Astro 507; Problem Set 1

Tom Wagg

January 20, 2022

Links for your convenience
The implementation of my simulation can be found here: https://github.com/TomWagg/thermo_winter22/blob/tom/pset1. The simulation code specifically is in `simulator.py`. All code for reproducing figures can be found in this folder under files called `problem_x.py`, where `x` is the problem number. Additionally, full size figure files are in the `figures` directory.

1a. ***Code Summary***

I initially wrote this simulation in a more object-oriented way but found that it was a tad slow and so switched to keeping track of everything in arrays and vectorising the code. The `Simulation` class handles pretty much everything and you can input the particle/box details in this Class.

I use cgs units for my variables: sizes are in centimetres, masses are in grams, times in seconds, and energies in ergs. I visualise the code using `tkinter` which a group of us set up with a lot of help from David! This can be toggled on or off by using the `visualise` parameter in the initialisation function of the class.

I determine whether a collision has occurred by checking whether these two conditions hold

$$\|\mathbf{x_1} - \mathbf{x_2}\| <= r_1 + r_2 \tag{1}$$

$$\mathbf{v_1} \cdot \mathbf{v_2} < 0 \tag{2}$$

where $\mathbf{x}$ is the position of a particle, r is the radius of a particle, $\mathbf{v}$ is the velocity of a particle and the subscript correspond to the different particles. These conditions ensure that the particles are in contact and moving towards one another.

For resolving particle collisions I solve these equations (which I got from here)

$$\mathbf{v}_1' = \mathbf{v}_1 - \frac{2m_2}{m_1 + m_2} \frac{\langle \mathbf{v}_1 - \mathbf{v}_2, \mathbf{x}_1 - \mathbf{x}_2 \rangle}{\|\mathbf{x}_1 - \mathbf{x}_2\|^2} (\mathbf{x}_1 - \mathbf{x}_2) \tag{3}$$

$$\mathbf{v}_2' = \mathbf{v}_2 - \frac{2m_1}{m_1 + m_2} \frac{\langle \mathbf{v}_2 - \mathbf{v}_1, \mathbf{x}_2 - \mathbf{x}_1 \rangle}{\|\mathbf{x}_2 - \mathbf{x}_1\|^2} (\mathbf{x}_2 - \mathbf{x}_1) \tag{4}$$

These equations essentially redistribute the velocity along the contact angle based on the initial velocities of each particle and their masses, m.

Finally, for wall collisions, I check whether a particle is within a radius of one of the edges of the box. If this is the case, I reflect the velocity component that is perpendicular to the wall.

1b. ***Snapshot Plots***

The initial and final distributions of particles are shown in Figure 1.

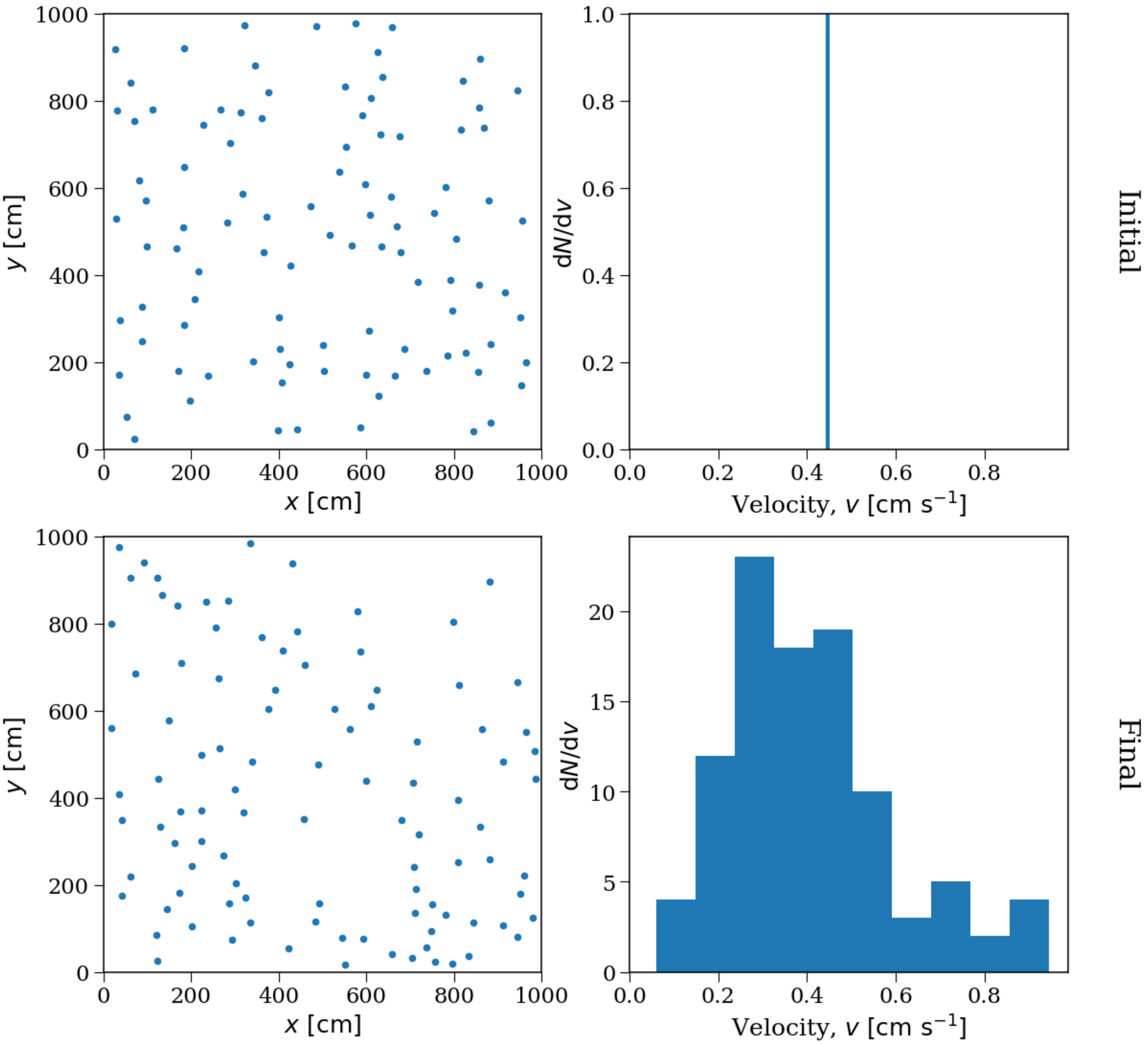

Figure 1: Initial and final distributions of the positions and velocities of the particles. The setting for this simulation were a box of size 1000cm, particles of size 20cm, initial energy of each particle of 0.1 erg and each with a mass of 1g.

Okay let's start with the given function

$$f(q,p) \propto \exp\left(-\frac{E}{k_B T}\right) \tag{5}$$

We can first use this to find the speed distribution $\frac{\mathrm{d}N}{\mathrm{d}v}$, where $v = |\mathbf{v}|$.

$$\begin{aligned} \mathrm{d}N &= A \exp\left(-\frac{E}{k_B T}\right) \mathrm{d}\mathbf{v} && (6)\\ &= Av \exp\left(-\frac{E}{k_B T}\right) \mathrm{d}v\,\mathrm{d}\theta && (7)\\ &= Av \exp\left(-\frac{mv^2}{2k_B T}\right) \mathrm{d}v\,\mathrm{d}\theta && (8) \end{aligned}$$

We can normalise this distribution since we know it must be equal to 1 and assume it is isotropic.

$$\begin{aligned} \int_0^\infty \int_0^{2\pi} Av \exp\left(-\frac{mv^2}{2k_B T}\right) \mathrm{d}v\,\mathrm{d}\theta = 1 && (9)\\ 2\pi A \int_0^\infty v \exp\left(-\frac{mv^2}{2k_B T}\right) \mathrm{d}v = 1 && (10)\\ 2\pi A \cdot \frac{k_B T}{m} = 1 && (11)\\ A = \frac{m}{2\pi k_B T} && (12) \end{aligned}$$

This implies that the final distribution (after integrating out the angles) is

$$\boxed{\frac{\mathrm{d}N}{\mathrm{d}v} = \frac{mv}{k_B T} \exp\left(-\frac{mv^2}{2k_B T}\right)} \tag{13}$$

Now I can be lazy and do a change of variables instead of going through that whole rigmarole again since we know that

$$\begin{aligned} v &= \left(\frac{2E}{m}\right)^{1/2} && (14)\\ \frac{\mathrm{d}v}{\mathrm{d}E} &= \left(\frac{2}{m}\right)^{1/2} \cdot \frac{1}{2} E^{-1/2} && (15)\\ &= \left(\frac{1}{2mE}\right)^{1/2} && (16) \end{aligned}$$

This means that we can write the energy distribution as

$$\begin{aligned} \frac{\mathrm{d}N}{\mathrm{d}E} &= \frac{\mathrm{d}N}{\mathrm{d}v}\frac{\mathrm{d}v}{\mathrm{d}E} && (17)\\ &= \left(\frac{1}{2mE}\right)^{1/2} \frac{m\left(\frac{2E}{m}\right)^{1/2}}{k_B T} \exp\left(-\frac{m\left(\frac{2E}{m}\right)}{2k_B T}\right) && (18) \end{aligned}$$

$$\boxed{\frac{\mathrm{d}N}{\mathrm{d}E} = \frac{1}{k_B T} \exp\left(-\frac{E}{k_B T}\right)} \tag{19}$$

2b. ***Show simulation approaches distribution***

In Figure 2, I show that my simulated speed and energy distributions approaches the distributions given in Equations 13 and 19. To make this plot I use a box of 300 particles of radius 3cm and mass 1g, where the box is 750cm in length and each particle starts with 1 erg of energy.

I then evolve the simulation for 10,000 seconds and save the distributions at this point. Next, to increase the number of samples I evolve it 100 more times for 100 seconds each and save the speeds and energies in each of these cases. In this way, I effectively have a sample of 30,000 particles instead of 300 and so the histogram looks nice and smooth (IMHO) :D

One can see that the simulation reproduces the analytic distributions quite nicely, fun!

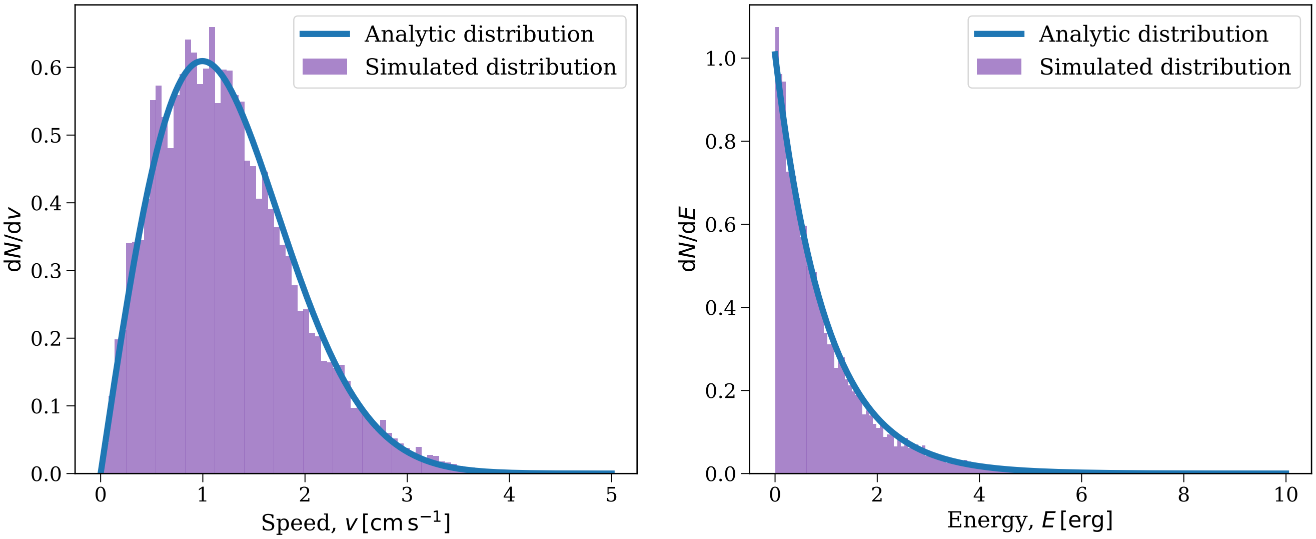

Figure 2: Plot showing the speed and energy distributions of the simulation converging to the expected analytic result. For the analytic distributions I use Equations 13 and 19.

2c. ***Multiple masses***

In Figure 3, I show the new distributions for when half of the masses are 10 times higher.

Let's talk about the speed distribution first (left side). The distribution no longer follows a single Maxwellian, but instead follows a *mixture* of two Maxwellians. The total distribution is found by summing the distribution using the v_{rms} from the lower mass and higher mass particles separately (and then normalising). This makes sense because the particles of different masses will move at different speeds when given the same kinetic energy and so they form two distributions.

The distribution for the energy is found in the same way. However, the distribution still resembles a single Maxwellian since the functional form doesn't explicitly depend on the mass. It also makes sense intuitively since each particle is given the same initial kinetic energy regardless of mass.

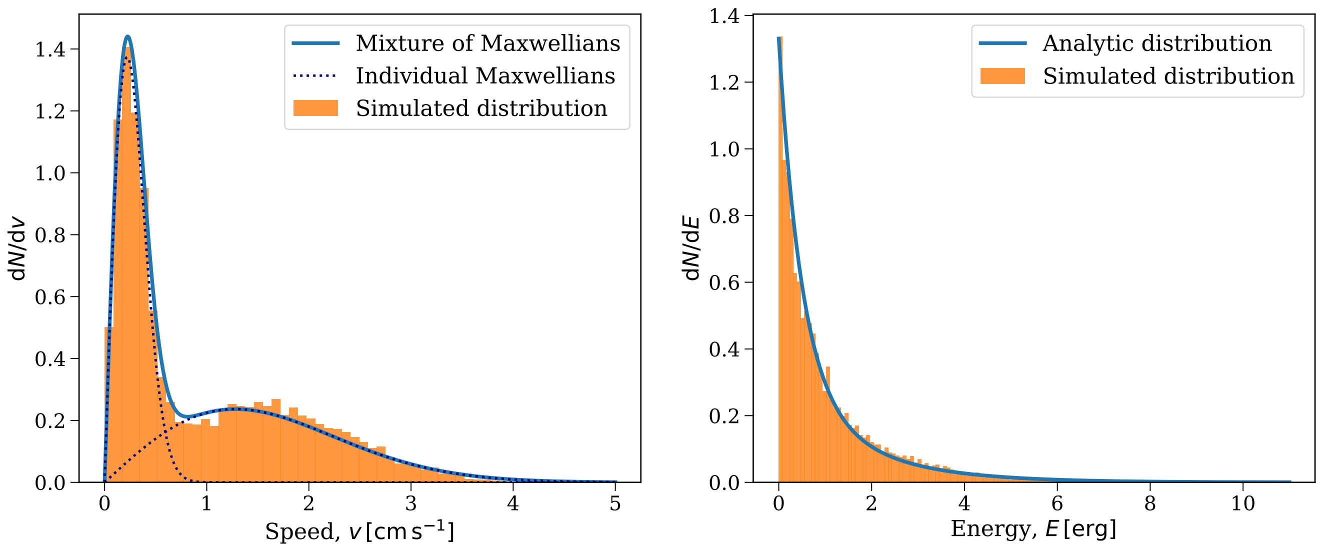

Figure 3: As Figure 2, except now half of the particles have masses of 10 g. For the speed distribution I overlay individual Maxwellians for each of the subpopulations as well as their normalised sum.

3a. ***Relaxation Criterion***

For a numerical criterion I used a K-S test on both the speed and the energy. I convert the distributions into CDFs and then used `scipy.stats.kstest` to attain p-values for both tests. Once both tests attain a p-value of at least 0.05 I assume that the system has relaxed to the steady state.

3b. ***Relaxation time derivation***

The relaxation timescale is essentially the amount of time it takes the system to forget its initial conditions. This loss of information will occur during collisions and thus we can write the relaxation timescale as a function of the number of collisions.

So, let's first derive the average time between collisions for a particle. We are given that the initial speed of the particle is v_0, its cross section is σ and the number density of particles is n. Given these values, we know that the area that the particle moves through in a time t is

$$A = \sigma v_0 t. \tag{20}$$

Then the number of collisions that occurs is this area multiplied by the number of density and it is then trivial to convert this to an average time until a collision.

$$n_{\text{collision}} = n\sigma v_0 t, \tag{21}$$

$$t_{\text{one-collision}} = \frac{1}{n\sigma v_0} \tag{22}$$

Finally, we can write that the relaxation timescale is the time for X collisions to occur and so it is

$$\boxed{\tau_{\text{relax}} = \frac{X}{n\sigma v_0}} \tag{23}$$

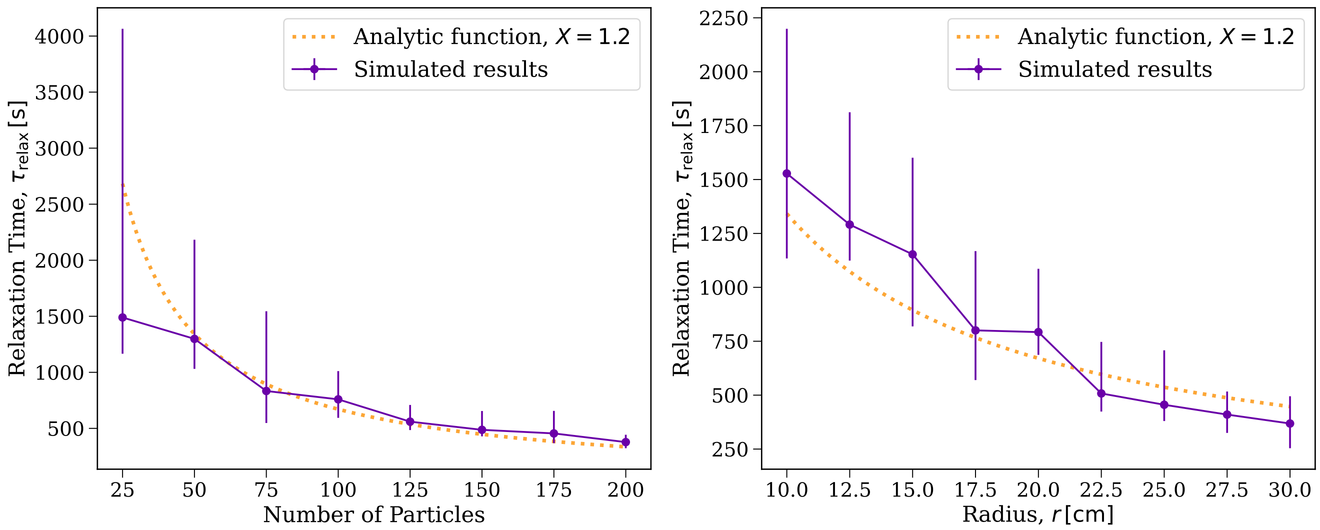

Figure 4: A comparison of the numerical criterion from 3a with the analytic formula in 3b. I use a value of $X = 1.2$ in Equation 23 since that appears to give a reasonable fit.

Deciding on the exact value of X is a more difficult decision since we would need to know how many collisions every particles needs to have to forget their initial conditions and so for now I simply state that $\tau_{\rm relax}$ scales in this way.

3c. ***Comparison with criterion***

I compare my answers in 3a and 3b in Figure 4. I found that value of $X = 1.2$ gave a reasonably good fit and so I used that in the plots. For the simulations I used the same default values as problem 1 and varied the number of particles and their size.

For each respective number of particles and radius, I ran the simulation 10 times and I plot these results as error bars spanning the full range with a point at the median of the values. You can see that the results approximately agree!

4. ***Pressure (Extra Credit)***

In Figure 5, I show that the relation $P = nk_BT$ holds for my simulation. For right-hand side we can write that

$$nk_BT = \frac{N}{A} \cdot \frac{1}{2} m v_{\rm rms}^2 \tag{24}$$

$$= \frac{N m v_{\rm rms}^2}{2L^2} \tag{25}$$

where N is the number of particles, L is the length of one side of the box, m is the mass of particles and $v_{\rm rms}$ is the root-mean-square velocity. Then for the left-hand side we can write that the average pressure on the wall of the box over the whole simulation is simply

$$P = \frac{\Delta p}{4L \cdot \Delta t} \tag{26}$$

$$= \frac{\sum_{i=1}^{N_{\rm collide}} 2m|v_{\perp,i}|}{4L \cdot \Delta t} \tag{27}$$

where N_{collide} is the number of collisions with the walls, $v_{\perp}$ is the speed in the direction perpendicular to the wall with which the particle collided and Δt is the length of time for which the simulation was run.

You can see from the bottom panel of Figure 5 that my simulation shows strong agreement between the value for P and nk_BT across many different values of N (and therefore n) and thus shows that this relation holds. For reference, I created this figure using a box of size 1000cm and particles with initial KE of 1 erg, masses of 1g and radii 5cm.

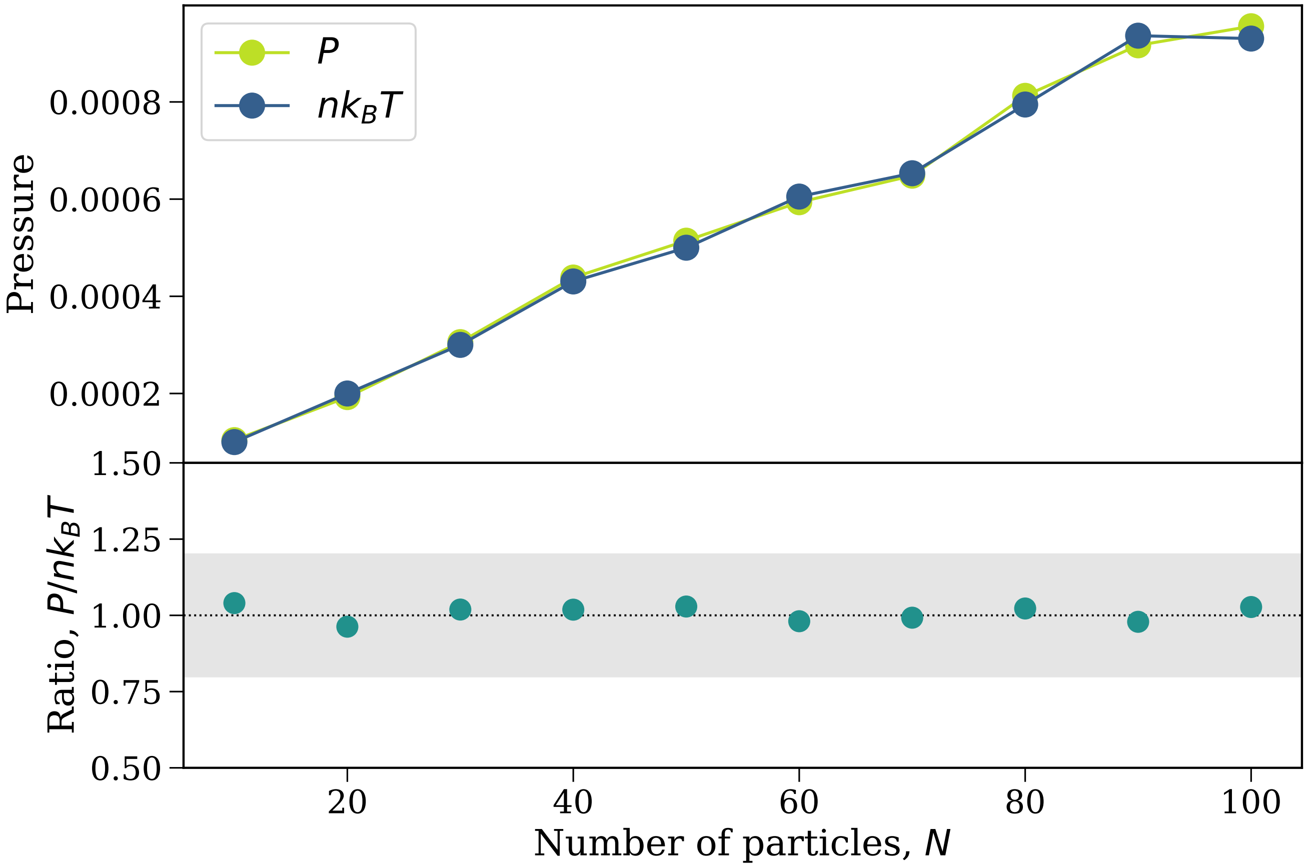

Figure 5: Demonstration that $P = nk_BT$ holds for my simulation. The top panel plots the values of P and nk_BT separately for different values of N. The bottom panel shows their ratio with a line through unity and shading to indicate a 20% difference.