

# Investigating the Kinetic Theory of Gases Using Python Simulation

Yaojia Huang & 01213992

October 30, 2018

## Abstract

The aim of this project is to simulate motions of gas molecules in a closed container through object-oriented programming in Python, and to investigate whether properties of the system accord with those described by the kinetic theory of gases. The result shows that the simulation successfully verified the Maxwell-Boltzmann distribution and ideal gas law. The behaviour of the simulated gas deviated from that described by the van der Waals equation as the container became overcrowded.

## 1 Introduction

Thermodynamics studies the laws of physics at microscopic level. However, it is hard to investigate microscopic properties of a system due to difficulties in measurement and enormous number of particles involved in it. Therefore, using computer simulations is an efficient way to investigate such systems.

In this project, the relation between different properties of gases was investigated by using Python simulation. The program was coded through object-oriented programming. Both the container and gas molecules were modelled as perfect spheres, and interactions between molecules were not considered. The results simulated by the program were compared with kinetic theory of gases such as the Maxwell-Boltzmann distribution, ideal gas law and van der Waals equation to check the consistency between them.

## 2 Theory

If a gas is allowed to evolve for a sufficiently long time in a perfectly elastic container, the speed distribution of gas molecules will eventually reach an equilibrium state [1]. In three-dimension, the probability density of a certain speed at the equilibrium state is given by the 3D Maxwell-Boltzmann distribution function, which is

$$f(v) = \sqrt{\left(\frac{m}{2\pi k_b T}\right)^3} 4\pi v^2 \exp\left(-\frac{mv^2}{2k_b T}\right), \quad (1)$$

where  $m$  is the mass of one gas molecule,  $T$  is the temperature of the gas and  $k_b$  is the Boltzmann constant.

The ideal gas law provides a relationship between the pressure, volume and temperature of gases whose

molecules have negligible size and no interactions between each other. It is expressed as

$$PV = Nk_b T, \quad (2)$$

where  $V$  is the volume of the gas and  $N$  is the number of gas molecules. Compared to the ideal gas law, the van der Waals equation is a more general equation that can be applied to real gases in most situations, and it is written as

$$\left(P - a\frac{N^2}{V}\right)(V - Nb) = Nk_b T, \quad (3)$$

where the constant  $a$  depends on intermolecular interactions and the constant  $b$  is the volume excluded by a single molecule. Since interaction forces between molecules were not implemented in the simulation, the constant  $a$  is essentially 0.

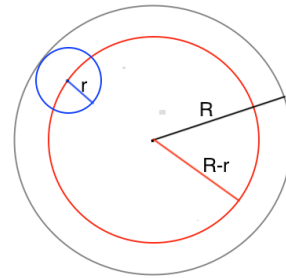


Figure 1: A model that shows a moment when a particle (blue sphere) touches the boundary of the container. The red sphere shows the actual volume of the gas in the container.

However, in the simulation, the ratio of the radius of particles to that of the container is too large; the volume of the gas is smaller than that of the container because the centre of each particle cannot reach the boundary of the container, which is made clear by Figure(1). Therefore, in this case, the van der Waals equation, with  $a$

being 0, should be

$$P = \frac{Nk_bT}{V' - Nb}, \quad (4)$$

where  $V'$  is the volume enclosed by the red sphere shown in Figure(1).  $V'$  and  $b$  are expressed as

$$V' = \frac{4\pi(R-r)^3}{3}, \quad (5)$$

$$b = \frac{1}{2} \frac{4\pi(2r)^3}{3} = \frac{16\pi r^3}{3}$$

(see p.15 in the lab book for the derivation of  $b$  [2]).

### 3 Method

The simulation was built using object-oriented programming in Python. The container and the particles were modelled as spheres and created in two separate classes. Both classes had attributes including radius, mass, position and velocity, which could be customised. The Container class inherited all methods from the Particle class so that collisions between the container and particles could be treated approximately in the same manner as those between particles.

The time needed for the next collision between two objects (two particles or one particle and the container) can be calculated using

$$(\underline{r}_1 + \underline{v}_1 \delta t - \underline{r}_2 - \underline{v}_2 \delta t)^2 = (R_1 \pm R_2)^2 [3], \quad (6)$$

where  $\underline{r}$  and  $\underline{v}$  are vector representations of the position and velocity of an object, and  $R$  is its radius. The “ $\pm$ ” sign on the RHS indicates two different types of collisions: the “+” sign is for particle-particle collisions and the “-” sign is for particle-container collisions. In order to simplify the problem, the radius of the container was treated as a negative number, so the minus sign on the RHS of the equation would no longer be needed.

The time of the next collision for every pair of objects was calculated. Since Equation(6) is essentially a quadratic equation, two solutions would be obtained for each case, but imaginary and negative solutions must be excluded because they mean that no collisions will take place. The smallest  $\delta t$  ( $\delta t_{min}$ ) represents the time after which the first collision will occur in the container.

The velocities of colliding objects before the collision were decomposed into two components, which were respectively normal ( $\underline{v}_{perp}$ ) and tangent ( $\underline{v}_{para}$ ) to the surfaces of the objects at colliding point. The normal velocity components of the objects were used to calculate their centre-of-mass velocity ( $\underline{v}'_{com}$ ) in that direction, which is

$$\underline{v}'_{com} = \frac{m_1 \underline{v}_{1perp} + m_2 \underline{v}_{2perp}}{m_1 + m_2}. \quad (7)$$

For a perfectly elastic collision,  $\underline{v}_{perp}$  would be inverted with respect to  $\underline{v}'_{com}$ , and  $\underline{v}_{para}$  would remain unaltered,

so the velocity of each particle after the collision would be

$$\begin{aligned} \underline{v}_{new} &= \underline{v}_{para} + \underline{v}'_{com} - (\underline{v}_{perp} - \underline{v}'_{com}) \\ &= \underline{v}_{para} + 2\underline{v}'_{com} - \underline{v}_{perp} \end{aligned} \quad (8)$$

(see pp.6-7 in the lab book for details [2]).

The simulation operates mainly in a while-loop nested in a third class called the Gas class, which is the daughter class of both the Container and Particle classes. The loop firstly finds  $\delta t_{min}$  at a certain moment, then the whole system moves for a time interval approximately equal to  $\delta t_{min}$ . After that, the Collide method is applied to the objects that should collide at that time, and the velocity attributes of the two objects are updated. The cycle repeats again and again so that the system evolves with time. This algorithm is very time-efficient if the frequency of collisions is not too high, and it ensures that every collision occurs at the instant when two objects are approximately in contact with each other however high their speeds are.

The pressure of the gas is needed for comparison with the kinetic theory of gases. Since force is the change of total momentum per unit time, the change of momentum of each particle after colliding with the container is recorded. The pressure is obtained by dividing the sum of momentum changes by the running time of the simulation and the surface area of the container.

### 4 Results and Discussion

To investigate the speed distribution of particles, 50 nitrogen molecules were simulated in a container with radius of  $10m$ . The mass and initial kinetic energy of each particle were set to be  $4.651e-26kg$  and  $6.213e-21J$ , which are the mass of one nitrogen molecule and its average kinetic energy at room temperature (300K). The radius of particles was  $0.5m$ , which ensures sufficient number of collisions.

Since 50 molecules were not enough to make a speed-distribution histogram, their speeds were recorded every  $0.001s$  between  $5s$  and  $20s$  (the beginning  $5s$  allowed the system to approximately reach the equilibrium state) with the premise that the ergodic hypothesis, which states that the time average of a Hamiltonian system is the same as its space average, is valid [4]. Therefore, the sample size became 750000.

The resultant histogram of speed distribution is normalised and fitted with Function(1), in which  $T$  is the unknown parameter that needs to be determined. The graph is shown in Figure(2).

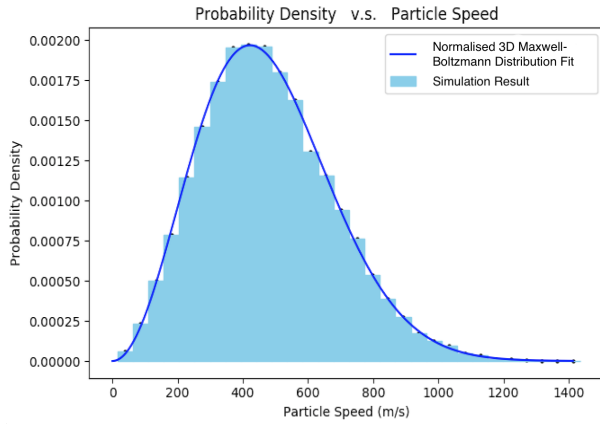


Figure 2: A 30-bin histogram of speed distribution of 50 nitrogen molecules. The speed of each molecule was recorded every  $0.001s$  within a  $15s$  interval. It is fitted with the Maxwell-Boltzmann distribution function.

The distribution function fits the histogram almost perfectly. The parameter,  $T$ , is determined to be  $300.1 \pm 1.6K$  compared to the theoretical value of  $300K$ . It shows that the simulation result agrees with the Maxwell-Boltzmann distribution.

Then, the simulation was executed several times by changing the radius of particles while keeping the size of the container constant. The procedure repeats for containers of different sizes.

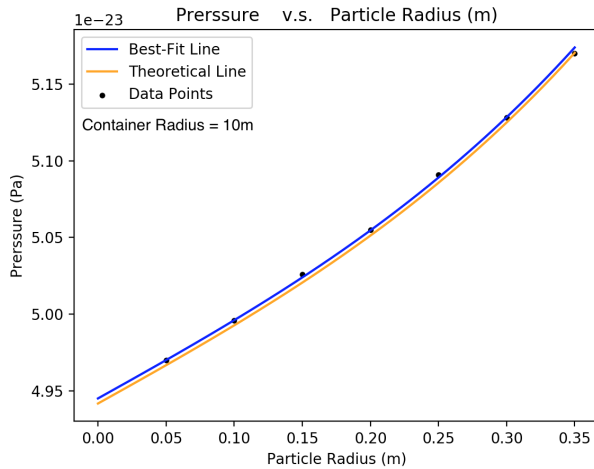


Figure 3: Plot of pressure v.s. particle radius in a container with radius  $10m$ . The orange line is the theoretical plot of Equation(4) with  $T = 300K$ . The best-fit line is very close to the theoretical line.

Figure(3) is a plot of pressure v.s. particle radius in a container with a radius of  $10m$ . The data were fitted with Equation(4), in which  $T$  is also the undetermined parameter. The result gives  $T = 300.2 \pm 0.04K$ . Though  $300K$  is not within the error range, it is close enough considering the sample size is small compared to real gases. Therefore, the simulation proves the validity of

the van der Waals equation to some degree.

However, the data deviate from Equation(4) as the container becomes overcrowded. Figure(4) is also a pressure v.s. particle radius plot, but this time particles are in a container with radius  $3m$ . It shows that the bigger the molecules are, the larger the deviation between simulation results and theoretical values is.

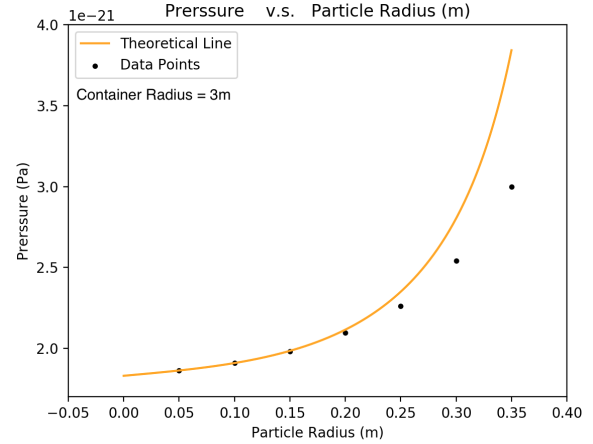


Figure 4: Plot of pressure v.s. particle radius in a container with radius  $3m$ . The data deviate from theoretical values as the radius of molecules becomes large.

Such deviation is expected because the van der Waals equation loses accuracy when molecules are so compact that the gas phase is indistinguishable from the liquid phase [5]. As long as simulation results converge to theoretical values as the molecule size approaches 0, the van der Waals equation still holds for most real gases.

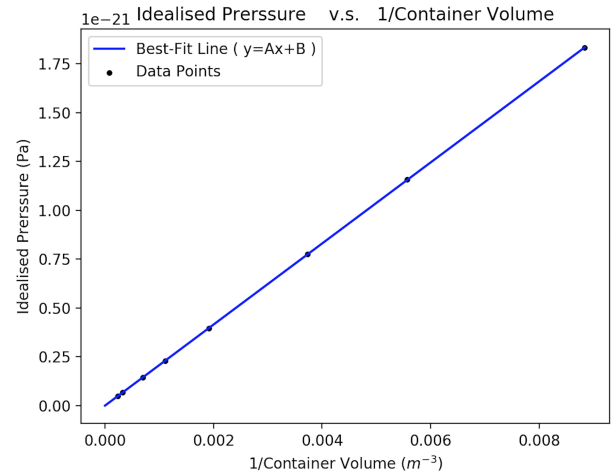


Figure 5: Plot of pressure v.s.  $1/\text{container volume}$  for ideal gas. The data are fitted with the ideal gas law.

The multiplicative inverse of a polynomial function is used to fit data sets that cannot be fitted by the van der Waals equation. The pressure of the ideal gas can be estimated by finding the value of the fitting

function at particle radius = 0. The estimated pressure v.s.  $1/(\text{container volume})$  is plotted in Figure(5). The resultant slop and interception are respectively  $(2.075 \pm 0.002)e-19 m^2 kg s^{-2}$  and  $(3.462 \pm 9.703)e-25 Pa$  compared to the theoretical value of  $2.072e-19 m^2 kg s^{-2}$  and 0 according to Equation(2), so the ideal gas law is also verified.

## 5 Conclusion

The purpose of this project is to create a simulation of gas-in-container system in Python using object-oriented programming. The simulation operates by repeatedly finding the time of the next collision, making the system evolve to that moment and then changing the velocities of colliding objects. The simulation was used to test the kinetic theory of gases. The observed speed distribution of gas molecules and the relationship between pressure, volume and temperature agree with the Maxwell-Boltzmann distribution for 3D gas and the ideal gas law, and it also shows that the van der Waals equa-

tion only holds for gases that are not close to the phase transition point.

## References

- [1] Maxwell, J. (1860). Illustrations of the dynamical theory of gases. London: The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, pp.22-27.
- [2] Y. Huang. 2nd Year Lab Book. 2017
- [3] Year 2 Computing Projects. Project B: Thermodynamics Snookered. Physics Dept. Imperial College London.
- [4] Oliveira, C. and Werlang, T. (2007). Ergodic hypothesis in classical statistical mechanics.
- [5] Griffith, T. (2016). Basic principles and calculations in process technology. New York: Prentice Hall, pp.113-114.