	0.91
2.00	-14.76	0.69
0.00	-16.79	0.92

Table 3: Log-log data and error from the experiment.

I then plotted this data using the linear regression code that I wrote at the beginning of the semester. Since we do not have any x uncertainties, I am not limited by the shortcomings of the code, which is not able to weight x uncertainties.

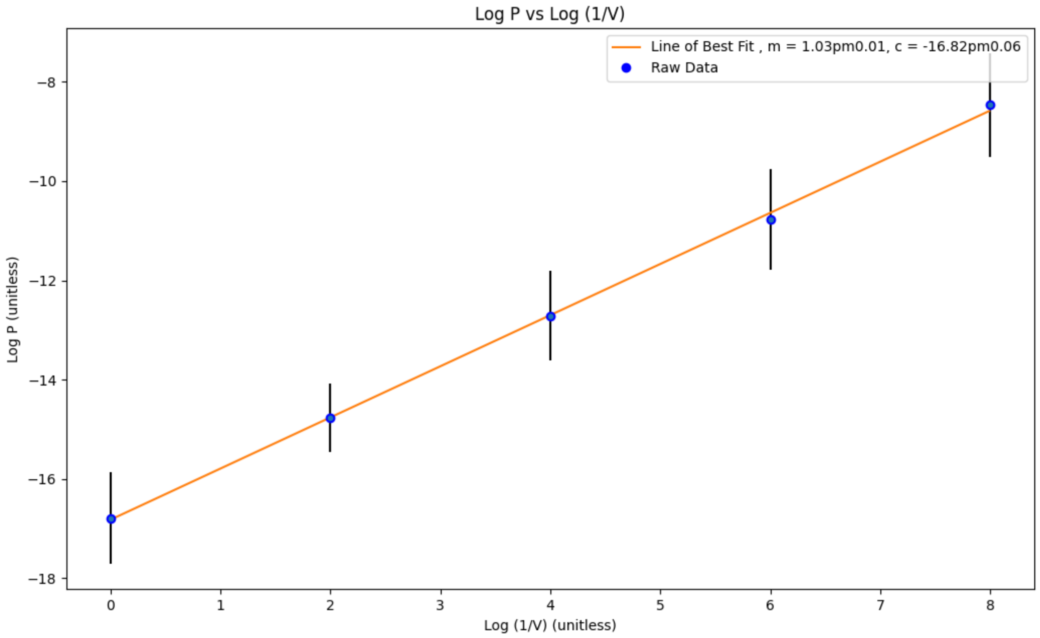

Figure 2: Linearised data. Gradient: $m = 1.03 \pm 0.01$, Intercept: $c = -16.82 \pm 0.06$

We can now calculate the experimental NkT value. For simplicity, I will refer to the experimental NkT value as C_e, and the theoretical value as C_t.

$$C_t = NkT = 10^5 \times 1.38 \times 10^{-23} \times 300 = 4.14 \times 10^{-16}$$
$$C_e = 10^m = 10^{-16.82} \approx 1.51 \times 10^{-17}$$

We also need to calculate an error for the experimental C_e value.

$$\begin{aligned} \Delta C_e &= C_e \frac{\Delta c}{|c|} \cdot \ln 10 \\ &= 1.51 \times 10^{-17} \cdot \frac{0.06}{16.82} \cdot \ln 10 \\ &= 1.2 \times 10^{-19} \end{aligned}$$

Our final value for C_e is the following:

$$C_e = (1.51 \pm 0.01) \times 10^{-17}$$

Discussion

The experimentally derived C_e value differs from the theoretical C_t by about a factor of 10, which is a fairly significant difference. There may be a few reasons for this. The lower experimental value indicates that there may be a lower effective temperature in the system, or perhaps there are particles escaping. Let us consider the first case. The temperature set in the code will change the normal distribution that gives particles their initial velocity. Initial velocities are drawn from $N(0, \sqrt{NkT})$. Due to the random nature of this sampling, it is possible that we simply had lower velocities, and thus a lower apparent temperature by chance. However, since there are 100000 particles in the system, it seems unlikely that this random probability would make any difference. A more likely possibility is some particles escaping. While the code does attempt to prevent particles from escaping, it is possible that if they acquire an extreme amount of speed, they could escape. The code will flip the sign of the particles' velocity upon hitting a wall. However, in reality there is no physical collision, since it is a simulation, and if the particle has enough speed it could escape the box and not return even after the velocities are flipped. However, I suspect that this would also be unlikely. At smaller scale testing with a smaller number of particles, it was clear that they were not easily able to escape. It is unlikely that a significant number of particles could escape even with 100000 particles present. A third possibility is the randomness associated with a low particle density. The more particles per volume, the less of a factor random variations in particle motion will have on results. Even with 100000 particles, there may be enough variation to cause some deviation from the expected result.

Part 2 - Simulating an interacting ideal gas

Introduction

The Lennard Jones Potential

In Part 1, we considered a gas where a particle is given an initial velocity and is allowed to move. However, this velocity is unchanged throughout the experiment, save for when the velocity is reversed upon collision with the walls. However, this does not change the magnitude of the velocity. A real gas however will have intereactions between the particles, whether that be an electrostatic attraction, or perhaps a nuclear force at sufficiently small distances. For this simulation, I will utilise a Lennard-Jones potential, which is shown below [2]:

$$V_{LJ}(r) = 4\epsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right] \tag{3}$$

Here, ϵ represents the depth of the potential well, and σ represents the size of the particle. The expression has two terms, one which provides an attraction and the other provides repulsion. Depending on the magnitude of r, either the attractive or repulsive component will be dominant. At large r, the attraction is stronger, and at small r, the repulsion is stronger. At very large distances, the interaction is negligible. I will use an alternate form of the expression , which is easier to implement in code [2]:

$$V(r) = \frac{A}{r^{12}} - \frac{B}{r^{6}} \tag{4}$$

where

- $A = 4\epsilon\sigma^{12}$
- $B = 4\epsilon\sigma^{6}$

For the purposes of the simulation, it would be better to work in Cartesian coordinates rather than cylindrical coordinates. This is a simple conversion to make, using $r = \sqrt{x^2+y^2}$.

$$\begin{aligned}
V(r) &= \frac{A}{r^{12}} - \frac{B}{r^{6}} \\
V(x,y) &= \frac{A}{(\sqrt{x^2+y^2})^{12}} - \frac{B}{(\sqrt{x^2+y^2})^{6}} \\
V(x,y) &= \frac{A}{(x^2+y^2)^{6}} - \frac{B}{(x^2+y^2)^{3}}
\end{aligned}$$

Here, x is the horizontal distance between particles, and y is the vertical distance. Next, we can calculate the force between particles:

$$\begin{aligned}
F = -\nabla V &= -\left(\frac{\partial V}{\partial x}, \frac{\partial V}{\partial y}\right) \\
&= -\left(\frac{\partial}{\partial x}\left[\frac{A}{(x^2+y^2)^6} - \frac{B}{(x^2+y^2)^3}\right], \frac{\partial}{\partial y}\left[\frac{A}{(x^2+y^2)^6} - \frac{B}{(x^2+y^2)^3}\right]\right) \\
&= \left(\frac{12Ax}{(x^2+y^2)^7} - \frac{6Bx}{(x^2+y^2)^4}, \frac{12Ay}{(x^2+y^2)^7} - \frac{6By}{(x^2+y^2)^4}\right)
\end{aligned}$$

We now want to determine the radius at which the potential is minimised. This is where the force between particles is zero. We can set the partial derivative equal to 0, and solve for $r_{min} = \sqrt{x^2 + y^2}$:

$$\begin{aligned}
\frac{12Ax}{(x^2+y^2)^7} - \frac{6Bx}{(x^2+y^2)^4} &= 0 \\
\frac{12Ax}{(x^2+y^2)^7} &= \frac{6Bx}{(x^2+y^2)^4} \\
\frac{2A}{(x^2+y^2)^3} &= B \\
\frac{2A}{r_{min}^6} &= B \\
r_{min}^6 &= \frac{2A}{B} \\
r_{min}^6 &= \frac{2 \cdot 4 \cdot \varepsilon \cdot \sigma^{12}}{4 \cdot \varepsilon \cdot \sigma^6} \\
r_{min} &= 2^{1/6}\sigma
\end{aligned}$$

I implemented the Lennard Jones potential into the simulation. This required some consideration of ideal parameters to use, since the situation is much more complicated than Part 1. Firstly, I decreased the box size to 50σ. We need the box to be in the same or similar order of magnitude to the particles to observe any interactions. At larger box sizes, this would not be easy to do. I also needed to select an ϵ and σ. For these I used $\epsilon = 1.95 \times 10^{-21}$ J and $\sigma = 0.34 \times 10^{-9}$ m. I also switched to utilising the velocity Verlet method instead of the Euler method to numerically integrate positions of the particles over time, which increased the accuracy and stability of the system, as shown in the code below.

```

        pos += vel * dt + 0.5 * acc * dt**2 #updating position of particle for half the timestep
        vel += 0.5 * acc * dt # updating velocity of particles for half the timestep

        acc.fill(0) # resetting all accelerations to zero

        for i in np.arange(N):

            for j in np.arange(i-1): #looping over upper triangular matrix, to calculate unique particle pairs

                x = pos[i,0] - pos[j,0] # x distance between particle i and j
                y = pos[i,1] - pos[j,1] # y distance between particle i and j

                acc[i,0] += mi * -Fx(x,y)   #calculating acceleration of particles
                acc[i,1] += mi * -Fy(x,y)
                acc[j,0] += mi * Fx(x,y)
                acc[j,1] += mi * Fy(x,y)

        vel += 0.5 * acc * dt # updating the velocities based on calculated accelerations
```

Calculating the temperature of the system

The temperature of the system is of great interest to us, and so we need a way to calculate it. We make use of the equipartition theorem, which is that at a temperature T, the average energy of any degree of freedom is $\frac{1}{2}kT$ [1]. Since we have a 2 dimensional simulation, we have two degrees of freedom instead of the typical 3. Using the equipartition theorem, we can calculate the temperature of the gas by taking the average kinetic energy of all the particles:

$$\begin{aligned}
\bar{E_k} &= \frac{1}{2} f k T \\
\frac{1}{N}\sum_{i=1}^{N}\left[\frac{1}{2}mv_{xi}^2 + \frac{1}{2}mv_{yi}^2\right] &= \frac{1}{2}2kT \\
\frac{m}{2N}\sum_{i=1}^{N}\left[v_{xi}^2 + v_{yi}^2\right] &= kT \\
T &= \frac{m}{2kN}\sum_{i=1}^{N}\left[v_{xi}^2 + v_{yi}^2\right]
\end{aligned}$$

This was implemented into the code. I also implemented a system to perform 5 trials of the experiment, which allows us to examine errors in the temperature over time.

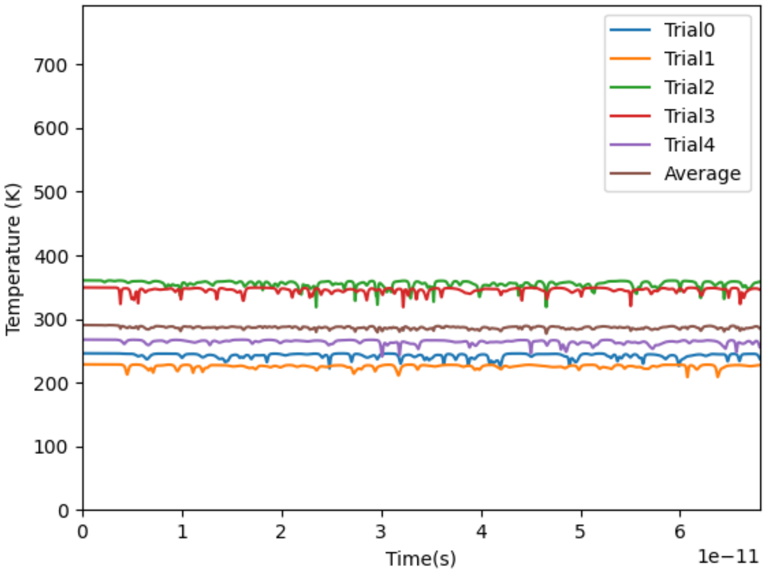

Figure 3: Temperature of the system for each trial and the average over all trials

Behaviour of the interacting system

Due to the increased computational overhead of the interactions, we cannot run the sheer volume of particles as from Part 1. However, even with a more reasonable 49 particles, we can make some observations. Firstly, we can make some qualitative visual observations. Compared to Part 1, where the particles aimlessly bounced around, the particles will draw each other in, due to the attractive component of the potential. Upon reaching a certain distance however, the particles begin to repel, causing them to separate. We can now examine the temperature of the particle over time. I expect that temperature of the particles will be roughly the input temperature of 300 K, but possibly higher or lower due to the fact that velocities are assigned from a normal distribution. With a relatively small number of particles compared to a real world system, this randomness will have more of an effect. I also expect the system to conserve energy, and remain constant in temperature over time, with perhaps some minor variations. There is no energy escaping or entering the system in any way, so there is nothing to reduce the temperature of the particles.

We see that the temperature for each trial remains roughly constant over time, with some minor variations. We can also see that there is some significant variation of the temperature between trials. However, this can be explained by the reasons I mentioned earlier. A notable fact is that the average temperature between trials is 300 K, which is what we expect. I suspect that running more trials would result in the average converging towards 300. Currently though, with 49 particles, we have an error in the temperature of roughly $\pm 80K$.

Modelling the cooling of a gas

When observing real world gasses, we can often observe a phase change when a gas reaches a certain temperature. We want to see if we can make a similar observation with the simulated particles. Previously, we noted that the temperature of the system remains constant over time. I now introduce a cooling factor $\lambda = 0.9995$. This factor will dissipate energy over time from the system, one per time step. This reduces the average velocity, and hence temperature of the system over time. As we reduce the temperature, we expect that the gas will begin to condense, possibly resulting in a liquid then solid phase. To study this phenomenon, we will observe temperature and pressure graphs over the lifetime of the experiment, as well as examine the physical distribution of the particles in the system. For this experiment, I reduced the size of the box further, to 15σ. This encourages the particles to interact more, which allows for phase transitions to occur more easily. Other notable parameters are an initial temperature of 500 K. I increased the initial temperature to allow more initial particle interactions, and also to provide a wider range of temperatures to observe particle behaviour. Parameters such as ϵ and σ are unchanged for this section.

Pressure and temperature

The simulation calculates the temperature and pressure in the system over time, which we can observe. The temperature calculation is unchanged from before, however the pressure calculation was redeveloped. In Part 1, we calculated the average pressure across the whole experiment. However, in this case, we want the pressure to update every frame with the simulation. To calculate pressure, I took the cumulative sum of impulses from collisions at each time step, then divided this by the length of the wall and the time that had passed. This code is shown below.

```
for t in np.arange(N_time):
            Pres[tn,t] = Jtc[t] / (2* t * dt * L) #cumulative sum of pressure divided by the
     time since beginning of experiment and length of wall
```

The simulation runs experiments 5 times, which allows us to plot multiple pressure and temperatures curves, providing an estimate of uncertainty for the experiment.

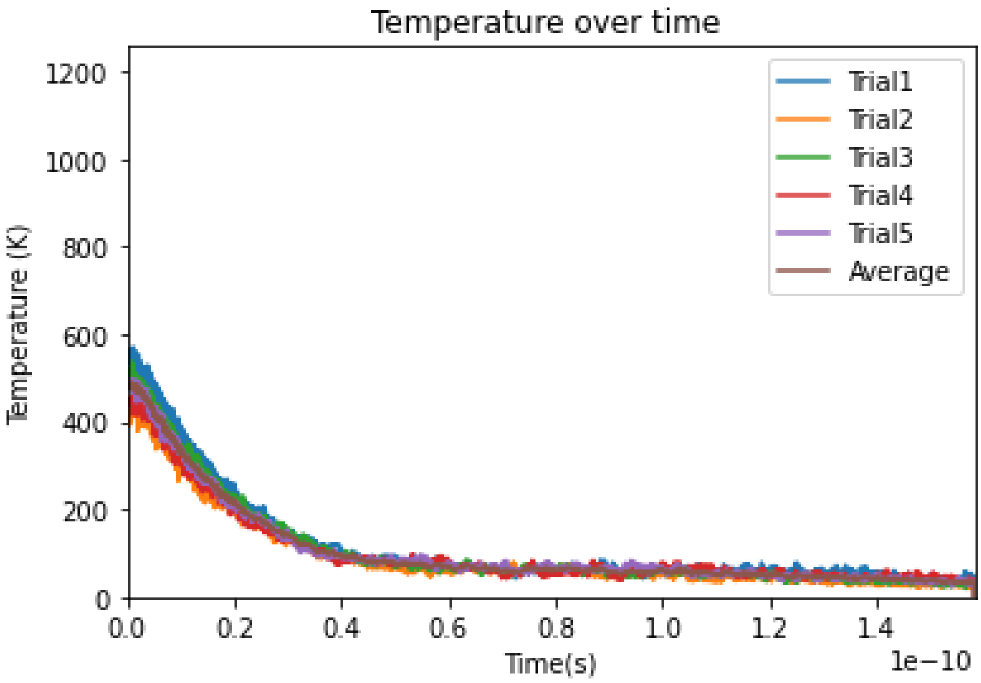

Figure 4: Temperature of the system over time

Figure 4 shows the temperature over time for the system for each trials, along with an average across trials. Since the dissipation factor λ removes a proportion of particle velocity each timestep, we expect an exponential decreasing graph for temperature, which is what we can observe in the figure. As with the non cooling case earlier, there is some some variation at the beginning for intial temperature, however for all trials the temperature converged after reaching about 180 K. There are still some variations and perturbations, but these are to be expected. Notably, there are some flat sections in the temperature curve, especially noticeable through 0.6×10^{-10} and 1×10^{-10}, which indicates a phase transition may have been occurring through this region.

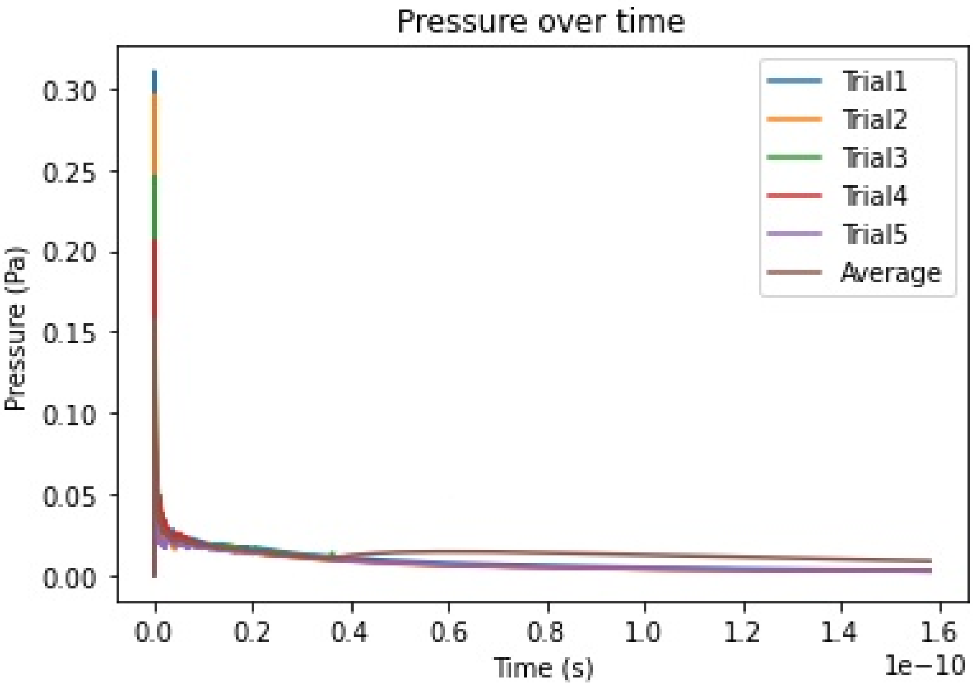

Figure 5: Pressure over time in the system

The pressure time graph shows a similar trend to the temperature graph, which is expected. At the beginning of the experiment, we have high pressures due to the regular collisions of the particles with the walls. However, as the experiment progresses, two major factors contribute to the drop in pressure. The first is that the particle speed begins to decrease due to the dissipation factor. A lower speed means less impulse acting on the walls, decreasing the pressure. The second factor is that due to the attractive forces between particles, as the speed decreases, the number of collisions with the walls also begins to decrease. This is very evident when the system in in a solid state, where the pressure drops to 0 since there are no collisions with the walls.

Phase transitions

When cooling a real world gas, we expect to see a gaseous, liquid and solid phase. These phases are characterised by the distance between particles, the speed of the particles, and perhaps some special behaviour as various structures form. The particles begin in gaseous phase, as they have a lot of kinetic energy and the attractive forces are not sufficient to keep them together. Below are some sample distributions of the particles in their gaseous phase:

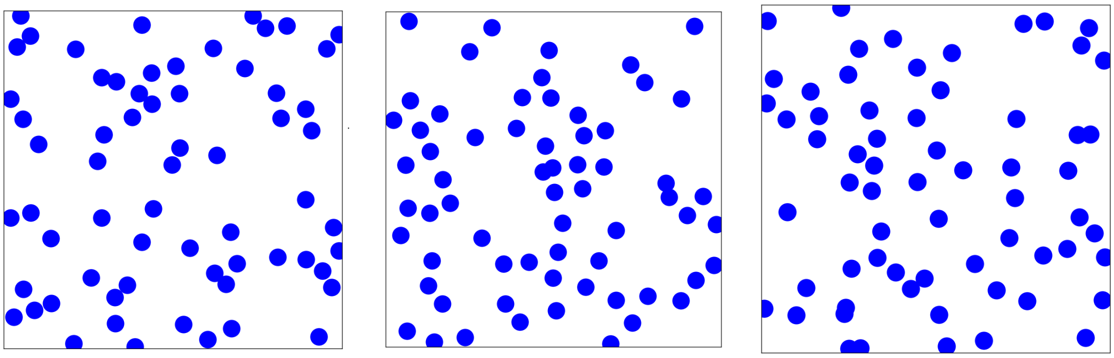

Figure 6: Gaseous distribution of particles

As we can see, the distributions are quite random. This is behaviour we expect from a gaseous substance. As the temperature cools further, we can begin to observe a liquid phase. In this phase, the particles are closer together, but still posses more velocity than in solid form:

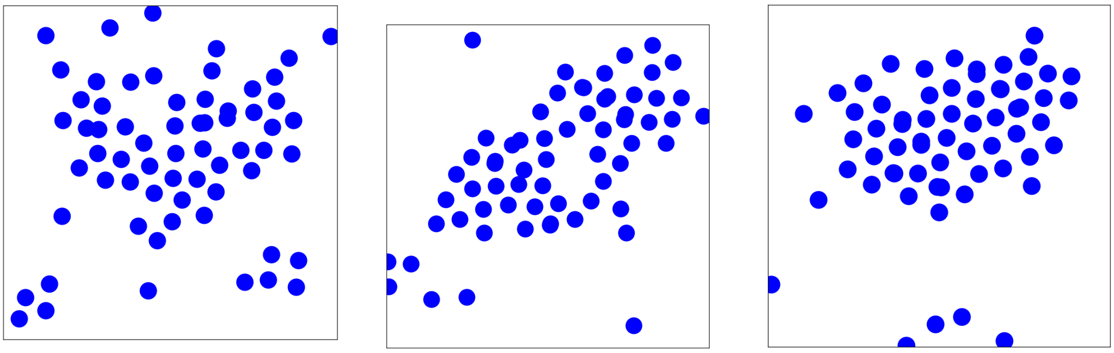

Figure 7: Liquid distributions of particles

We see that in liquid form, most particles are together, however they possess an irregular macroscopic shape, and are not very rigid. I have included a few successive liquid distributions in Figure 7. As the temperature decreases, the particles do begin to approach a solid state, which is shown below.

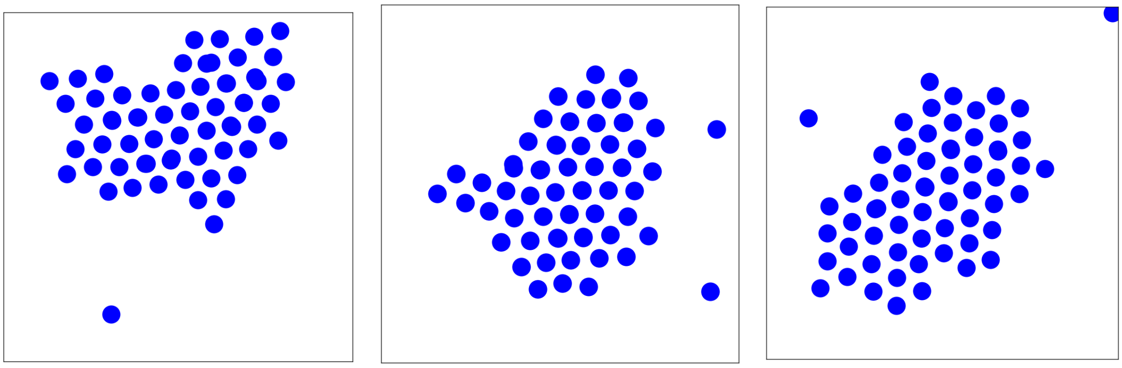

Figure 8: Solid distributions of particles

As expected, in solid form the particles are the close together. Notably, we can observe these regular crystalline structures forming, as each particle settles into the close approach distance with other particles. There are a few particles that have become separated, which will not have the velocity to rejoin the others. This is a result of the small particle number. A larger particle number would ensure consistent interaction between particles. However, we are limited computationally here, so a few lonely particles is unfortunate but not devastating.

References

[1] Schroeder, D.V. *Thermal Physics* San Francisco: Addison Wesley Longman (1999)

[2] Buschow, K. H. J *Encyclopedia of Materials: Science and Technology* Pergamon Imprint (2001)

[3] Argun A. , Callegari A. Volpe G. *Simulation of complex systems* Bristol England Temple Circus, Temple Way, Bristol BS1 6HG, UK : IOP Publishing (2021)

[4] Das M. , Schmidt C. F. Murrell M. Introduction to Active Matter Soft Matter (2020)

[5] Keith Morris (2016) Aberystwyth, Ceredigion, Wales, UK, GB, Great Britain, Europe