Emergence of Statistical Behavior in a 2D Molecular Dynamics Simulation

Sathishkumar A
[a] *Department of Physics, Indian Institute of Technology Madras, Chennai"*

ARTICLE INFO

Keywords:
Speed distribution
Ideal gas
Entropy
Grand Canonical Ensemble

ABSTRACT

The work presents the ideal gas molecular dynamics simulation in a confined 2D box to study how the statistical behaviours and its principles are arising from the classical Newtonian dynamics. The particles are modelled as ideal gas discs undergoing elastic collision with each other and reflective collisions with the boundaries of box. The velocity of different particles is recorded over time to examine their statistical behaviour.

Introduction

This report is written as an enthusiastic project to understand the stochastic nature of particles in real life which can be seen from Newtonian Dynamics. In this simulation, randomly placed 10,000 particles are allowed to elastically collide with each other in ideal situation where there are no inter molecular and external force. From this simulation, we want to prove that the nature always loves to make a stable relationship (equilibrium) with maximum disorder.

Initial Simulation Setup

This small world has a population of 10,000, with length of 4000 meters, the population of particles is taken in the way needs to go too far to see another fellow particle. The Density of the particles is so low that, initially there are very few collisions that can possibly happen at a time.

$$\rho = \frac{10000}{(4000)^2} = 6.25 \times 10^{-4}\ NL^{-2}$$

Packing fraction of the setup,

$$\omega = \left(\frac{\pi * 0.7^2}{4000^2}\right) \approx 9.621 \times 10^{-8}$$

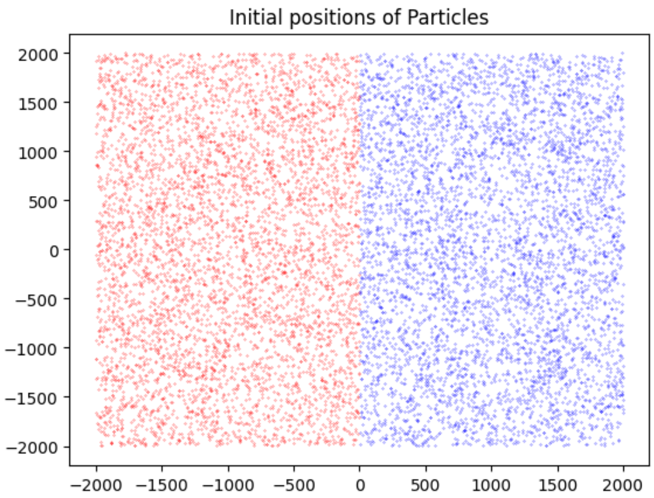
The Position of each particle is generated as a tensor of Rank 2 with dimension of 10000×2 with radius of 0.7 m. For some visualization purposes, particles with positions satisfying $x < 0$ are categorized as left-side particles and are colored red, while particles with $x \geq 0$ are categorized as right-side particles and are colored blue.

Mass of single particle is taken identical to a hydrogen atom,

$$m = 10^{-24}\ kg$$

The Right sided particles have the velocities directed along negative x axis, given by

$$v_{0r} = -v_{0rx}\, e_i\ ; \qquad v_{0rx} \in [499,500]$$

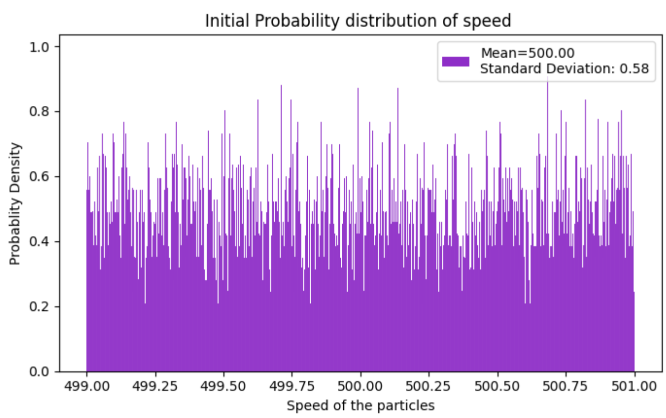


Note: The size of particles is enlarged for visualization purposes and it doesn't reflect the real parameters of the particles in simulation.

The Left sided particles have the velocities directed along positive x axis, given by

$$v_{0l} = v_{0lx} e_i$$

$$v_{0rx} \in [500,501]$$



Time evolution Algorithm

Position update

The positions of all particles are updated using their current velocities according to

$$r(t + \Delta t) = r(t) + v \cdot \Delta t$$

Wall Collisions

Particles that reach the boundaries of the box undergo elastic reflections, reversing the velocity component perpendicular to the wall.

Particle–Particle Collisions

Pairs of particles whose separation becomes smaller than twice the particle radius are considered to collide. Their velocities are updated using the conservation of linear momentum and kinetic energy, assuming perfectly elastic collisions.

$$\vec{v}_r = \vec{v}_1 - \vec{v}_2 \quad ; \quad \vec{r}_r = \vec{r}_1 - \vec{r}_2$$

$$\vec{v}_1' = \vec{v}_1 - \frac{(\vec{v}_r \cdot \vec{r}_r)(\vec{r}_r)}{|\vec{r}_r|^2}$$

$$\vec{v}_2' = \vec{v}_2 + \frac{(\vec{v}_r \cdot \vec{r}_r)(\vec{r}_r)}{|\vec{r}_r|^2}$$

Simulation Procedure

Computational Procedure

The simulation and data analysis were performed using Python. Numerical computations were carried out using NumPy, Pytorch, while data handling and visualization were performed using Pandas and Matplotlib. Animations and trajectory visualizations were generated using Plotly. The position and velocity of every particle are stored in Pytorch for the GPU utilization.

Relative Position

Computation for this simulation starts with the combinations of all the particles in the box. Pairing 2 particles which can be done by labelling an index to each particle.

$$i = [0, N - 1]$$

The relative positions of all particles with respect to one particle n_i are found by this combination,

$$^{10000}C_2 = 4.9995 \times 10^7$$

Relative distance of all the particles to can be found by comparing 50 million pairs position and its velocity for a millisecond. For this much of heavy computation, Pytorch tensor is used to utilize the 'cuda' architecture.

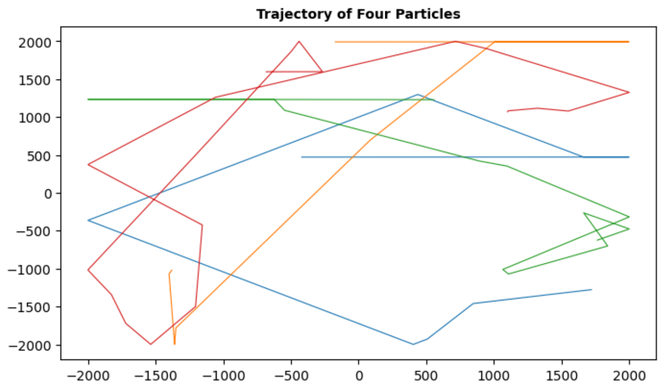
Collision Tracker

By comparing the Relative position for each pair, if the collision condition is true, then the particles are repelled with new velocity with collision equation.

$$|\vec{r}_r| \leq 2r$$

Position Tracker

Position of four randomly taken particles are tracked as an array which has discrete datapoints for 25 seconds with interval of a milli second.



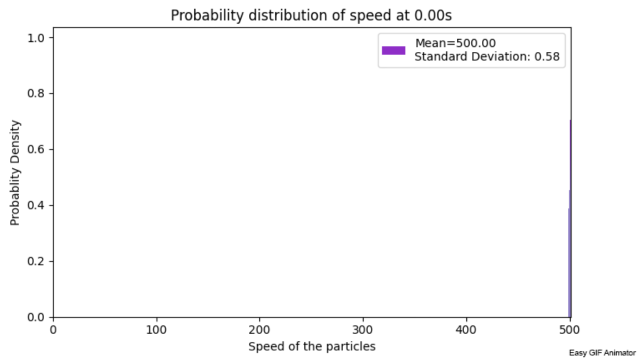
Speed Tracker

The speed is tracked for 25 seconds with interval of a milli second to study the speed, Temperature, energy, entropy evolution in the system

Inference

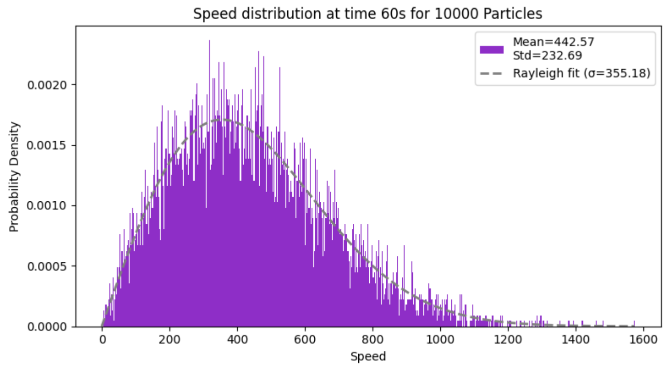
Speed Distribution

In this simulation, the initial velocities and speeds are comparatively uniform (minor deviations for the velocities were added in x direction to prevent python from throwing error for skewed data to plot histogram). So, initially the mean of the speed data is approximately equal to 500 m/s and standard deviation is approximately equal to 0. The motto of the whole project is to analyze the time evolution of speeds until it gets to the equilibrium.



The mean of the speed distribution decreases as time goes on, standard deviation gets increases, it emphasizes the second law of thermodynamics,

> "Disorder of an isolated system always increases over time, or remains constant in ideal, reversible cases."



From the observations, we can conclude that velocity distribution follows the Rayleigh Distribution for the speeds in equilibrium. General form of Rayleigh Distribution can be given as,

$$f(v) = \frac{v}{\sigma^2} e^{-\frac{v^2}{(2\sigma)^2}}$$

Where "σ" is the scale factor of Rayleigh's Distribution,

$$\sigma^2 = \frac{2}{4-\pi} Var(v)$$

Time (s)	Mean	Standard Deviation
0	500	0.58
0.5	499	9.98
1	499.21	28.15
10	465	183.63
20	450.26	217.41
30	445.72	226.57
40	443.32	231.24
50	443.78	230.34
60	442.56	232.68

The values shown above are the standard deviation and mean values of speed distribution over different time period.

Entropy Evolution

Entropy of a system for an isolated system is defined as,

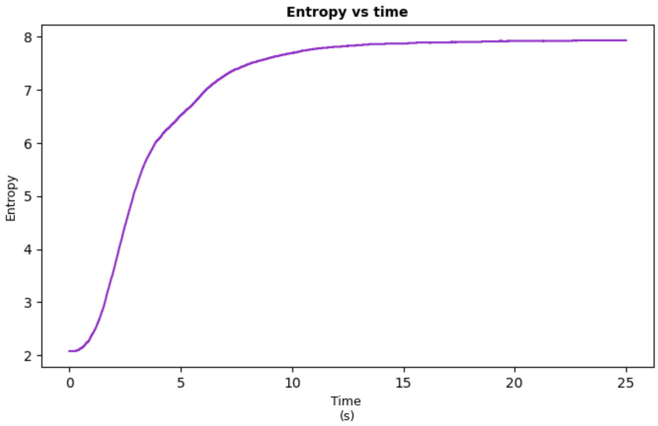
$$S = k \ln(\,\Omega\,)$$

As our sample size is too high to compute all the possible microstates for this system, the Gibb's formula of entropy is used.

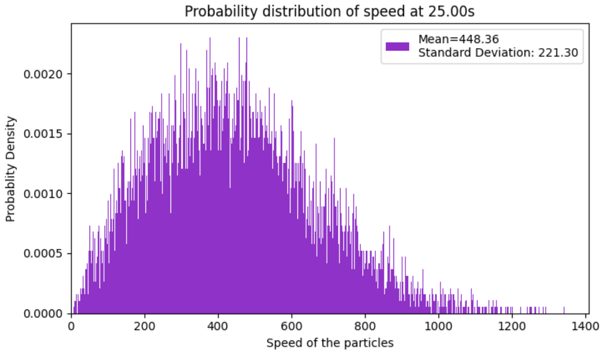
$$S = \sum_i p(v_i) \ln(p(v_i)$$

A normalized Histogram is made in range of 0 to 4000m/s with bin width (Δv) of 2000. Probability of finding the particle at particular time in the between of $v_i$can be approximated as

$$p(v_i) = H(\,v_i)dv$$



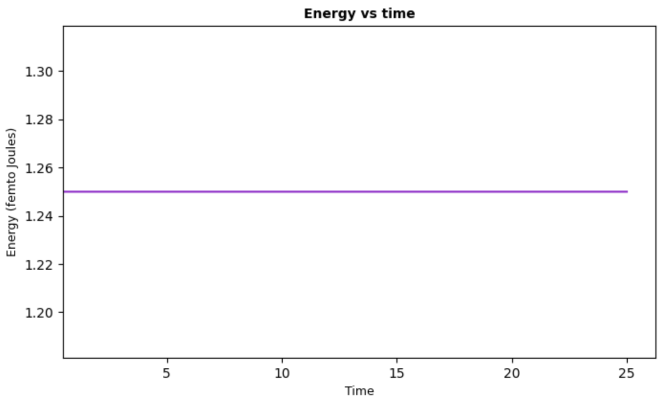
As collisions become more frequent, the distribution spreads and entropy increases. Eventually, the entropy reaches a steady value, indicating that the system has reached equilibrium.



The figure shown above is the distribution of the speeds in 25th second, the distribution shape has been remained same until end of the simulation.

Energy and Temperature Evolution:

From the speeds of each particle, we can find the Kinetic energy of the system. One of the key assumptions of this project is 'Particles are ideal, and participating in elastic collisions. The velocity evolution is built on law of conservation of momentum and kinetic energy. Tracking energy of each particle should lead to a flat line if our simulation is conserving total energy.



In classical statistical mechanics, temperature is proportional to the average kinetic energy of the particles. Therefore, the conservation of kinetic energy naturally leads to a constant temperature.

$$\langle K.E \rangle = constant$$

$$k_b T = \langle K.E \rangle = constant$$