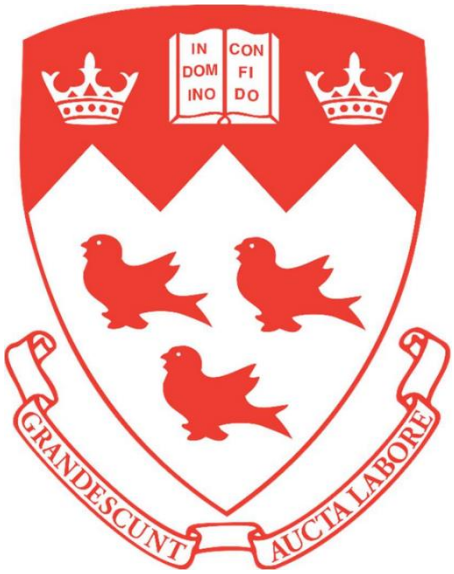
# McGill University

## Department of Mechanical Engineering




# MECH 578 2D Gas Simulation and Validation of Maxwell Boltzmann Distribution

Individual Final Report Part I

Zongxuan Lin 260780979

October 6th, 2022

## Contents

# Derivation of 2D Maxwell Distribution

According to Maxwell, the distribution of gas speed follows normal distribution centered at zero velocity. $f(Cx) = A * e^{-B*(Cx)^2}$.

Known $\int_{-\infty}^{\infty} f(Cx)dCx = 1$ (1) and $\int_{-\infty}^{\infty} e^{-B*(Cx)^2}dCx = \sqrt{\frac{\pi}{B}}$ (2)

Equate (1) and (2), we get $A * \sqrt{\frac{\pi}{B}} = 1$, so $A = \sqrt{\frac{B}{\pi}}$ .

The Distribution function becomes $f(Cx) = \sqrt{\frac{B}{\pi}} * e^{-B*(Cx)^2}$

Next step would be to deduce the value of RMS speed in one direction. Known that the molecules are balls with 2 degrees of freedom. The energy should be evenly distributed in each direction. Thereby, we have $P * V = \frac{1}{2} m_{tot} * C_{rms}^2$. Thus, $P = \frac{1}{2}\rho * C_{rms}^2$. Thereby, $C_{rms}$ is Calculated. $C_{rms}^2 = \frac{2P}{\rho}$.

Since kinetic energy is evenly distributed, so is the $C_{rms}$ in x and y direction.

$C_{rms,x}^2 + C_{rms,y}^2 = C_{rms}^2 = \frac{2P}{\rho}$ (3) $C_{rms,x}^2 = C_{rms,y}^2$ (4)

Solve above equations, and $C_{rms}$ in x and y direction are calculated: $C_{rms,x}^2 = C_{rms,y}^2 = \frac{P}{\rho}$

Now calculate $C_{rms}$ based on the distribution function we have. $\int_{-\infty}^{\infty} f(Cx) * (Cx)^2 dCx = C_{rms,x}^2$

Known $\int_{-\infty}^{\infty} e^{-B*(Cx)^2}dCx = \frac{1}{2} * \sqrt{\frac{\pi}{B^3}}$ .

Thus, $\int_{-\infty}^{\infty} f(Cx) * (Cx)^2 dCx = \sqrt{\frac{B}{\pi}} * \int_{-\infty}^{\infty} e^{-B*(Cx)^2}dCx = \sqrt{\frac{B}{\pi}} * \frac{1}{2} * \sqrt{\frac{\pi}{B^3}} = \frac{1}{2B}$

Therefore, $C_{rms,x}^2 = \frac{1}{2B} = \frac{P}{\rho}$. B is calculated to be: $B = \frac{\rho}{2P}$

From course note: $P = \frac{NK_BT}{V} = \frac{\rho K_B T}{m}$. N is the total number of molecules, and m is the mass per molecule. $K_B$ is the Boltzmann Constant; T is the temperature of the environment.

Thus, $B = \frac{\rho}{2P} = \frac{m}{2K_BT}$

**ANS:** $\boldsymbol{f(Cx) = \sqrt{\frac{m}{2\pi K_B T}} * e^{-\left(\frac{m}{2K_BT}\right)*(Cx)^2}}$

**ANS:** $\boldsymbol{f(Cy) = \sqrt{\frac{m}{2\pi K_B T}} * e^{-(\frac{m}{2K_BT})*(Cy)^2}}$

Time the 2 equations together we get: $f(Cx) * f(Cy) = \frac{m}{2\pi K_B T} * e^{-\left(\frac{m}{2K_BT}\right)*((Cy)^2+(Cx)^2)}$

Known that $dxdy = cdcd\theta$, we transform the above equation into the following equation.

$$X(c)dc = \int_0^{2\pi} \frac{m}{2\pi K_B T} * e^{-\left(\frac{m}{2K_BT}\right)*C^2} cdcd\theta = \cancel{2\pi} * \frac{m}{\cancel{2\pi} K_B T} * e^{-\left(\frac{m}{2K_BT}\right)*C^2} cdc$$

**ANS:** $X(c) = * \frac{m*c}{K_B T} * e^{-\left(\frac{m}{2K_B T}\right)*C^2}$

Hence the speed Maxwellian distribution of 2d gas is deduced.

# Design and Improvement of Simulation Algorithm

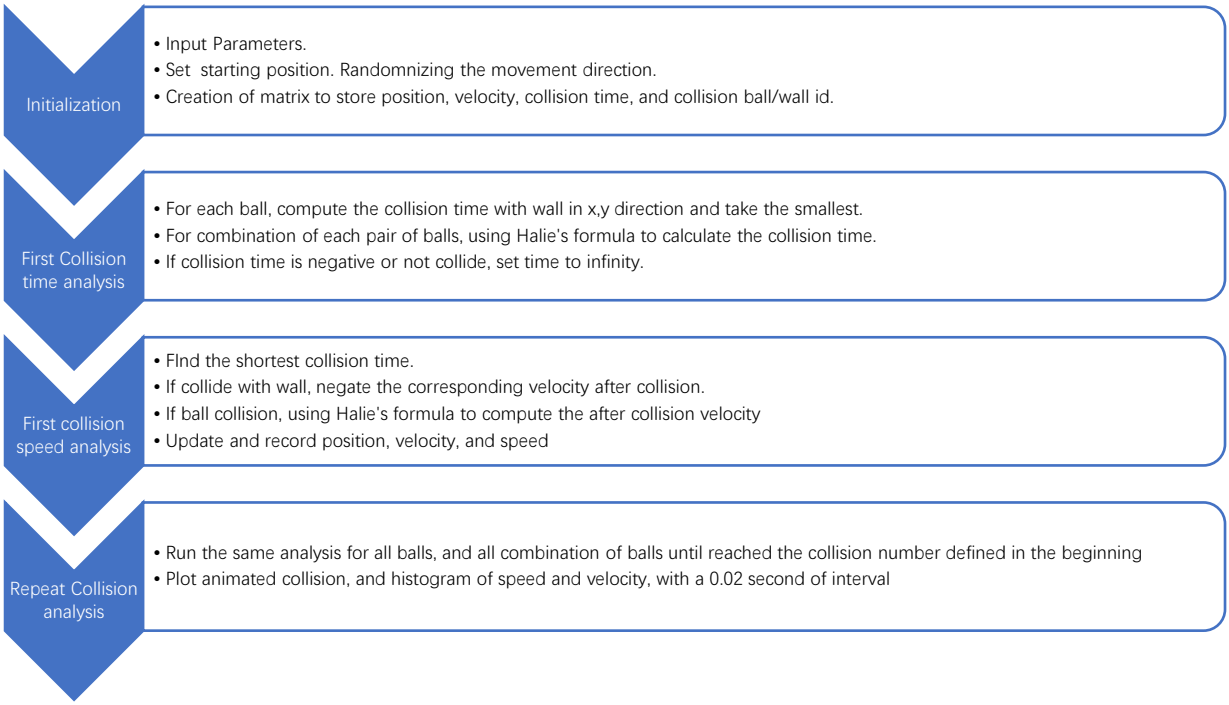



**Figure.1 Beta version Algorithm**

This algorithm does give accurate simulation in the beginning of simulation. However, after a certain number of cycles, the algorithm would stuck into infinite loop at a certain collision using Lulu's blog's formula. With Halie's formula, the balls seem to be sticking with each other after some time. The beta version algorithm is also quite slow to run, it takes over an hour to run 1000 simulation of collisions.

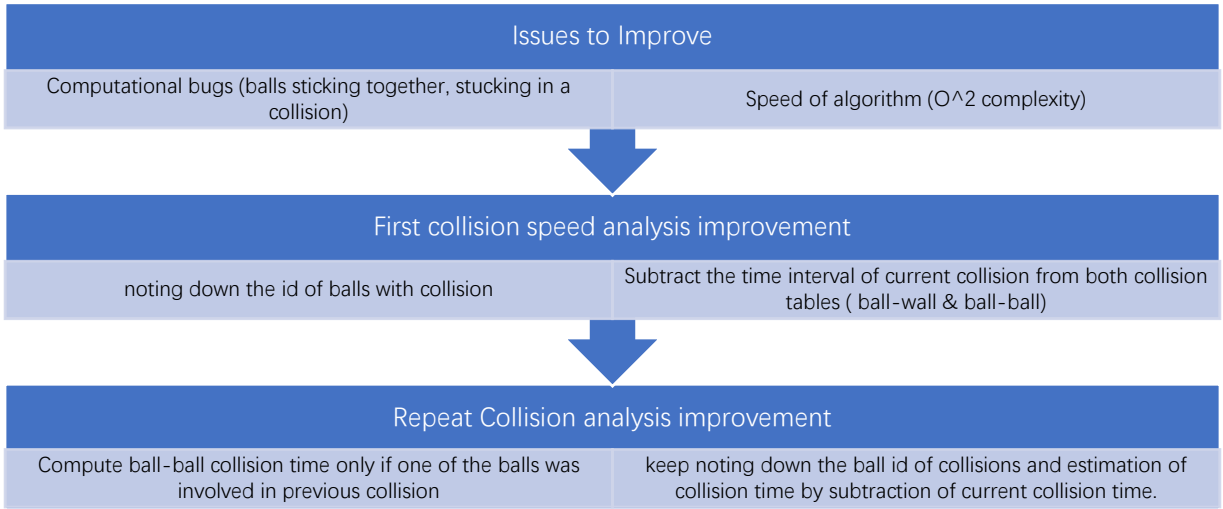



**Figure.2 Algorithm Updates to Improve Speed.**

Figure.2 illustrate the algorithm of improved speed. Apart from the process mentioned in the figure, all the other process are the same as the previous algorithm. The speed of this algorithm is 100 times faster than before, taking only less than a minute to calculate 10000 collisions.

However, using subtractions to update a table of floating-point numbers cause error when the number is subtracted to close zero. Double precision floating point number could lose precision when it gets closer to zero. As a result, many balls that are supposed to collide in a short period of time pass through each other. In the simulation, the code takes a very long time to get 10000 collisions since it does not identify all the collision happened in the simulation.

There are 2 ways to improve solve the problem, one is to use symbolic math for calculation of each collision, which would take more RAM, and slow down the speed in a significant way. The other one is to check if each pair of molecules overlap each other after updating their position. In such a way, we should be able to find out the collisions that are happening simultaneously while not slowing down the simulation significantly.

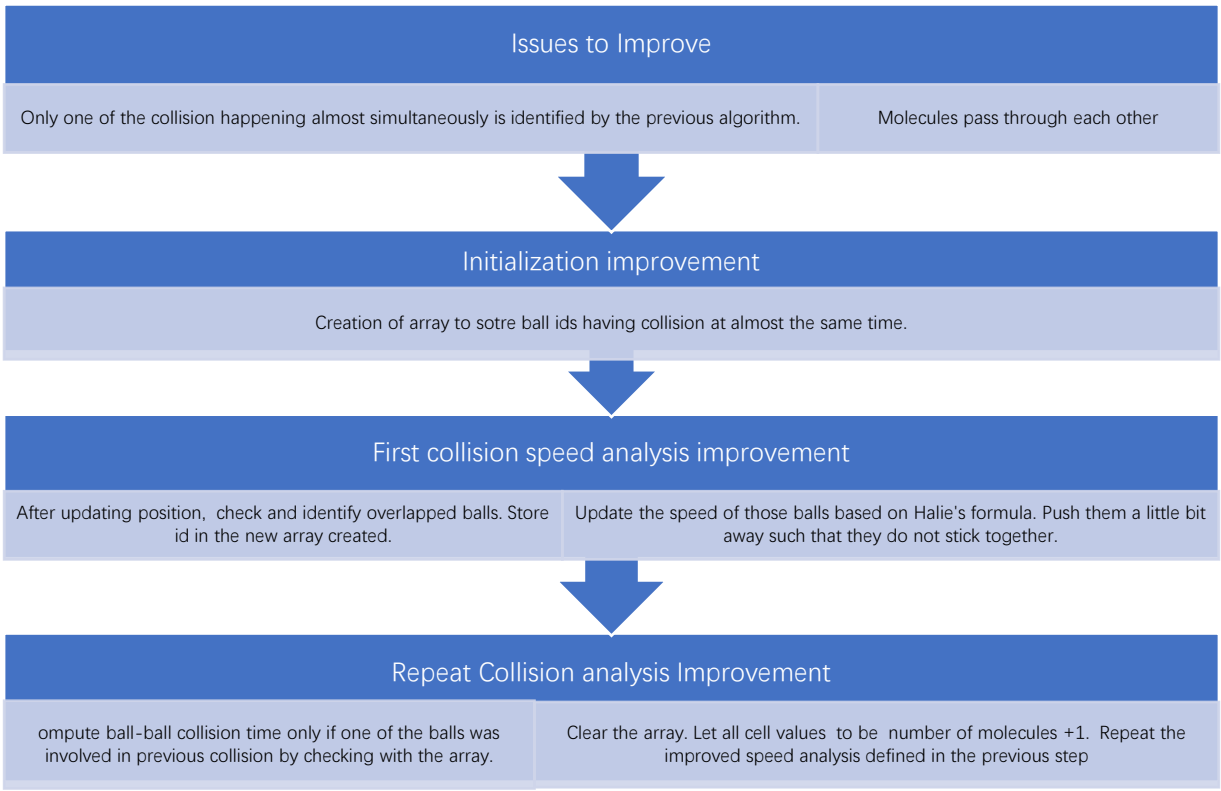



**Figure.3 Algorithm Updates Fixing Calculation of 'Simultaneous' Collision**

This final version of the simulation algorithm gives accurate result. It is more stable in the long simulation since it avoids the molecules passing through each other problem of the previous version. The molecule would not stick to each other as in the beta version. The speed of the algorithm is slowed down a bit. It takes 10 to 20 minutes to run 10000 simulations with this algorithm.

**Table. 1 Summary of Speed of Developed Algorithms**

| | Beta Version | 2nd Version | Current Version |
|---|---|---|---|
| Time Complexity | O^2 | O | O(log(O)) |
| 10000 simulations time | >1h | 1 minute | 10 ~ 20 minute |

# Simulation Results

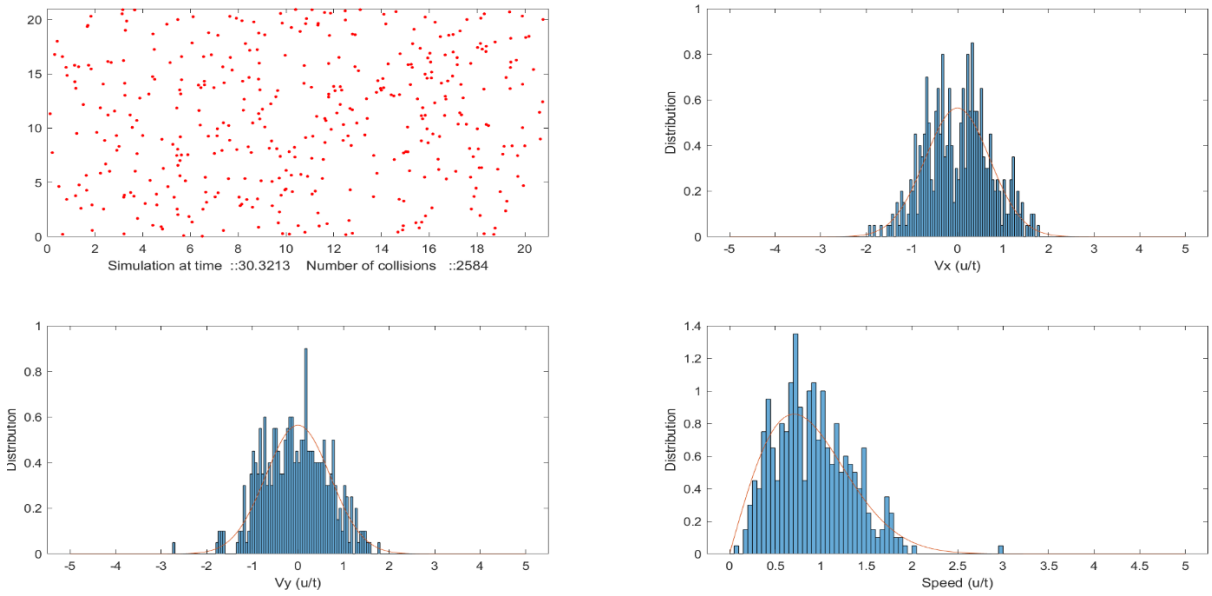



**Figure.4 Condition at 30.3213s**

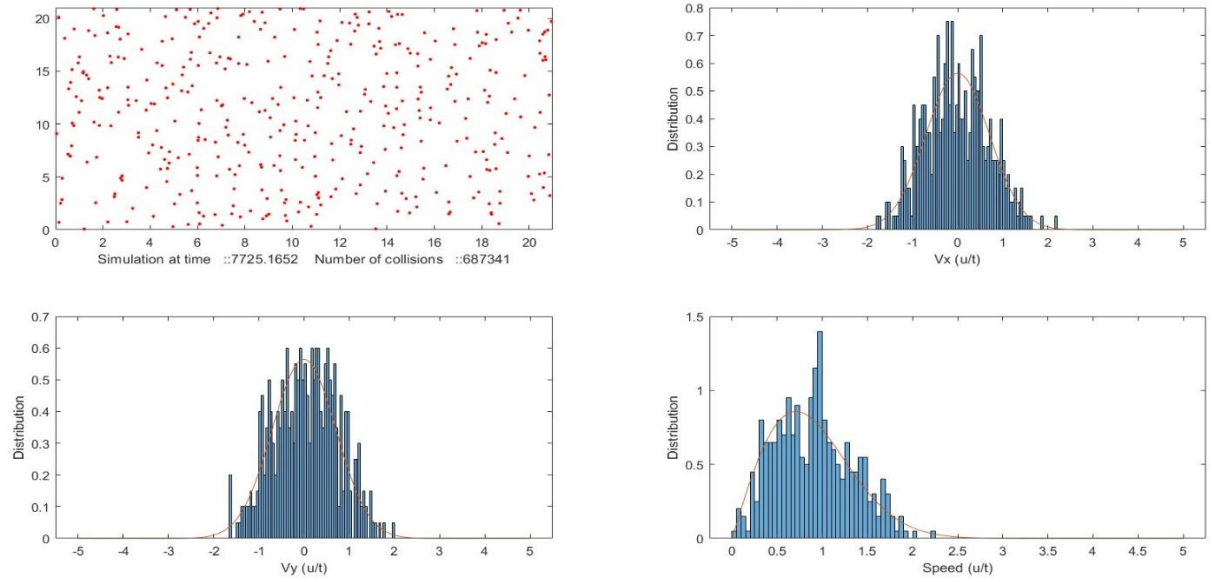



**Figure.5 Final Condition after 687341 collisions**

According to the simulation, the velocity and speed distribution of the molecules gradually approaches the Maxwellian distribution derived in step 1 from the beginning to 30s. Afterwards, the speed and velocity distribution of the molecules oscillates between around the Maxwellian distribution, but never perfectly reach the precise Maxwellian distribution. This is because we only have 400 molecules, but 200 intervals for velocity distribution and 100 intervals for speed distribution. One collision could cause spike and grooves in the histogram and making it to not perfectly match the Maxwellian distribution curve.

Thus, the time for molecule to reach Maxwellian distribution is 30.3213s. There are a total of 2584 collisions. In other words, it requires on average 6.46 collisions per molecule for 400 molecules to reach Maxwellian distribution.

# Reference

*[1] Elastic collision - equations and simulation*. Elastic collision - Equations and simulation | Lulu's blog. (n.d.). Retrieved October 6, 2022, from https://lucidar.me/en/mechanics/elastic-collision-equations-simulation/

[2] J. M. Haile, "Molecular Dynamics Simulation: Elementary Methods," John Wiley & Sons, New York, 1992.

J. M. Haile, "Molecular Dynamics Simulation: Elementary Methods," John Wiley & Sons, New York, 1992.

# Appendix

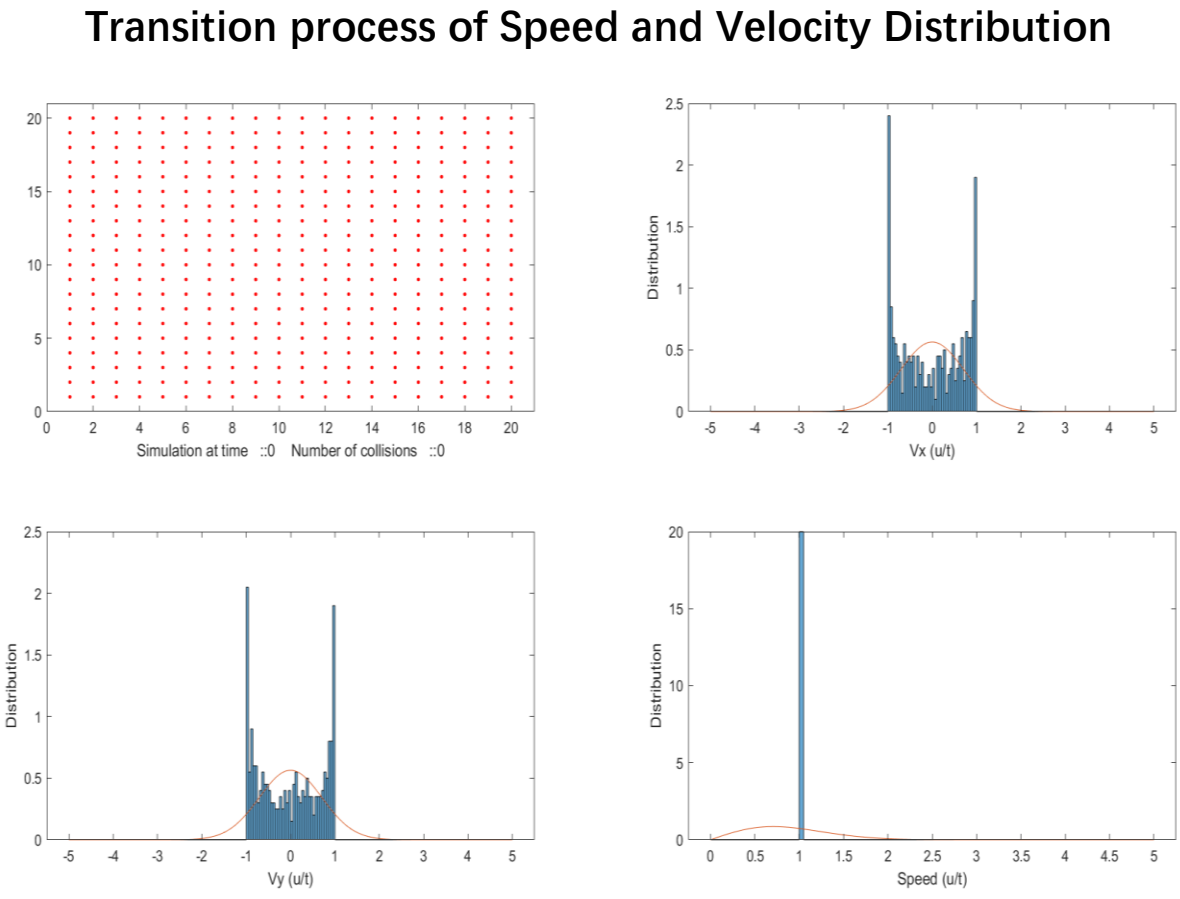



**Figure.6 Initial Condition**

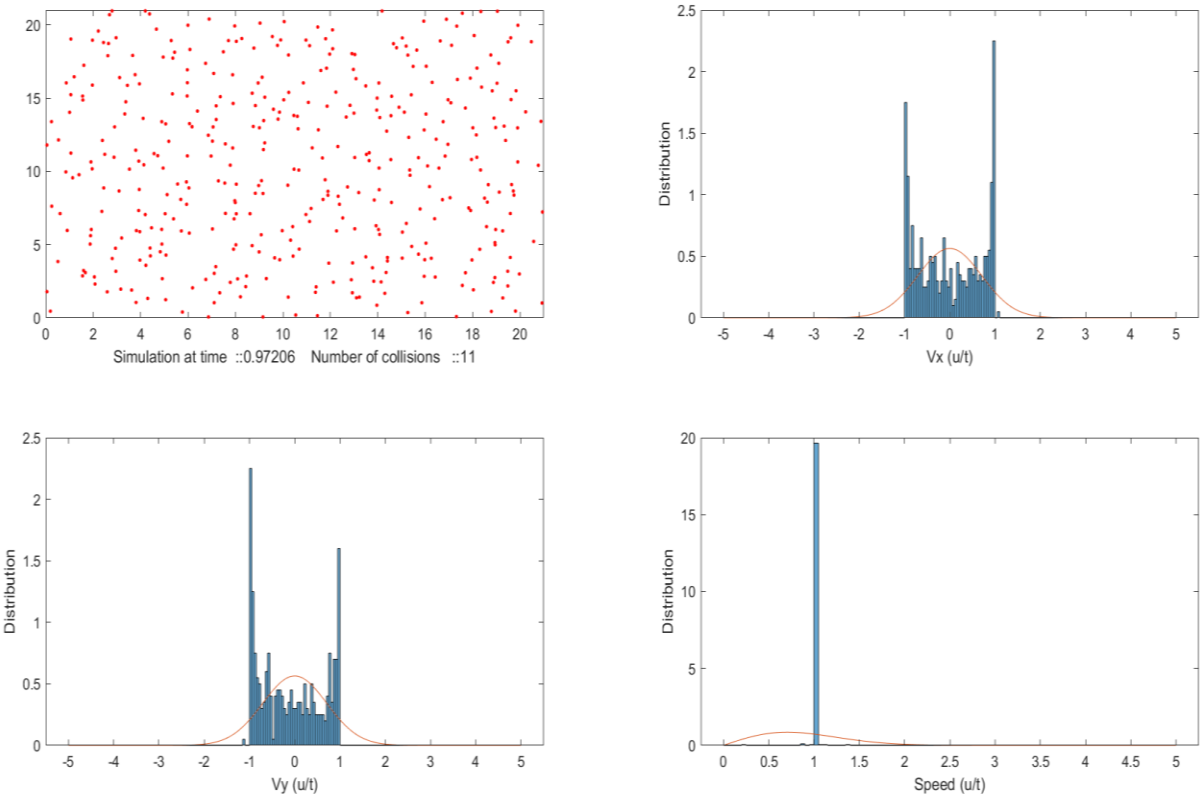



**Figure.7 Condition at time 0.97206s**

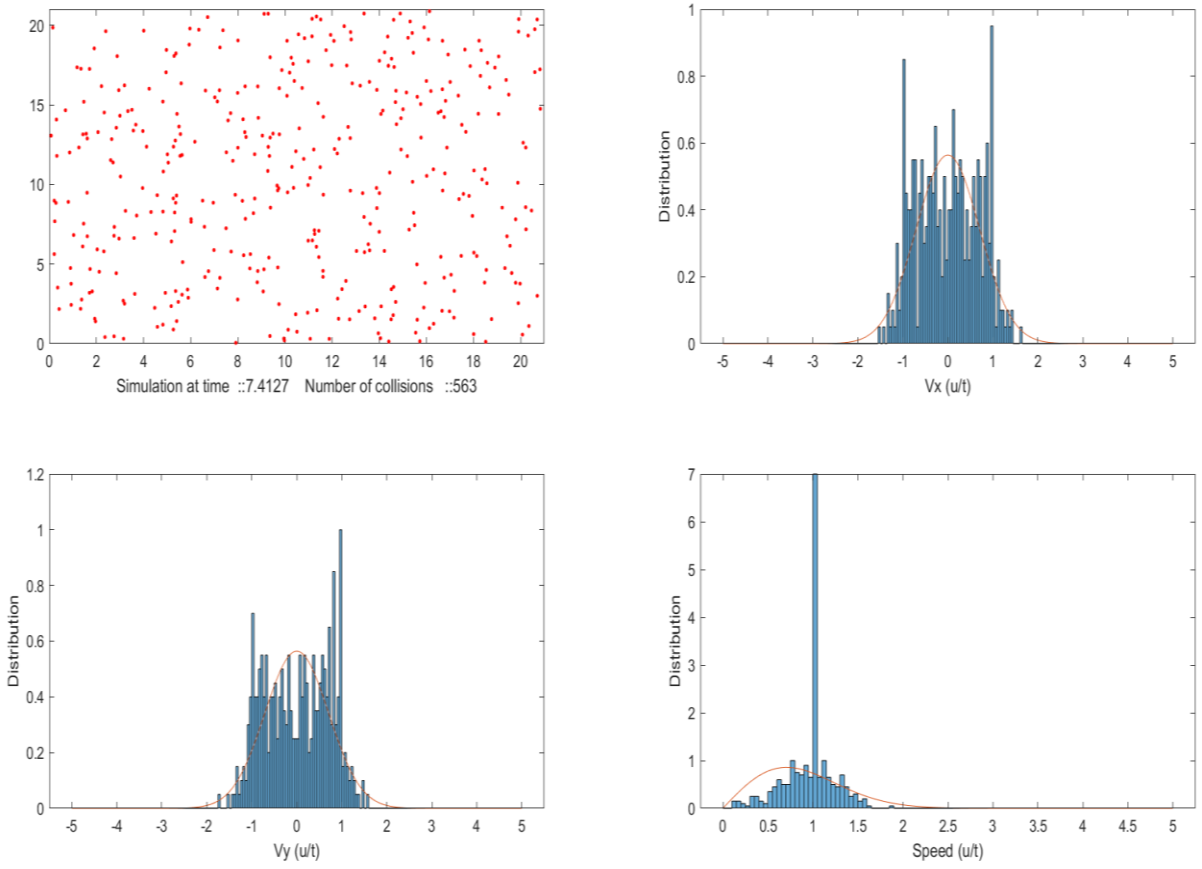



**Figure.8 Condition at 7.4127s**

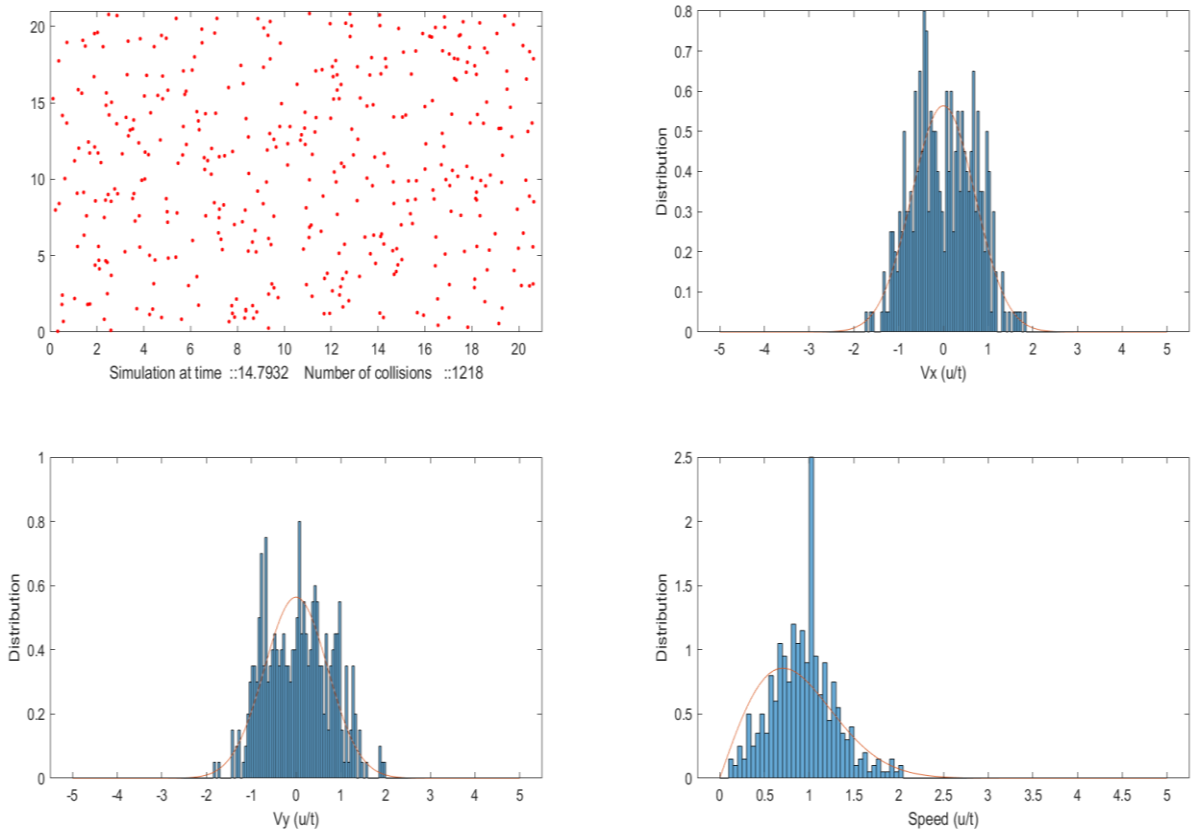



**Figure.9 Condition at 14.7932s**

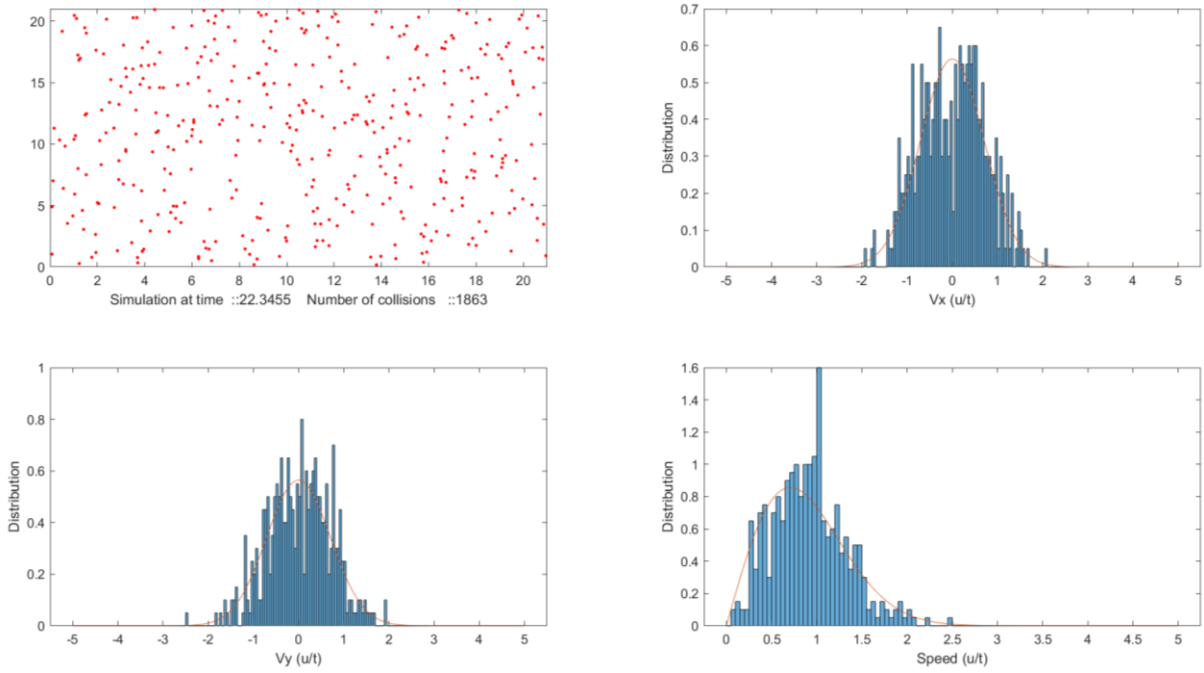



**Figure.10 Condition at 22.3455s**

# MATLAB Source Code

**https://github.com/Zongxuan-Lin/MECH-578-2D-Gas-Elastic-Collision-Simulation/blob/main/ZLCOde.m**