Thermodynamics Snookered – Summary Report

Elizabeth S.Z. Tan

# Developing the simulation – Algorithms devised beyond the script

## Collision

One of the main attributes of this class was the *collide* attribute. When the balls collide, their velocities change. The new velocities are given by the equation

(1)

where **v**1’ is the velocity of the first ball after collision, **v**1 is the velocity of the first ball before collision, m1 and m2 are the masses of ball 1 and ball 2 respectively, and **r**1 and **r**2 are the position vectors of ball 1 and ball 2 respectively [1]. The velocity after collision of ball 2 can be found with the same equation but interchanging the value of the velocity vector, mass, and position vector of ball 1 with ball 2.

## Instantiating the balls in the simulation

The *Simulation* class allows you to choose how many balls are in the container. The balls are instantiated in the container with random velocities and positions (all positions are inside the container) – but this method could result in the overlap of balls. However, if the balls overlap, it is required for them to “phase” out of each other (otherwise this would result in overlapping balls being stuck, effectively, and oscillating about each other). The *time\_to\_collision* attribute of the *Ball*class took this issue into account. When solving the quadratic, (Task 1) if one of the two solutions for the time to collision was positive and the other negative, it means that they are overlapping and the function returns NaN.

## Calculating the next collision in the simulation

In the *Simulation* class, the attribute *next\_collision* produces a square matrix of size equal to the number of balls to store the *time\_to\_collision* information of each ball with every other ball. Each element ij of the matrix (tij) is the time to collision of ball i with ball j. The upper half, including the diagonal, is filled with NaNs. This is because the matrix is symmetric (a collision of ball i with ball j is the same as a collision of ball j with ball i), and elements ii or jj of the matrix (the diagonals) signifies a ball colliding with itself, which practically has no meaning. The algorithm finds the lowest possible time to collision, dt, then all balls move for time dt, until the soonest collision takes place. This time matrix is recalculated after every collision. An example matrix for two balls and the container is shown below (ball 0 is the container).

## Calculating pressure

This simulation is not one of an ideal gas. Molecules in an ideal gas have negligible size compared to the container, which is not true in the simulation. We therefore cannot use the ideal gas laws to calculate pressure – this must be done from first principles. The equation for pressure is

As impulse is given by

We can calculate the impulse by calculating the change in momentum of the balls colliding with the container then divide by the duration to find the force exerted. As the simulation is in two dimensions, the ‘area’ in the pressure equation is the circumference of the container. This is how pressure was calculated.

# Results obtained

## Figure 1 – Histogram of ball distance from container centre

In *Figure 1* there is a linear line which is plotted using the centres of the tops of the histogram. We would expect a linear plot for this data. One can imagine the container to be made up of lots of (an infinite number, in fact) of concentric circles. The number of points able to fit on each concentric circle is proportional to the circumference, 2πr. So, when the radius of the concentric circles increases, the number of points on each circle increases linearly. Since we expect the balls in our simulation to be distributed uniformly in the container, we can use this model to describe the relationship between the average ball distance from the centre, by imagining each ball being placed on a concentric circle. *Figure 1* doesn’t show a very sharp linear plot, but this is due to the randomisation of the ball’s velocities and positions. Should there be more balls and more frames considered, the plot will look more linear.

## Figure 2 – Histogram of inter-ball separation

The inter-ball separation is given by

where r is the separation, R is the radius of the container and A is the scaling factor (the original function didn’t include A as it was a probability density function) [2]. The *optimize* function plotted this function onto the histogram as shown in *Figure 2*. The outputted values of A and R are 280 ± 40 and 15.1 ± 3.6 units2 respectively. A is arbitrary (depends on number of balls plotted). The radius of the container, R, outputted by the *optimize* function is almost half the actual value. The reason for this is that *Equation 4* assumes the balls to be point-particles with no size.

## Figures 3 and 4 – System kinetic energy and momentum

## Unsurprisingly, the total kinetic energy and momentum of the system remains constant. The collisions between the balls are perfectly elastic, so no kinetic energy is lost. Although the kinetic energy and momentum of the container are not taken into account in these calculations – the system is comprised of only the balls inside the container – the container was programmed to have a very high mass (~1020) so the kinetic energy and momentum transferred to the container is negligible.

## Figure 5 and 6 – Pressure versus temperature and how changing the radius affects the equation of state

## The ideal gas law is as follows

## where P is the pressure of the gas in Pascals, V is the volume of the gas in m3, N is the number of molecules of the gas, and k is Boltzmann’s constant (1.38 × 10-23 J/K) [3]. Rearranging this, the gradient is equal to

## 

The value of the gradient produced when fitting a linear plot to *Figure 5*, however, is more than that, with gradient equal to 2.95 ± 0.11 × 10-24 . The reason for this discrepancy lies in the ideal gas assumptions. Particles of an ideal gas have negligible volume which is not true for this case because the balls take up significant volume of the container.

*Figure 6* explores this discrepancy further. It plots the gradient of the P-T plots for balls of six different radii. One can see that the gradients are getting smaller and converging upon the theoretical ideal gas value of the gradient with the given parameters.

## Figure 7 – Histogram of ball speeds

*Figure 7* shows a histogram following a Maxwell-Boltzmann distribution. The Maxwell-Boltzmann function plotted onto the curve is given by

where m is the mass of the particles in kg, v is the velocity of the particles in m/s, T is the temperature in K, and A is a scaling constant [4]. The parameters outputted by the *optimize* function are m = 0.46 ± 0.16 kg, T = 1.6 ± 0.6 × 1022 K and A = 1590 ± 210. Although the shape of the histogram fits this curve, the mass parameter given by the *optimize* function is less than half the actual value (actual value being m = 1). Again, this is likely due to the simulation not modelling an ideal gas.

## Figure 8 – Plot of data used to fit van der Waal’s law

The van der Waal’s equation is given by

where V is the volume of the container (in this case V would be the area, since the simulation is in 2D), a is the van der Waal’s constant to measure the attractive force of the molecules in the gas and b is a volume correction factor [4]. The parameters found from the *optimize* function are N = 130 ± 4, V = 2280 ± 20, a = 25 ± 2 and b = 13 ± 1. The number of balls and the area of the container is similar to the actual number – 100 balls and an area of 2830 units2. The values of a should be small, as no attraction was coded into the simulation.

# Conclusion and Evaluation

This 2-D simulation plots balls with random positions and velocities into a container to model the behaviour of gas particles. It explores the distance between the gas particles, how well the model fits an ideal gas. It also explores how the speeds of the gases fit a Maxwell-Boltzmann distribution. Finally, van der Waal’s equation was used to calculate the estimates of the simulation parameters.

There were several things that could have been improved about this simulation.

## Improving the code

A minor bug which has not been fixed is that the code does not run unless we animate the balls. Given more time, this could be fixed.

When the balls are instantiated, they have the capability to overlap which is suboptimal as this cannot happen in real life and takes the simulation a few frames to run without any overlap. A better method would be to instantiate the positions of the balls systematically, so there is no chance of overlap. However, given the time restraints, it was difficult to maintain the ability to easily change the number of balls, radius of balls and radius of the container.

Also, when instantiating the balls, the randomness of the positions and velocities meant that the plots produced could be different for each individual run. It would be more consistent to *fix* the random seed so that the random values for position and velocity were constant for each run of the simulation, or systematically placing the balls would also be a solution to this issue (by fixing the initial positions).

A method can be implemented to make the simulation run faster. When calculating the time matrix in the *next\_collision* attribute of the *Simulation* class, instead of recalculating the whole time-matrix, it would be more efficient to only recalculate the time to collision of the rows and columns corresponding to the balls i and j which have just collided.

To create *Figure 5*, the pressures and temperatures were recorded manually, as to change the temperature, the velocity generating function needed to be changed, and this was in a different file. To produce more data points, it would be better if this process was automated.

## Extension to the investigations

The histograms shown in *Figure 1* and *Figure 7* only had ten bins. When more bins were plotted, the plot ended up looking very jagged. An improvement would be to use more frames and thus collect more data to produce a smoother histogram with more bins. This would also decrease the uncertainty inthe plots on top of the histogram and make the plots smoother.

Because of the randomised method of instantiating the balls, the shape of the plots could differ a lot for each run of the simulation. Given more time, it would be better to run the simulation for longer (the longer the simulation runs, the more time it has to reach an equilibrium state) before plotting the data. It would be interesting to see and compare the outputted plots from running many different simulations with the same initial parameters.

In *Figure 5*, an extension would be to plot more data points for the decreasing ball radius to better visualise the gradients tending to the value of the ideal gas law gradient.

# References

1. J. Raymond “10.4.1 Elastic collisions”. *A radically modern approach to introductory physics: Volume 1: Fundamental principles*. Socorro, NM: New Mexico Tech Press.
2. J. Cohen, D. Courgeau. “Modeling distances between humans using Taylor’s law and geometric probability”. Available online < <https://www.researchgate.net/publication/312294065_Modeling_distances_between_humans_using_Taylor%27s_law_and_geometric_probability>> Accessed on 8 December 2021
3. WJEC Eduqas “Physics Data Booklet”. Available online < <https://www.physicstutoronline.co.uk/wp-content/uploads/2019/05/WJEC-A-LEVEL-PHYSICS-DATA-SHEET-S17-2420U10-1A.pdf>> Accessed on 8 December 2021
4. A. MacKinnon (2016). Revised by R. Kingham (2020), “Project B: Thermodynamics Snookered”. *Imperial College Physics Year 2 Computing*

# Chart Description automatically generatedPlotsChart Description automatically generated

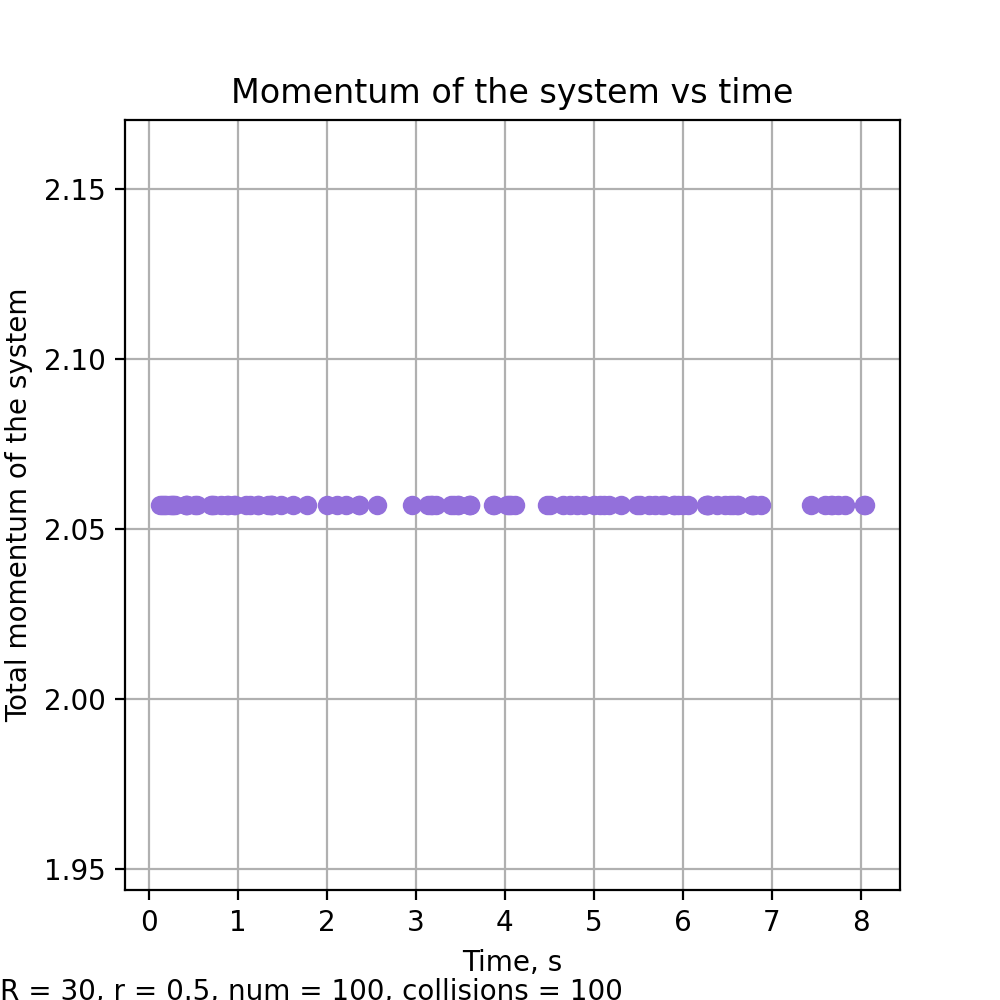
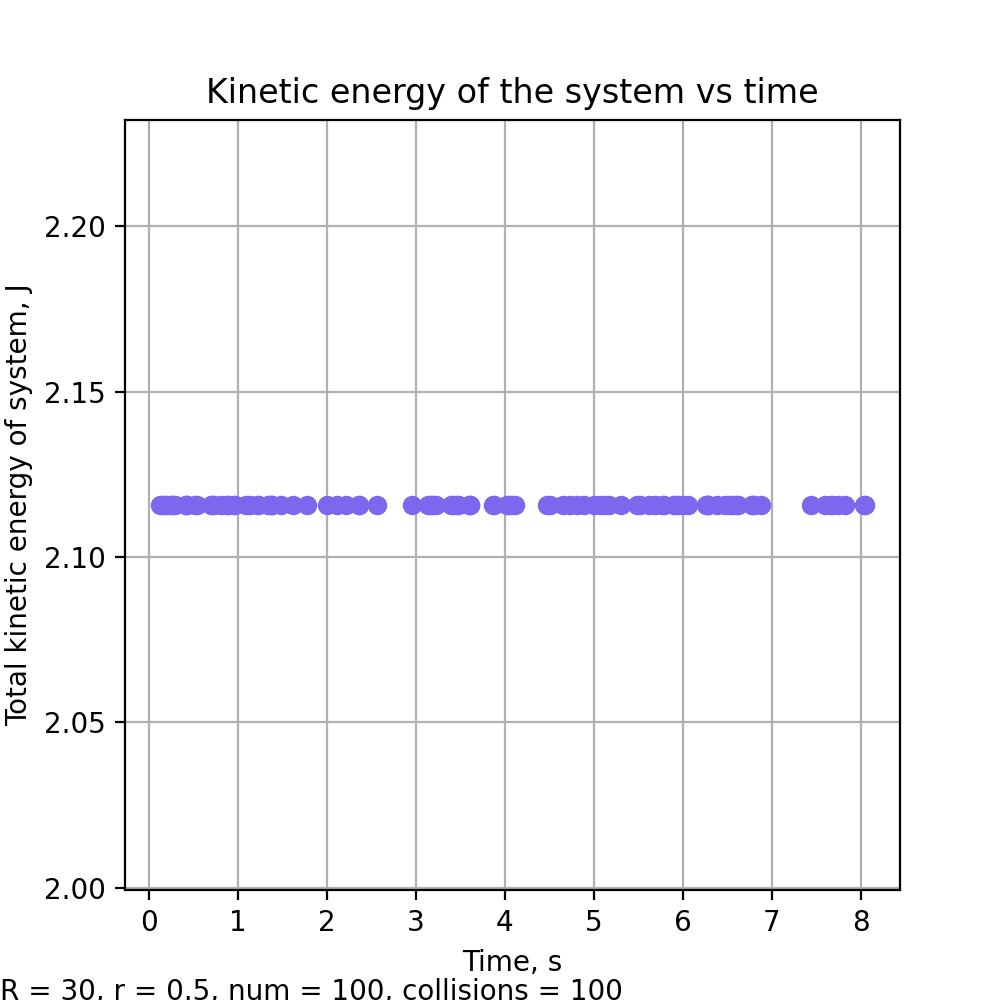


Figure 1 – Histogram of ball distance from container centre (bins = 10). Black line shows a linear plot onto the points of the bins centres.

Figure 2 – Histogram of inter-ball separation (bins = 50). Black line shows Equation 4 fitted onto the plot using the points at the centre of each bin. Parameters given are A = ﻿2800 and R = 15.1.

Figure 3 – System kinetic energy versus time

Figure 4 – Momentum versus time

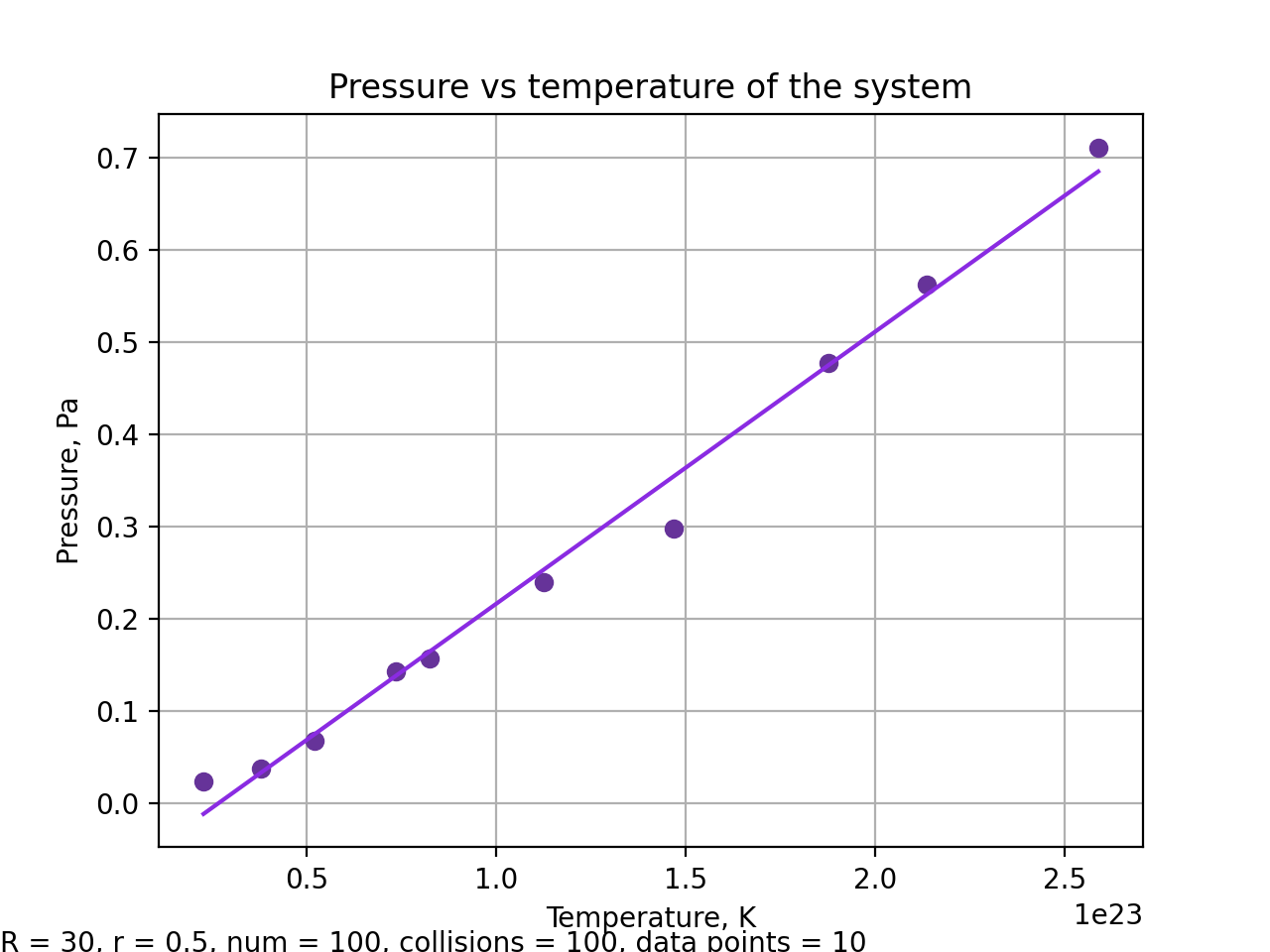
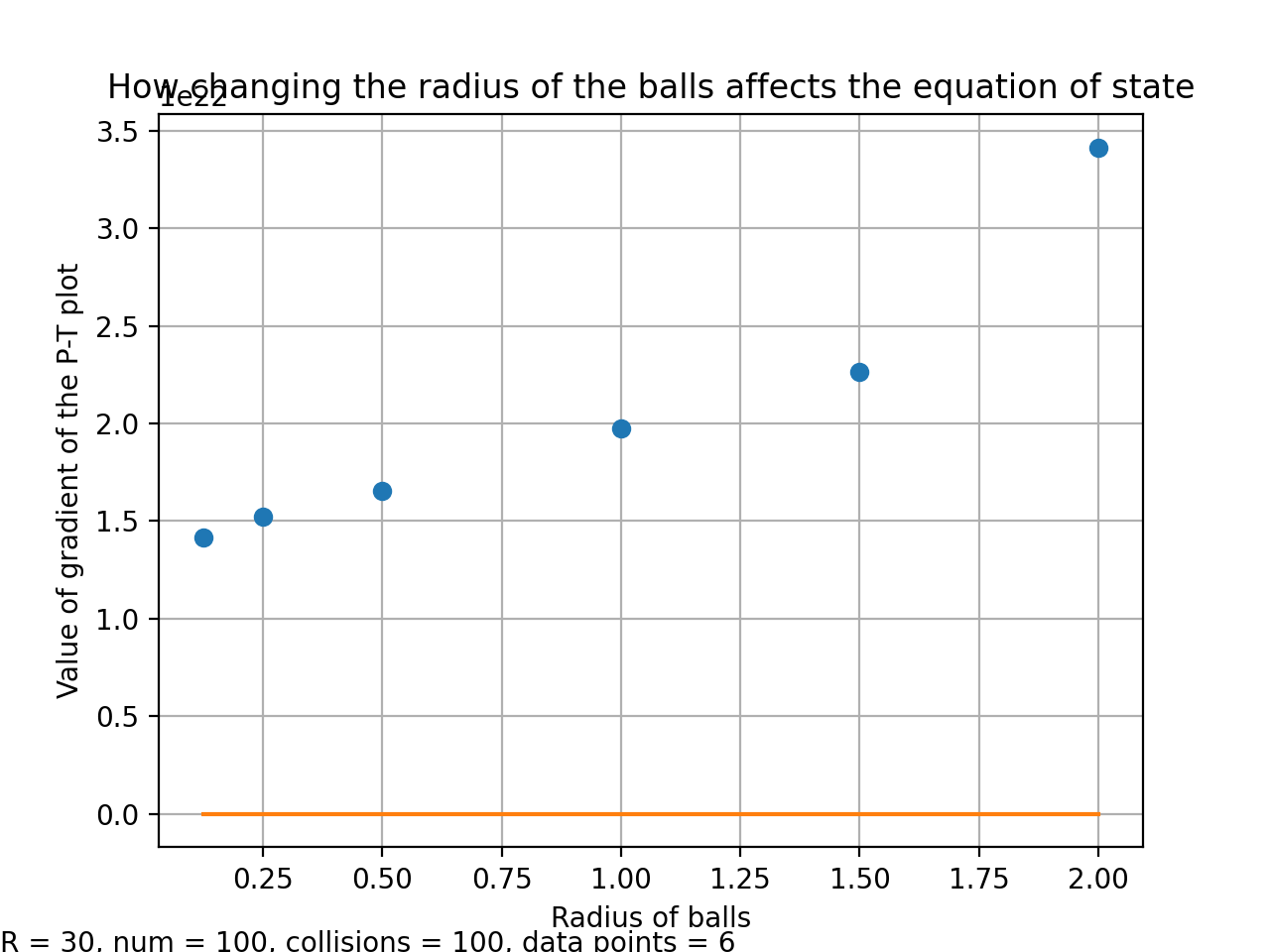


Figure 6 – A plot to illustrate how changing the radius affects the equation of state. The orange line is the value of the gradient should the simulation be considering an ideal gas.



1e22

# Chart, line chart Description automatically generatedReferences

Figure 7 – Histogram of ball speeds. Black line shows a Maxwell-Boltzmann function (Equation 5) plotted onto the bin centres. Parameters are m = 0.461, T = 1.63 × 1022 and A = 1590.

Figure 5 – Pressure versus temperature. Purple line shows a linear fit onto the data points.

Figure 8 – Plot of data used to fit van der Waal’s law. Pink line shows Equation 6 plotted onto the points. Parameters are N = 129, V = 2280, a = 24.7 and b = 13.0.

1. <https://en.wikipedia.org/wiki/Elastic_collision>
2. Madhuwrit’s link
3. A level formula book