Material Modeling Homework 3 Report

Gengyao Qiu

October 20, 2025

1 Introduction

In this assignment, we mainly used Python programming to implement algorithms such as differential equations and random walks, which are widely applied in computational materials science.

2 Results and Discussions

2.1 Oscillatory Motion and Chaos

2.1.1 Part 1

Giving that the linear, damped, pendulum defined as:

$$\frac{\mathrm{d}^2\theta}{\mathrm{d}t^2} = -f\frac{g}{l}\theta - 2\gamma\frac{\mathrm{d}\theta}{\mathrm{d}t} + \alpha_D \sin(\Omega_D t) \tag{1}$$

We know the resonance will occurs when:

$$\Omega_D = \sqrt{\frac{g}{l} - 2\gamma^2} \approx 0.935 \tag{2}$$

In this condition, the amplitude of pendulum is around 0.4 rad, which can't been treated with small-angle approximation.

2.1.2 Part 2

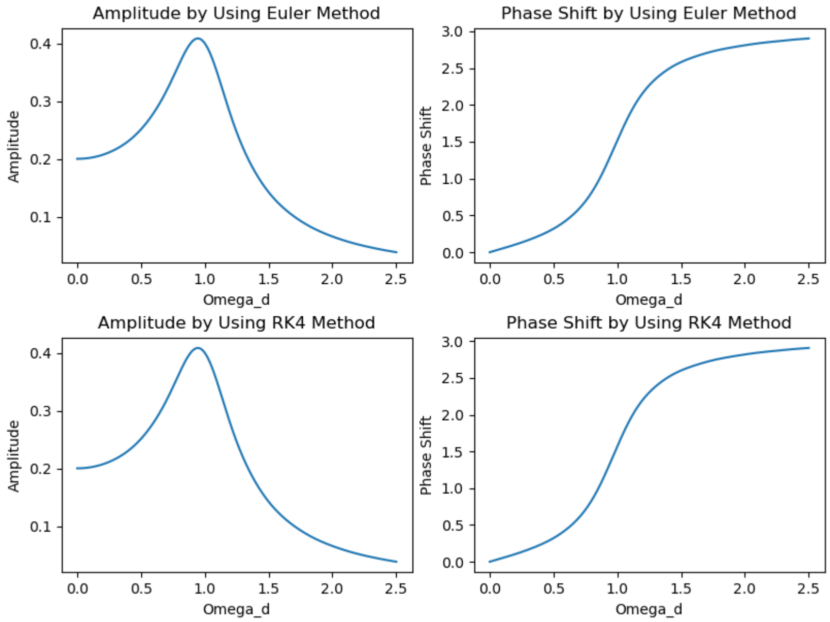


Figure 1: The amplitude and phase shift of pendulum by using Euler-Cromer and Runge-Kutta 4th order methods.

We can calculate the full-width as half maximum (FWHM) numerically, which are around 1.17 and 1.13 for Euler-Cromer and RK4 respectively, this is around 4 times of gamma. In figure 1, we know that the amplitude is around 0.2 and phase shift is around 0 while the Ω_D is around 0, and amplitude will come to 0 when Ω_D very high, at this time, the phase shift will approach to π, this means the motion is in opposite phase to the driving force.

2.1.3 Part 3

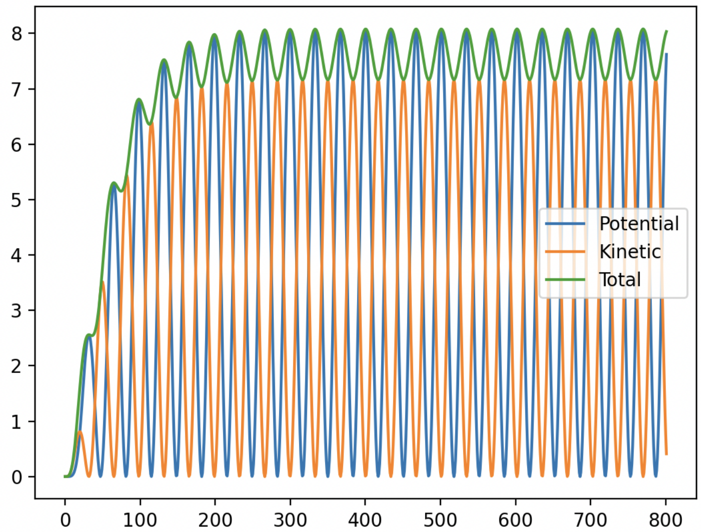


Figure 2: The potential, kinetic and total energies at resonance frequency.

Based on the figure 2, we know that the total energy for this system are not a constant, this may caused by the work done by the driving force.

2.1.4 Part 4

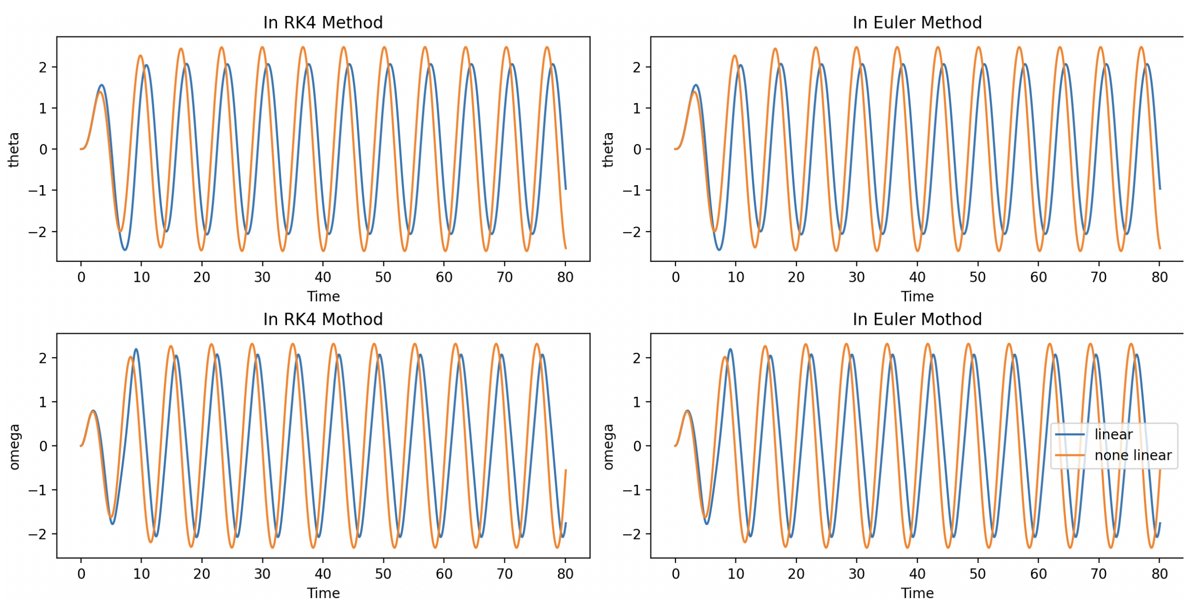


Figure 3: The theta and omega in linear and nonlinear condition.

2.1.5 Part 5

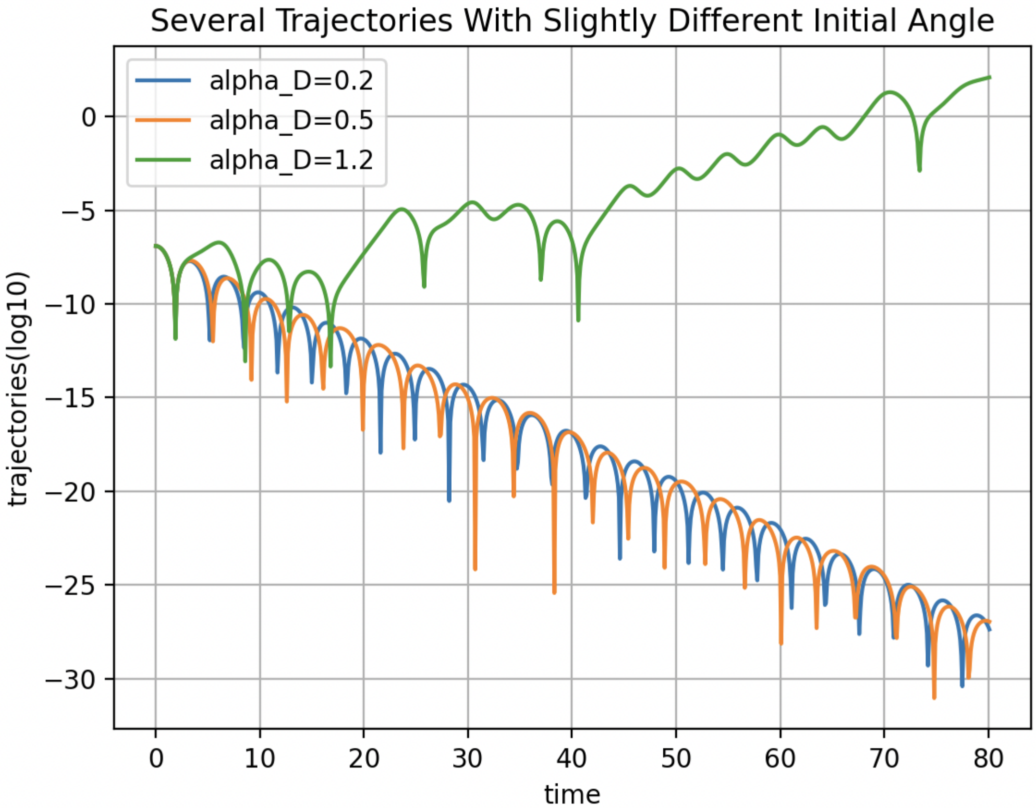


Figure 4: The theta and omega in linear and nonlinear condition.

The Lyapunov exponent λ are the slope of curve in figure 4, using linear regression method we know they are -0.25, -0.25, 0.13 respectively. This shows that system comes to chaos while $\alpha_D = 1.2 \text{rad/s}^2$

2.2 Poisson Equation for Dipole

2.2.1 Part 1

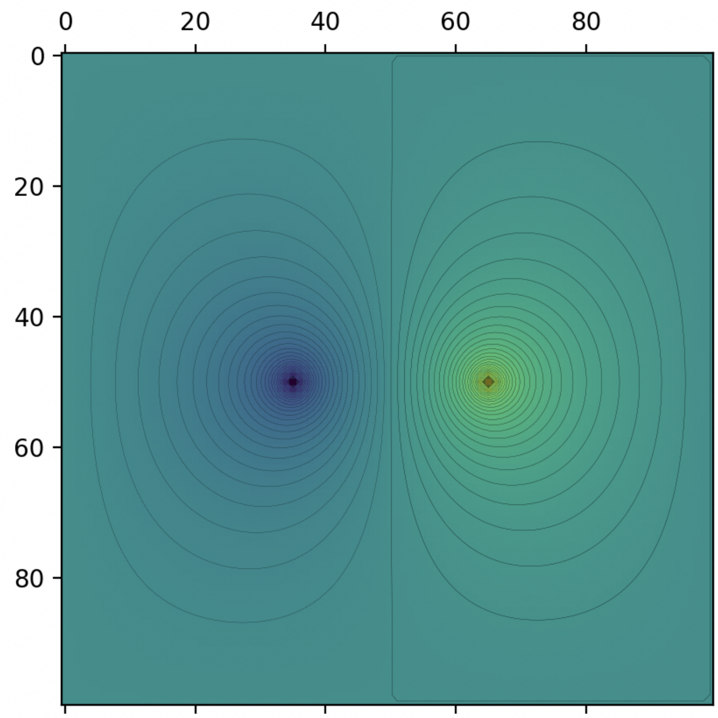


Figure 5: The potential heat map for a dipole.

2.2.2 Part 2

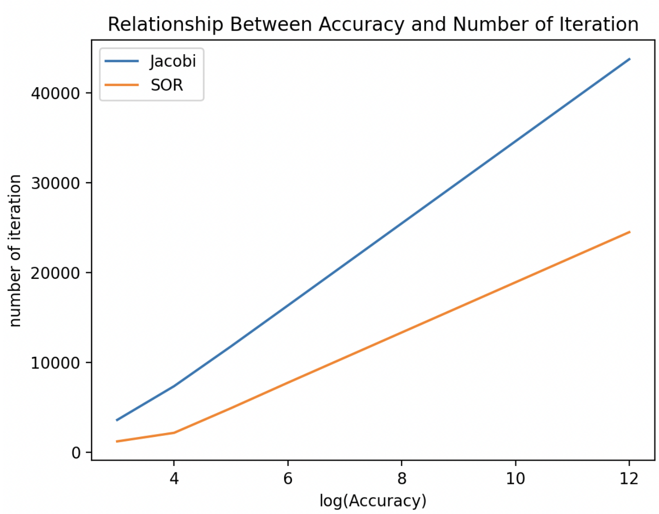


Figure 6: Relationship between accuracy(error) and number of iteration.

The figure 6 shows that the $N_{iter} \propto \log \epsilon$. And the SOR method is quicker than Jacobi.

2.2.3 Part 3

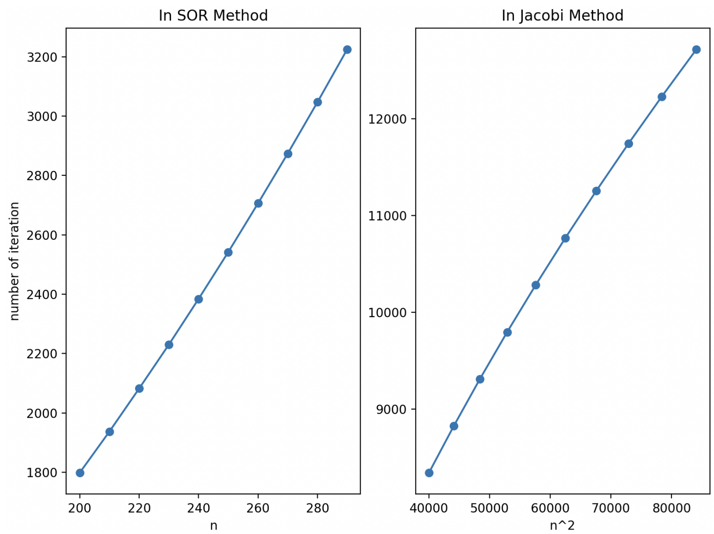
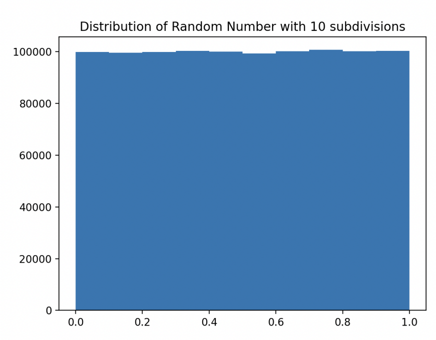


Figure 7: Relationship between number of iteration and number of grid points.

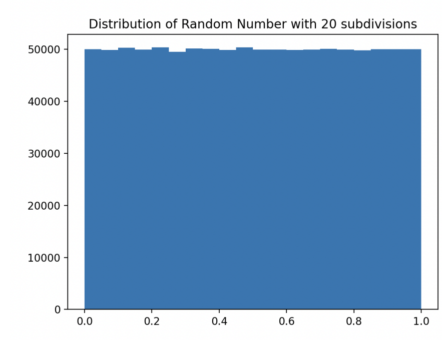
In figure 7, we found $N_{\text{iter}} \propto n^2$ and $N_{\text{iter}} \propto n$, and this is consistent with what we expected.

2.3 Random Numbers

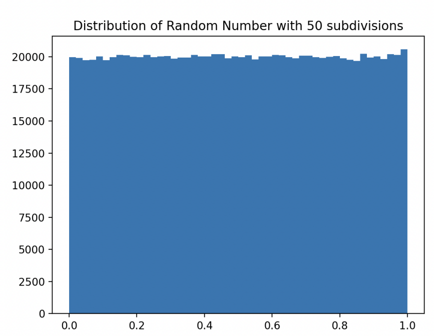
2.3.1 Part 1



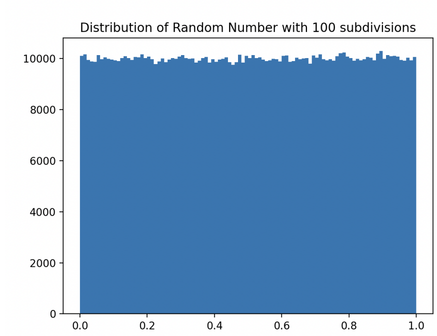
(a) 10 subdivisions



(b) 20 subdivisions



(c) 50 subdivisions



(d) 100 subdivisions

Figure 8: Distributions for random numbers in different subdivisions.

2.3.2 Part 2

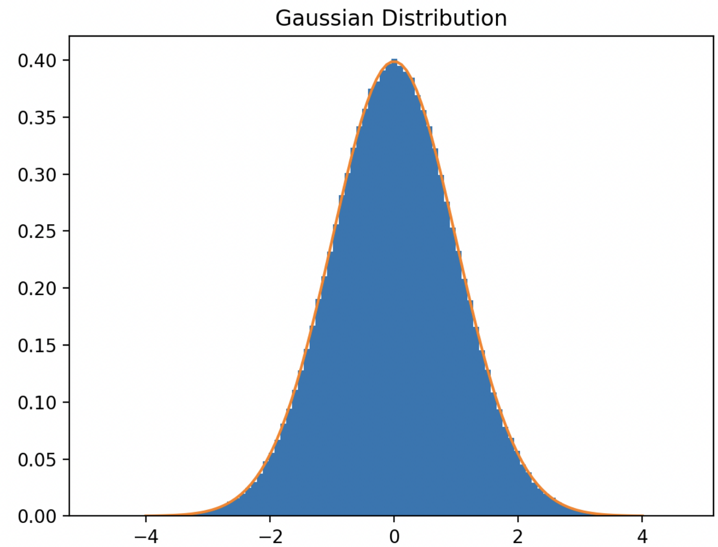


Figure 9: Gaussian distribution

2.4 2D Random Walk

2.4.1 Part 1

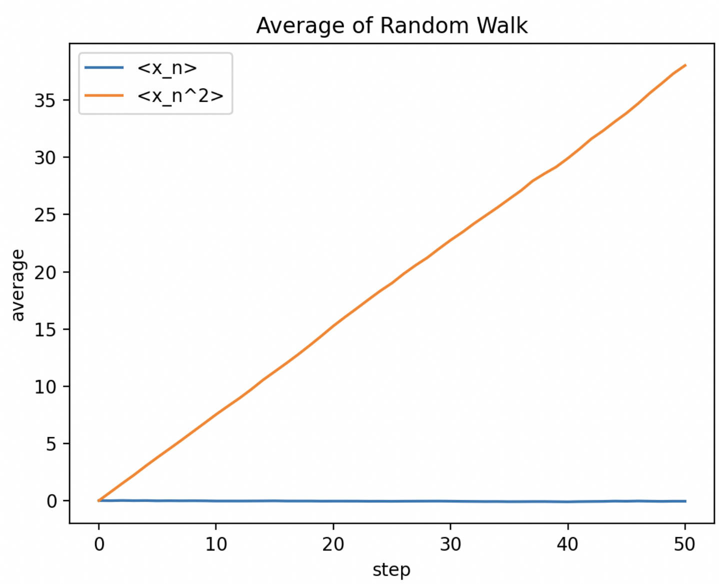


Figure 10: Average numbers of random walk.

Based on figure 10, the $< x_n > \propto step$ and $< x_n^2 >= 0$

2.4.2 Part 2

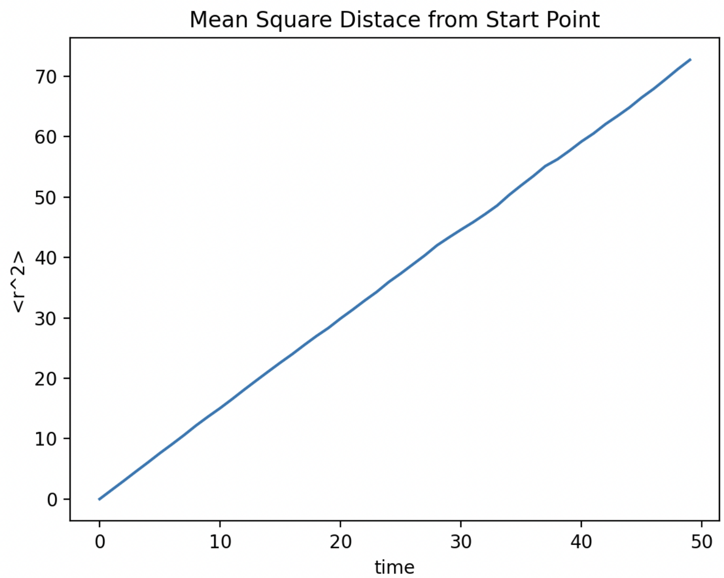


Figure 11: Mean square distance from start point

2.5 Diffusion Equation

2.5.1 Part 1

Spacial expectation value:

$$< x^2 >= \int_{-\infty}^{\infty} x^2 \rho(x,t)\mathrm{d}x = \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{\infty} x^2 \exp(-\frac{x^2}{2\sigma^2})\mathrm{d}x \tag{3}$$

Knowing that:

$$\int_{-\infty}^{\infty} x^2 \exp(-\frac{x^2}{2\sigma^2})\mathrm{d}x = \frac{1}{2}\sqrt{\pi}(\frac{1}{2\sigma^2})^{-3/2} = \frac{1}{2}\sqrt{\pi}(2\sigma^2)^{3/2} \tag{4}$$

where $\alpha = 1/(2\sigma^2)$, so:

$$< x^2 >= \frac{1}{\sqrt{2\pi}\sigma} \cdot \frac{1}{2}\sqrt{\pi}(2\sigma^2)^{3/2} = \sigma^2 \tag{5}$$

2.5.2 Part 2

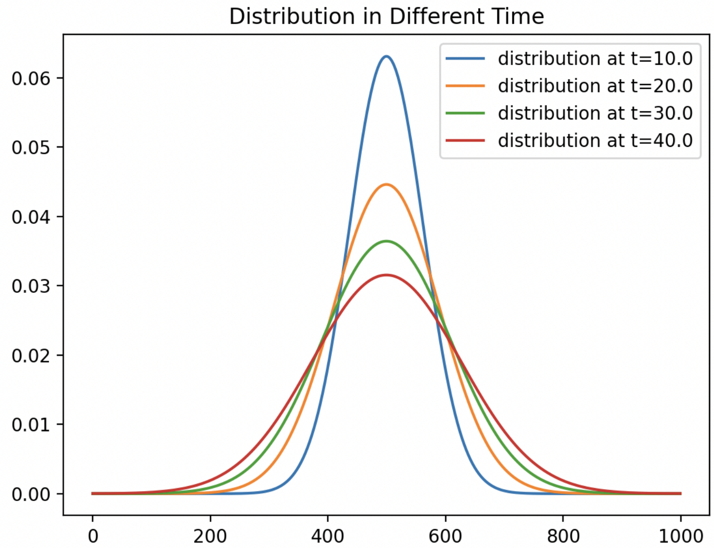


Figure 12: Distribution in different time

The program's out put is `sqrt(2Dt)=[ 6.32455532 8.94427191 10.95445115 12.64911064], sigma=[ 6.32530632 8.94480296 10.95488476 12.64948616]`. This shows that $\sqrt{2Dt} = \sigma(t)$

2.6 Mixing of two Gases

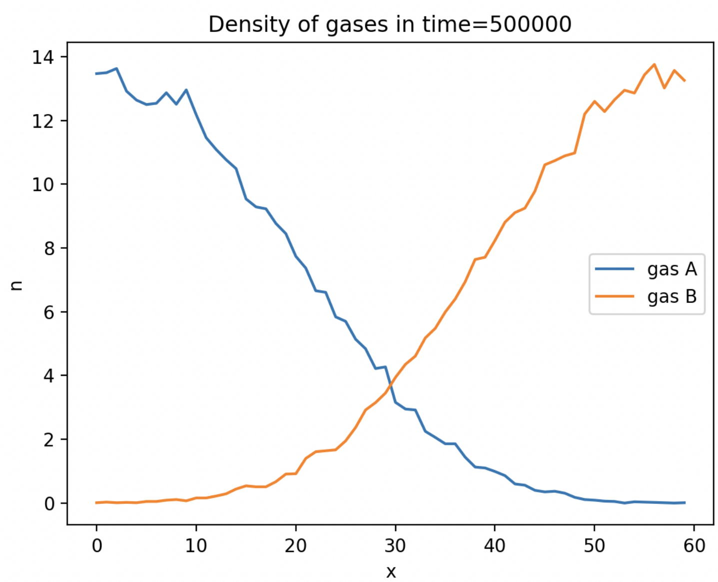
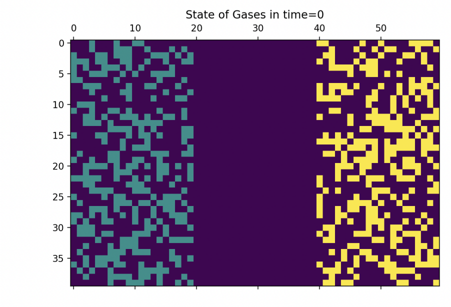
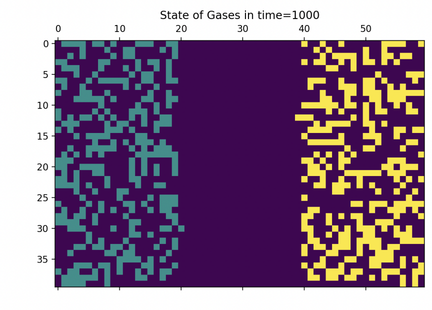


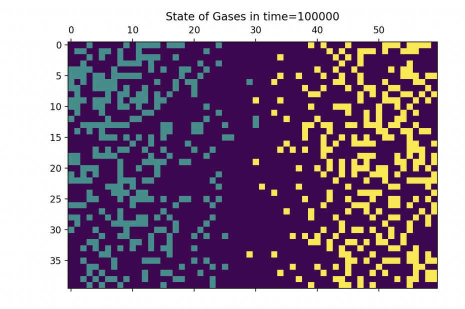
Figure 13: The density of gases, the plot have averaged over 100 trials.



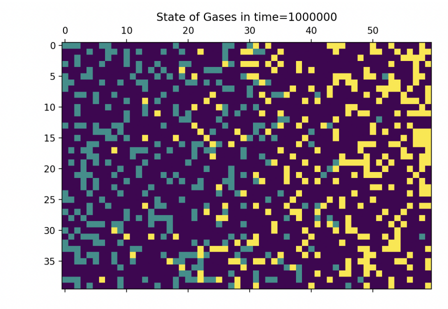
(a) time=0



(b) time=1000



(c) time=100000



(d) time=1000000

Figure 14: Sample configurations of the grid at different time.

3 Contributions

In this project, I engaged in group discussion and help my group members to debug the Python code in all sections. During the project, I discussed the possibilities that may cause the difference between our output, and also discussed how to analysis the deviation. But we finished our homework include coding and formula derivation independently.