Thermodynamics Snookered

Ali Al-Ali
Blackett Laboratory, Imperial College London

***Abstract*—The limits of the ideal gas model were tested by varying ball size and number. For suitable parameters, the Maxwell-Boltzmann distribution was found to approximate the system well. However, for larger ball radii, Van Der Waals' parameter, b, becomes significant as the equation of state changes.**

A. Initialising the balls

The balls are initialised in an ordered, yet pseudo-random, manner to ensure that they aren't initialised overlapping which would create errors. The container is automatically initialised with the negative of `container_radius` to avoid user error, with a mass of $1 \times 10^{100}\, kg$ to keep it fixed and with a ball id of -1 to differentiate it from the balls when debugging.

The `get_available_positions` method creates a list of possible positions for the balls to fill in such a way that they don't overlap, chooses which positions to fill with balls pseudo-randomly and returns this as a list which is then used to provide the ball positions when initialising the balls. It works by creating a number of concentric circles inside the container, on which the potential positions are evenly spaced. The method calculates the number of concentric circles to use by dividing the container radius by the ball diameter, giving the number of circles that would initialise the balls all touching each other if all positions are filled. This number is then scaled down by a factor of 0.75 to ensure that it's impossible for balls on adjacent circles to be initialised touching. This number of concentric circles is then inscribed into the container and the method loops through each circle and calculates the theoretical maximum number of balls allowed on that circle by dividing that circle's circumference by the ball diameter. Again, this number is scaled down by a factor of 0.75 to make it impossible for balls on the same circle to be initialised touching. The centre of the container is always reserved as a potential ball position (as this is the only position allowed for the "0th" circle) and the rest of the potential positions are placed evenly around each circle using polar coordinates. If the number of balls the user tries to initialise the simulation with is greater than the number of potential positions, an `Exception` is raised that informs the user of the maximum number of balls of that radius that they can have in their simulation.

Once the potential positions are determined, the method then pseudo-randomly selects positions for the balls to initialise in from the list of all possible positions.

I. Running the simulation

`next_collision` uses `itertools.combinations` [1] to create a list of tuples of every possible combination of two `Ball` objects to loop through and find the time to collision for every pair, which it appends to a list. The minimum time in this list is then taken to be to the time to the next collision and the method moves all the balls to this time and collides the two balls that are due to collide next. `run` calls this for every frame in the simulation, resulting in order $\mathcal{O}(n^2)$ complexity. The algorithm can be reduced to order $\mathcal{O}(n)$ complexity by checking if the collision times for each pair have already been recalculated, and only recalculating the collision times for pairs involving the balls that collided last if they have. These methods also keep track of physical observables.

II. Debugging

The `run` method provides an optional animation as a debugging tool for qualitative analysis of simulation behaviour, in addition to this, pseudo-random noise was used in the ball patches to give each ball a unique colour and each frame was saved as a `.png` file to facilitate frame-by-frame monitoring of a specific ball's behaviour while debugging. To ensure reproducible ball initialisation, which is crucial in monitoring how the simulation responds to changes in code, `np.random.seed` was used in `Simulation`.

For quantitative testing, a print statement was inserted in `run` that displays which balls are colliding for every frame. This allowed for writing commands that use `ball_id`'s in `run` to check that the balls are colliding with each other and not just with the container.

III. Physics investigations

A. Simulation checks

To verify that the balls are colliding correctly, the distance that the balls extend from the centre and the inter-ball separation were plotted in Fig. 1 and Fig. 2 respectively. As expected, Fig. 1 shows that the distance between centre of the balls and the centre of the container can range up to the container radius subtract the ball radius, confirming that the balls do not escape the container. It can also be seen that the distribution is skewed towards longer distances, as most balls would be expected to be around the circumference. Meanwhile, Fig. 2 gives a symmetrical distribution and shows that the particles can be separated at most by the container diameter subtract the diameter of one ball, as expected. These plots confirmed the balls were colliding as intended, providing confidence in the subsequent physics investigations.

B. Conservation laws

After confirming the balls were colliding as intended, the next test of the simulation was to ensure that it obeys the conservation of energy and of momentum. As we are modelling the gas as hard balls that collide elastically, with no

potentials between them, the mean kinetic energy remains constant with time as expected, with a negligible standard deviation of $2.46 \times 10^{-13}\,J$. Likewise, the magnitude of the system's momentum also remains constant with time, with a negligible standard deviation of $1.14 \times 10^{-14}\,kgms^{-1}$, as shown in Fig. 3.

Both standard deviations can be attributed to the floating point calculations used, so we can be confident that the simulation obeys the conservation laws stated, given the statistical significance of these results.

C. The ideal gas law

For an ideal gas, the equation of state is given by

$$PV = Nk_{\mathrm{B}}T, \tag{1}$$

where P, V, N, k_{B} and T denote pressure on the container, container volume, number of particles, the Maxwell-Boltzmann constant and temperature respectively. This means that in 2D, pressure should vary linearly with temperature for an ideal gas in a container of fixed area. To test this, a plot of pressure against temperature was made for ball radii 10 times, 100 times and 1000 times smaller than the container radius and a line was fitted to each, as shown in Fig. 4. The temperature of the system is related to the average kinetic energy of the system, $\langle E_k \rangle = \frac{E_k}{N}$, by $T = \frac{\langle E_k \rangle}{k_{\mathrm{B}}}$ in 2D [2]. Therefore, given E_k is constant, as in Fig. 3, temperature is also constant for a given simulation, so many simulations were iterated through with different `vfactors` to vary temperature, taking the last temperature in each iteration for the data points. Whereas, pressure increases with time initially, before eventually settling at a steady value as it is calculated as a moving average which becomes more precise and statistically significant the longer the simulation has run for, as shown in Fig. 7. Therefore, the mean of the last 10 pressure measurements was taken for the data points.

Fig. 4 shows that balls of radius $0.1\,m$ and $0.01\,m$ produce very similar gradients, while balls of radius $1\,m$ give a gradient approximately twice as steep. This is to be expected as these smaller ball radii satisfy the ideal gas assumption that the particles have negligible volume compared to the container's volume [2], while the larger balls violate this. However, comparing the gradient produced by the $0.01\,m$ radius balls, $(2.7 \pm 0.3) \times 10^{-25}\,JK^{-1}m^{-2}$, to $10.1 \times 10^{-25}\,JK^{-1}m^{-2}$– the gradient expected by the ideal gas equation– we conclude that it is out of the limits of experimental uncertainty allowed for agreement with theory by $7 \times 10^{-25}\,JK^{-1}m^{-2}$. This can be attributed to only using 23 balls in the simulations for the plot (as this is the maximum number of $1\,m$ radii balls allowed in the container– an area for `get_available_positions` to improve on in future versions), which violates the ideal gas assumption of the gas containing a very large number of particles [2]. However, even 1000 balls (which is nearing the limit of the computational capability of the simulation) would be insufficient for this assumption.

D. The Maxwell-Boltzmann Distribution

The Maxwell-Boltzmann distribution for an ideal gas in 2D is given by

$$f(v) = \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^3 4\pi v \exp\left(-\frac{v^2}{2\sigma^2}\right),\ \sigma = \sqrt{\frac{k_{\mathrm{B}}T}{m}}, \tag{2}$$

where v is the particle speed, σ denotes the standard deviation of the distribution and m is the mass of one particle. For small enough ball radii, this should approximate the simulation velocity distribution well after it is run for many frames because the balls would have been given the chance to reach thermodynamic equilibrium, they would have negligible area compared to the container so they would approximate to an ideal gas, and the ball velocities are random and uncorrelated.

To investigate this, a plot of the velocity distribution of balls in the simulation after many frames was created with a Maxwell-Boltzmann distribution fit overlaid using Scipy's `curve_fit` method to find σ from our data, as shown in Fig. 5. The theoretical variance was calculated to be $1.00\,m^2s^{-2}$, while the empirical variance was found to be $0.92 \pm 0.06\,m^2s^{-2}$. While these values are similar, they do not agree within the limits of experimental accuracy which could be due to the gas not approximating well enough to an ideal gas as the number of balls is much less than Avogadro's number.

E. Van Der Waals' law

For a real gas, the equation of state is given by

$$\left(P + a\left(\frac{N^2}{V^2}\right)\right)(V - Nb) = Nk_{\mathrm{B}}T \tag{3}$$

where a and b constitute corrections for the ideal gas law for intermolecular forces and non-negligible particle volume respectively. Given that our simulation treats particles as hard balls, with no attractive potentials between them, $a = 0$ in our case. Future extensions of the simulation should incorporate Van Der Waals forces using the Lennard-Jones Potential. Whereas, we would expect b to be proportional to particle area in 2D as Nb constitutes the volume that must be subtracted from the container volume to account for the particles' size. Fig. 6 allows us to determine b from the y-intercept of the plot as $b = 0.04 \pm 0.01m^2$– approximately a factor of 1.3 higher than the ball area. This overestimation is due to the area between circles when packed together, which accounts for the extra area that the algorithm subtracts from the container. For larger balls in larger containers, this factor increases linearly as there is more area between larger packed balls. This can be seen by running the plot with the same parameters for a container radii of $10 - 10.1\,m$ and ball radius of $1\,m$ (this plot was left out in this report for brevity).

REFERENCES

[1] *itertools — Functions creating iterators for efficient looping, release 2.3*. Python Software Foundation, 2020. [Online]. Available: https://docs.python.org/2/library/itertools.html

[2] V. Tymms, *Second Year Thermodynamics and Structure of Matter Notes*. Imperial College London, 2020.

IV. FIGURES

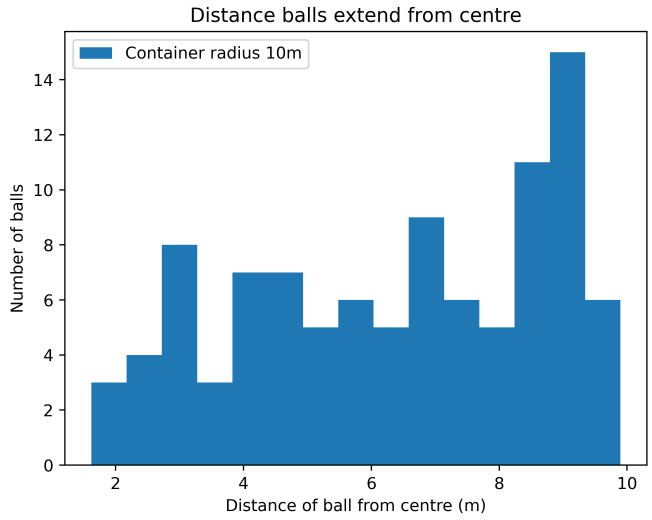

Fig. 1: Task 9– histogram of the distance balls extend from the centre. It can be seen that the balls remain within the container, mostly near the circumference. Parameters used: 300 frames, 15 bins and 100 balls of radius $0.1\,m$, mass $1\,kg$ and normal initial velocity distribution centred on $0\,ms^{-1}$, with a standard deviation of $10\,ms^{-1}$.

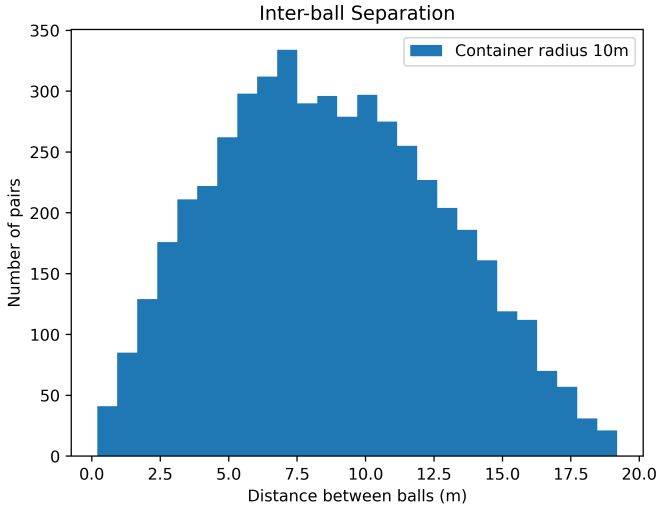

Fig. 2: Task 9– histogram of inter-ball separation. It can be seen that the balls remain within the container and mostly stay a radius apart. Parameters used: 300 frames and 100 balls of radius $0.1\,m$, mass $1\,kg$ and normal initial velocity distribution centred on $0\,ms^{-1}$, with a standard deviation of $10\,ms^{-1}$. The number of bins was determined by the Freedman-Diaconis Rule.

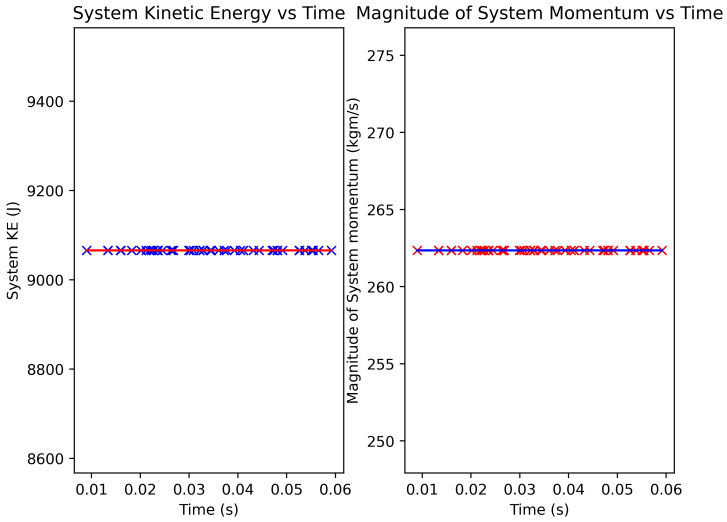

Fig. 3: Task 11– plots demonstrating the conservation of energy and of momentum in the system. Parameters used: 50 frames, container radius 10m, 100 balls of radius $0.5\,m$, mass $1\,kg$ and normal initial velocity distribution centred on $0\,ms^{-1}$, with a standard deviation of $10\,ms^{-1}$.

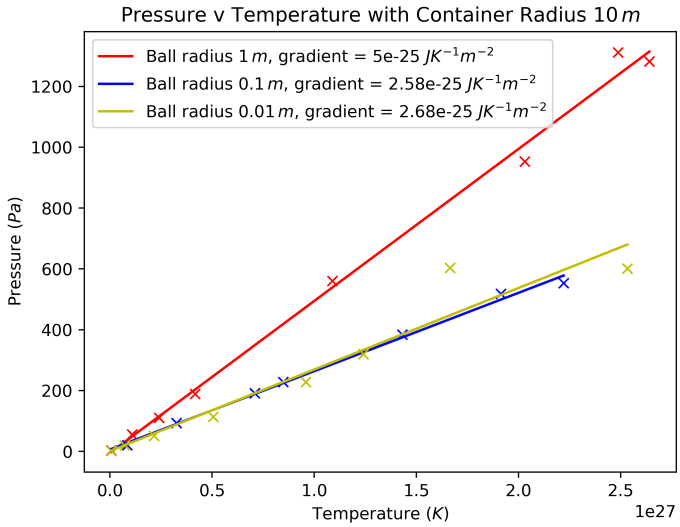

Fig. 4: Task 11 and 12– equilibrium pressure against temperature for different ball radii. Parameters used: 300 frames, 23 balls of mass $1\,kg$ and normal initial velocity distribution centred on $0\,ms^{-1}$, with a standard deviation of $10\,ms^{-1}$.

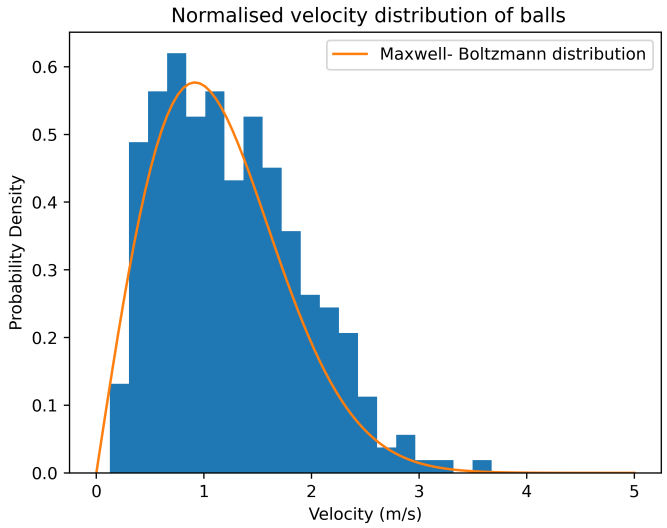

Fig. 5: Task 13– Velocity pdf with theoretical Maxwell-Boltzmann pdf for an ideal gas overlaid. It can be seen that the two plots largely agree as the ball area is negligible compared to the container area here, so the gas can be approximated by an ideal gas. Parameters used: 300 frames, 20 bins, container radius $10\,m$, 300 balls with radius $0.01\,m$, mass $1\,kg$ and normal initial velocity distribution centred on $0\,ms^{-1}$, with a standard deviation of $1\,ms^{-1}$

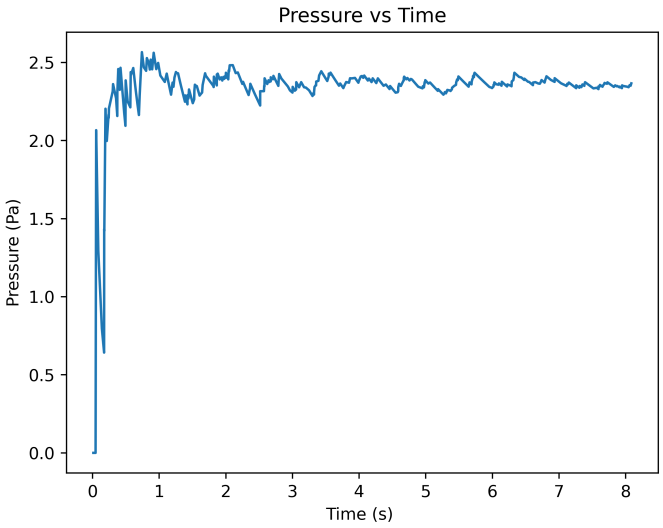

Fig. 7: Plot of pressure against time. It can be seen that the average pressure on the container becomes less volatile with time. Parameters used: 500 frames, container radius $10\,m$ and 25 balls of radius $0.5\,m$, mass $1\,kg$ and normal initial velocity distribution centred on $0\,ms^{-1}$, with a standard deviation of $10\,ms^{-1}$

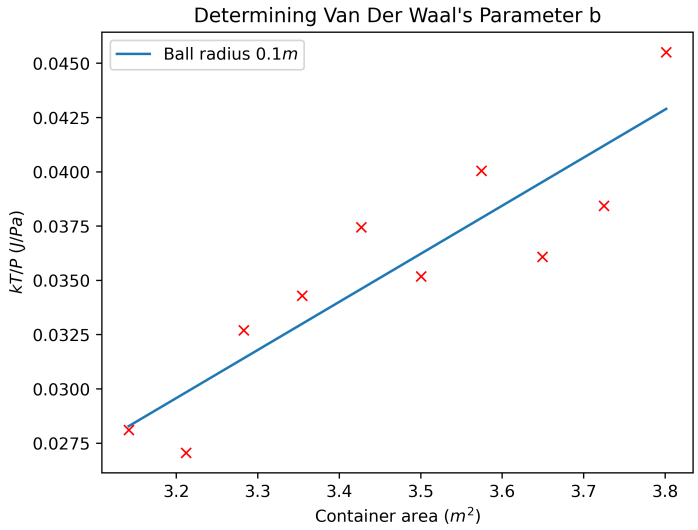

Fig. 6: Task 14– Plot of $1/P$ (Temperature is kept constant) against container area (2D volume) to determine b. 10 container radii between 10 and $10.1\,m$ were used to only use container areas that would keep the ball area relatively large enough for the gas to be non-ideal. Other parameters used: 600 frames and 23 balls (as this is the maximum number of balls this size allowed in the containers) of radius $0.1\,m$, mass $1\,kg$ and normal initial velocity distribution centred on $0\,ms^{-1}$, with a standard deviation of $10\,ms^{-1}$