Tagged Particle Diffusion in Quasi One-Dimensional Gas with Hamiltonian Dynamics

A PROJECT REPORT

Submitted as part of BBD451 B. Tech Major Project (BB1)

Submitted by:

Arya Ladhani

2020BB10026

Guided by:

Professor Anjan Roy

DEPARTMENT OF BIOCHEMICAL ENGINEERING AND BIOTECHNOLOGY INDIAN INSTITUTE OF TECHNOLOGY DELHI

November 2023

DECLARATION

I certify that

a) the work contained in this report is original and has been done by me under the guidance of my supervisor(s).
b) I have followed the guidelines provided by the Department in preparing the report.
c) I have conformed to the norms and guidelines given in the Honour Code of Conduct of the Institute.
d) whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the report and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.

Signed by:

Arya Ladhani

CERTIFICATE

It is certified that the work contained in this report titled “Tagged Particle Diffusion in Quasi One-Dimensional Gas with Hamiltonian Dynamics” is the original work done by and has been carried out under my supervision.

Signed by:

Professor Anjan Roy

Table of Contents

1. List of figures

2. List of equations

3. List of Tables

4. Abstract

In this study, we examine the mean square displacement (MSD) of a tagged particle in a one-dimensional gas, implementing Hamiltonian dynamics. The system, confined in a finite box, achieves thermal equilibrium, allowing us to explore the dynamics of particles in a purely deterministic environment. Our simulations reveal that the MSD of the tagged particle exhibits a linear growth with time, consistent with theoretical expectations for systems with uniform particle masses. At extended durations, the MSD approaches a saturation point, reflecting the finite-size effects inherent to the system. This behaviour, along with the observed damped oscillatory traits at longer timescales, highlights the complex nature of diffusion processes in one-dimensional systems. Our findings provide robust support to the theoretical framework of tagged particle diffusion and enhance the understanding of Hamiltonian dynamics in governing transport phenomena at microscopic scales.

Keywords: Tagged Particle Diffusion, Hamiltonian Dynamics, One-Dimensional Gas, Mean Square Displacement, Thermal Equilibrium, Finite-Size Effects, Deterministic Particle Systems, Long-Time Dynamics

5. Introduction

5.1 Diffusion

Diffusion is a spontaneous transport phenomenon whereby particles, such as atoms, molecules, or ions, migrate from regions of higher concentration to regions of lower concentration. This movement is a result of the random thermal agitation intrinsic to the particles at temperatures above absolute zero. It is a vital process, underpinning a myriad of systems and applications across different phases of matter, be it gases, liquids, or solids.

In biological contexts, diffusion is indispensable for the delivery of nutrients and oxygen to cells, as well as the removal of waste products. This process is governed by the permeability of cell membranes and the concentration gradients across them. Similarly, in ecosystems, diffusion is responsible for the natural balancing of substances within habitats.

In environmental science, diffusion is a key factor in the dispersion of pollutants, contributing to both the dilution and the spread of contaminants in air and water bodies. It is an essential consideration in the design of strategies for pollution control and remediation.

Chemical engineering and industrial processes extensively rely on diffusion. It is instrumental in the mixing of reactants to ensure uniform chemical reactions. In the field of materials science, diffusion is the driving force behind the homogenization of alloys and the doping process in semiconductor manufacturing, where controlled diffusion of impurities into materials tailors their electrical properties.

Moreover, diffusion is not limited to passive transport; it can also be influenced by external fields or gradients in pressure and temperature, leading to phenomena such as osmosis and thermophoresis.

5.2 Single-file diffusion

[6] Single-file diffusion represents a specialized and intriguing category of diffusion that occurs within channels so narrow that particles can only move in a linear sequence, akin to cars in a traffic jam. This constrained movement significantly diverges from the classical notion of Brownian motion, where particles have the liberty to move in three dimensions and can manoeuvre around each other.

The unique characteristics of single-file diffusion have profound implications in various fields. For example, in biological systems, it is vital for understanding the transport of ions and molecules through narrow channels in cell membranes, which is essential for cell function and signalling. In materials science, the principles of single-file diffusion are important for the design of molecular sieves or filtration systems that can precisely separate particles based on size.

The study of single-file diffusion can be approached using computational simulations that model mechanical collision theory, allowing researchers to observe the interactions between particles within these confined spaces. These models can shed light on the statistical behaviours of particles and help in predicting the transport properties of the system under study.

5.3 Collision Theory

[5]Collision theory plays a pivotal role in understanding the dynamical outcomes of interacting bodies, particularly within the realm of classical mechanics. At the heart of this theory lies the law of conservation of momentum, which serves as a cornerstone principle asserting that in the absence of external forces, the total momentum of a system remains constant. This invariance holds true irrespective of the complexity of interactions taking place within the system.

Applying this principle, one can dissect the interactions in a basic two-body system to gain insights into the nature of the collision process. When two bodies collide in an isolated system where no external forces act, their combined momentum before and after the collision remains unchanged. This scenario provides a powerful tool for analysis, allowing predictions about post-collision velocities and trajectories. The conservation laws apply to both perfectly elastic collisions, where kinetic energy is also conserved, and inelastic collisions, where kinetic energy is not conserved and may be converted into other forms of energy such as heat or sound. The law of conservation of momentum can be written as:

$$m_1u_1 + m_2u_2 = m_1v_1 + m_2v_2$$

Where, m_i is the mass of the i^{th} component in the system, u_i is the velocity of the i^{th} component before the collision and v_i is the velocity of the i^{th} component after the collision.

The collision theory is incomplete without the equation for co-efficient of restitution, which is an indicator of the total energy of the system and tells us about the elasticity of the collision (perfectly elastic/ elastic/ perfectly inelastic). It is given by the following formula:

$$e = \frac{v_2 - v_1}{u_1 - u_2}$$

The coefficient of restitution (e) is a critical parameter in collision dynamics, quantifying how much kinetic energy is conserved in the aftermath of a collision. It is defined as the ratio of relative speed after the collision to the relative speed before the collision. This coefficient is bounded between 0 and 1, where 0 indicates a perfectly inelastic collision and 1 signifies a perfectly elastic collision.

In scenarios involving two particles with equal mass undergoing a collision, the coefficient of restitution plays a pivotal role in determining the outcome of their velocities. A negative coefficient would imply that the particles gain energy in the collision, which violates the principle of conservation of energy. Therefore, the coefficient of restitution must always be non-negative.

During a perfectly elastic collision where e=1, two particles of equal mass essentially exchange velocities. To elucidate, if two objects — Object A and Object B — collide, and if Object A is moving while Object B is stationary, after a perfectly elastic collision, Object A will come to a complete stop and Object B will move away with the velocity that Object A had before the collision.

For perfectly elastic collisions, the conservation of kinetic energy is also applied, which in turn results in the exchange of velocities for equal mass particles.

When considering the collision of a particle with a wall, the situation simplifies further. If a particle collides elastically with a wall, it will rebound with the same speed but in the opposite direction. This occurs because the wall is typically considered to have an infinite mass compared to the particle, and thus, its velocity remains unchanged (essentially zero). The particle's velocity change can be

understood intuitively in the context of an elastic collision without delving into the complexities of momentum conservation equations.

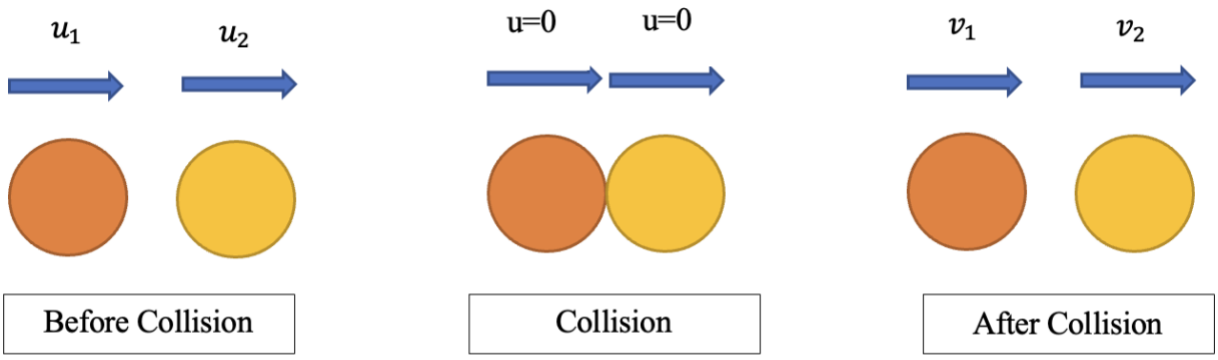


Figure 1 How collision occurs in a 2-body system

5.4 Bulk diffusion

Bulk diffusion is a key transport mechanism that characterizes the stochastic motion of particles through a substance from areas of higher concentration to regions of lower concentration. This process is integral to a multitude of natural phenomena and industrial applications. It encompasses the dispersal of gases within the Earth's atmosphere, which is crucial for sustaining life, to the intermixing of substances that drive chemical reactions, a foundation of chemical engineering.

The mathematical description of bulk diffusion is encapsulated by Fick's laws of diffusion. These laws quantitatively relate the diffusion flux to the concentration gradient, providing a predictive framework for the mass transfer rate. The first law posits a direct proportionality between the diffusion flux and the gradient, while the second law offers a description of how diffusion causes the concentration field to change over time.

5.5 Fick's Laws of Diffusion

1. **[4]**First Law of Diffusion: Fick's first law states that the flux of a substance is directly proportional to the concentration gradient and is inversely proportional to the diffusion coefficient. In simpler terms, it tells us that substances will naturally flow from areas of higher concentration to areas of lower concentration. Typically, the law is stated as:

$$J = -D \cdot \frac{\partial C}{\partial x}$$

Where:

J represents the flux of the substance, which is the amount of substance crossing a unit area per unit time.

D is the diffusion coefficient or diffusivity of the substance, which quantifies how easily the substance can move through the medium.

$\frac{\partial C}{\partial x}$ is the concentration gradient, indicating how the concentration of the substance changes with respect to the distance along a specified direction (typically x-axis).

2. **[4]**<u>Second Law of Diffusion:</u> Fick's second law provides a more detailed description of how the concentration of a diffusing substance changes with time.

$$\partial C/_{\partial t} = D.\nabla^2 \mathrm{C}$$

Where:

$\partial C/_{\partial t}$ represents the change in concentration of diffusing substance with time.

D represents the diffusion coefficient.

$\nabla^2 \mathrm{C}$ is the Laplacian of the concentration, representing the second spatial derivative of concentration.

6. Literature Review

6.1 Event- Driven Algorithm

In the realm of Hamiltonian dynamics, systems evolve deterministically, and any apparent randomness arises from uncertainties in initial conditions. One seminal result in this domain is Jepsen's investigation [1] of tagged particle diffusion within a one-dimensional gas of hard, elastically colliding particles with equal masses. For an infinite system at a fixed particle density, Jepsen demonstrated that the mean square deviation (MSD) of a tagged particle from its initial position grows linearly with time (t).

In this experiment, we aim to replicate and extend the findings of Jepsen [1]. We set up the following initial conditions:

1. **N-Particle Ensemble:** We consider an ensemble of N particles confined within a container of length L.
2. **Equal Mass:** All particles have equal mass, ensuring that collision dynamics are solely determined by particle velocities and interactions.
3. **Gaussian Distribution:** Initial particle positions are distributed according to a Gaussian distribution within the container.
4. **Boltzmann Distribution:** Particle velocities follow a Boltzmann distribution, representing the thermal motion of the system.
5. **Elastic Collisions:** All particles undergo perfectly elastic collisions, conserving both kinetic energy and momentum.
6. **Non-Crossing Particles:** We ensure that no two particles can cross each other during their motion.
7. **Point Particles:** Particles are treated as point masses with zero radius, simplifying collision modelling.

Event-Driven Algorithms and Molecular Diffusion: Event-driven algorithms are computational tools commonly employed in molecular dynamics simulations, particularly for studying molecular diffusion and other dynamic processes at atomic and molecular scales. These algorithms efficiently update the positions and velocities of individual particles based on discrete events, such as collisions or reactions. They are well-suited for modelling molecular diffusion as they accurately track particle motion while preserving energy and momentum.

Algorithm Workflow for Molecular Diffusion:

1. **Initialization:** The simulation commences with an initial configuration of particles, each possessing defined positions and velocities. Often, the simulation box is made periodic to mimic an effectively infinite system.

2. **Event Detection:** The algorithm continuously monitors the system to identify potential events, primarily involving particle collisions or interactions in the context of molecular diffusion.
3. **Event Selection:** The algorithm selects the next event to process based on the anticipated occurrence time. This typically involves identifying the event with the smallest time increment among all potential events.
4. **Event Execution:** The selected event is executed, leading to updates in the positions and velocities of the particles involved. For molecular diffusion, this often corresponds to particles undergoing random displacements due to thermal motion.
5. **Time Advancement:** After event execution, the simulation time advances to match the event's time.
6. **Energy and Momentum Conservation:** Event-driven algorithms ensure that energy and momentum are conserved by appropriately adjusting particle velocities after each event.
7. **Repeat:** The cycle of event detection, selection, execution, and time advancement repeats until the simulation reaches a specified endpoint, such as a predefined simulation time or the achievement of specific simulation objectives.

Exploring Single-File Diffusion in the Context of Heat Conduction: Previous work has extensively studied single-file diffusion concerning heat conduction [7-10]. However, this perspective focuses on the propagation of conserved quantities as functions of position and time without considering the identity of each particle. This fundamentally changes the dynamics of the system. For instance, in an equal mass hard particle gas, conserved quantities propagate ballistically, resulting in thermal conductivity proportional to N. Conversely, tagged particle dynamics within the same system exhibit diffusive behaviour. Therefore, investigating diffusion from the perspective of tagged particles requires a distinct approach.

6.2 Gaussian Distribution

In our simulation experiment, we utilized the Gaussian distribution to randomly distribute N particles within a container of length 'L.' This choice was made to mimic the natural behaviour of particles, especially under the influence of thermal motion. To achieve this, we employed the **np.random.randn()** function from the NumPy library.

NumPy's **np.random.randn()** function is a valuable tool for generating random values following a standard Gaussian distribution (mean = 0, standard deviation = 1). We then scaled these random values by 'L,' adapting them to our specific simulation setup.

By doing so, we effectively placed particles at random positions within the container while ensuring that their distribution adhered to a Gaussian pattern. These positions served as the initial conditions for our simulation.

This approach allowed us to precisely model the initial state of our system, capturing the effects of thermal motion and enabling a detailed exploration of particle dynamics in the context of single-file diffusion.

6.3Velocity generation using Boltzmann distribution

[13]The Maxwell–Boltzmann distribution is a fundamental concept in statistical physics that provides insights into the distribution of speeds of particles within a gas. This distribution is represented as a bell-shaped curve, with its peak indicating the most probable speed, while the probability of finding a molecule with a higher or lower speed decreases symmetrically on either side of the peak.

Temperature Dependency: One of the remarkable features of the Maxwell–Boltzmann distribution is its strong dependence on the temperature of the gas. This temperature-speed relationship is critical for understanding how gases behave under different thermal conditions:

- **High Temperatures:** At elevated temperatures, gas molecules possess greater kinetic energy. Consequently, they move at higher speeds. This phenomenon leads to a broader distribution curve, as there is a higher probability of finding molecules with faster speeds. The entire curve shifts to the right, reflecting the increased thermal energy of the gas.
- **Low Temperatures:** Conversely, lower temperatures result in gas molecules having less energy and moving more slowly. In this scenario, the Maxwell–Boltzmann distribution curve becomes narrower and shifts to the left. This shift signifies that there is a higher likelihood of observing molecules with slower speeds in a cooler gas.

Statistical Physics and the Boltzmann Velocity Distribution: To obtain meaningful thermodynamic averages and observables in molecular simulations, it is essential to ensure that the simulations adhere to the principles of statistical physics. This is where the Boltzmann velocity distribution comes into play:

- **Specialized Distribution:** The Boltzmann velocity distribution is a specialized form of the Maxwell–Boltzmann distribution tailored for describing the distribution of particle velocities at a given temperature. It characterizes the probability of finding particles with specific velocities in a gas or liquid system.
- **Thermal Realism:** By employing the Boltzmann velocity distribution, simulations achieve a high degree of thermal realism. This means that the simulated particles' velocities closely mimic the thermal behaviour of real-world particles in a system at a specific temperature.
- **Accurate Representation:** As a result, simulations can accurately model how particles interact, diffuse, and exchange energy, all in accordance with the fundamental laws of thermodynamics. The Boltzmann velocity distribution ensures that the simulated particles' velocities are statistically consistent with the given temperature, enabling the calculation of thermodynamic properties, such as pressure, temperature, and energy, that are in line with real-world observations.

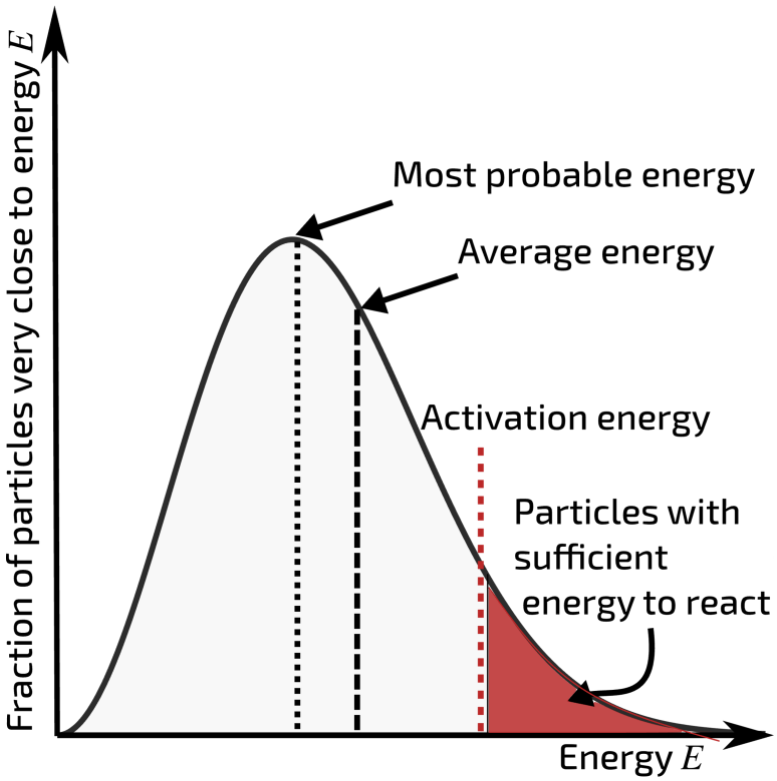


Figure 2 Maxwell- Boltzmann velocity distribution curve

We acquire meaningful thermodynamic averages and observables by ensuring that our simulations follow the rules of statistical physics, which is made possible by the Boltzmann velocity distribution.

For a system containing a large number of identical non-interacting, non-relativistic classical particles in thermodynamic equilibrium, the fraction of the particles within an infinitesimal element of the three-dimensional velocity space d^3v, centred on a velocity vector of magnitude v, is given by

$$f(v)\, d^3v = \left[\frac{m}{2\pi kT}\right]^{\frac{3}{2}} \exp\left(-\frac{mv^2}{2kT}\right) d^3v,$$

where:

- m is the particle mass
- k is the Boltzmann Constant
- T is the thermodynamic temperature
- $f(v)$ is a probability distribution function, properly normalized so that integration of left hand side over all velocities is unity

7. Materials and Methods

Full codes of the event-driven algorithm and simulation algorithm are available here.

7.1 Using the event-driven algorithm

According to assumption [**Error! Reference source not found.**], no two particles cross each other, therefore, the relative positions of the particles will remain the same throughout the simulation. This also means that are only three possible collisions:

1. Collision of the first particle with the wall
2. Collision of the last particle with the wall
3. Collision of any two particles which have a positive relative velocity of approaching each other

To use the event-driven algorithm, we first calculate which is the next possible collisions. This we do by calculating the time duration of all possible collisions (between adjacent molecules) and find out the smallest time of collision. This is the collision which occurs the fastest in that timeframe. So using the algorithm we execute the collision. In this step, we follow these steps:

1. Reverse the velocities of the colliding particles (according to collision theory)
2. Update the positions of all particles (by adding the distance a particle travelled in that timespan)
3. Reducing the remaining time by the time of next collision

These steps are repeated until the remaining time in simulation becomes 0.

Here is a brief explanation of the functions used:

1. **next_collision Function:**
    - This function calculates and schedules the next collision events between particles and with the container walls.
    - It initializes an event queue (**event_queue**) to manage upcoming collisions.
    - It loops through all pairs of particles and calculates the time (**dt**) until they will collide if they have different velocities. If **dt** is positive, it schedules this collision event in the event queue.
    - Wall collisions are also considered, and events for them are scheduled based on the time it takes for a particle to hit a wall.
    - The events are stored in the event queue, sorted by their collision times.
    - The function returns the event queue.
2. **update_positions Function:**
    - This function updates the positions of particles based on their current positions and velocities and a given time increment (**dt**).

- It uses simple kinematics to update the positions by adding the product of velocities and time increment to the current positions.

3. **perform_realization Function:**
    - This is the main function that performs a single realization of the simulation.
    - It initializes the positions and velocities of particles, records the initial position of a specific particle (**particle_index**), and sets up the simulation time and event queue.
    - The function iterates for a specified number of time steps, advancing the simulation by processing events.
    - Within each step, it checks for the next event in the event queue and processes events until the end of the current time step.
    - Depending on the type of collision (particle-particle or particle-wall), it updates positions and velocities accordingly.
    - The function calculates the mean squared displacement (MSD) of the selected particle at each step and stores the results in an array (**msd**).
    - Finally, it returns the MSD data, which measures how the position of the selected particle changes over time during the realization.

To understand how the event-driven algorithm is used, let us consider the following example of 5 particles.

Positions (m)	**Velocities (ms^{-1})**
0.300	-110.50
0.450	200.15
0.570	300.40
0.600	-400.50
0.800	175.60

In this example, the first collision occurs between particle 3 and 4 at 4.28×10^{-5} s. So after the first iteration, the time of simulation is reduced by 4.28×10^{-5} s, particle 3 has velocity -400.50 ms^{-1} and particle 4 has velocity 300.40 ms^{-1}. The position of each particle has been updated and the same algorithm is repeated.

While the ensemble mentioned in the above examination may seem very simple, when the system is scaled-up to 100 particles and simulation time is increased to 1000 s, the number of collisions happening increase manifold. The time gap between each collision also decreases due to the close spacing of particles.

7.2 Main Algorithm

To analyse the relation of mean squared displacement with time across different realizations, we run the main algorithm:

for every realization:

1. Initiate parameters (temperature, simulation time)
2. Generate new ensemble with different velocities and positions of particles
3. Initiate data structure to store MSD of tagged particle (after uniform time steps)
4. Run simulation for every timestep:
    a. Check if any two particles exist such that the difference in their positions is either 0 or less than 0.
    b. Update the position of each particle by adding timestep*v_i to their current position
    c. Record Mean Squared Displacement of tagged particle $(x(t) - x(0))^2$
    d. Check if current time – start time < simulation time; If yes, stop simulation else continue
5. Record positions of the particle after every timestep

Then the mean-squared-displacement of tagged particle is then averaged over several iterations for each timestep and the trend is then analysed.

8. Experiments and Observations

In our research, we conduct two pivotal experiments to ensure the reliability of our simulation framework.

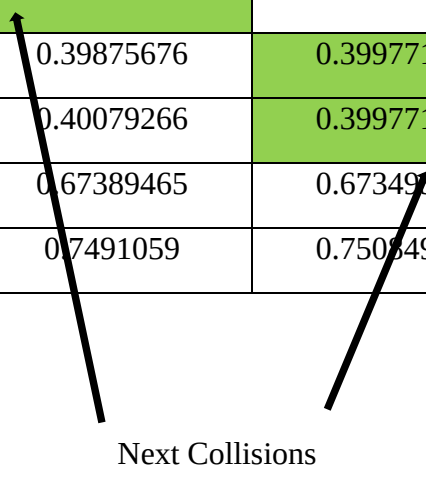
Experiment 1: Microscale Collision Analysis

In this initial experiment, we meticulously investigate microscale particle collisions. Our focus is on two critical aspects:

1. **Collision Precision:** We scrutinize particle trajectories to guarantee that no particle overlaps, and our collision detection is highly accurate. This validation reaffirms the fidelity of our simulation.
2. **Algorithm Verification:** We employ an event-driven algorithm, initializing it with key parameters:
    - Container Length: 1 m
    - Simulation Time: 0.0005 s
    - Temperature: 400 K
    - Number of Particles: 10

All particles, assigned equal masses, undergo Gaussian Distribution for positions and Boltzmann Velocity Distribution for velocities. Our simulations then commence, following the event-driven algorithm to meticulously study the latest collision event. This experiment ensures the integrity and precision of our simulation's collision handling.

Time	0	2.3904E-05	2.6482E-05
1th_particle_position	0.04642303	0.037454	0.03648673
2th_particle_position	0.08421002	0.07030359	0.06880385
3th_particle_position	0.10583463	0.10060589	0.100042
4th_particle_position	0.1223067	0.11098898	0.10976842
5th_particle_position	0.25055337	0.26076317	0.25990664
6th_particle_position	0.26870535	0.26076317	0.26186424
7th_particle_position	0.38934349	0.39875676	0.39977194
8th_particle_position	0.41025737	0.40079266	0.39977194
9th_particle_position	0.67759806	0.67389465	0.67349526
10th_particle_position	0.73293697	0.7491059	0.75084964



Collision #1:

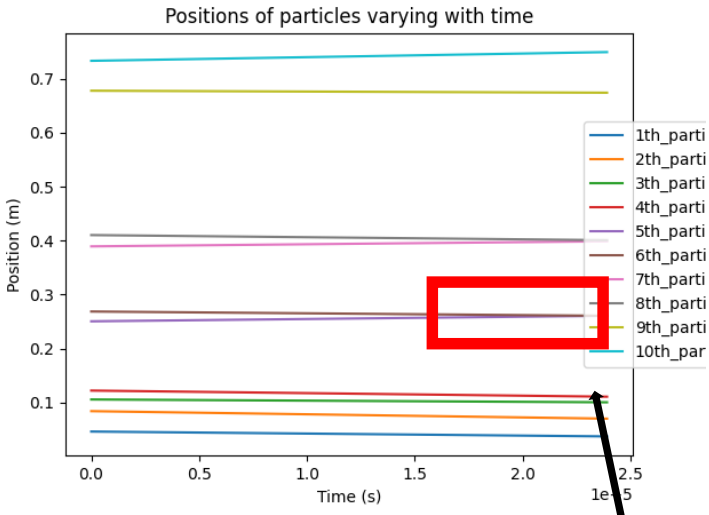


Figure 3: Positions of particles with respect to time for first collision

Collision #2:

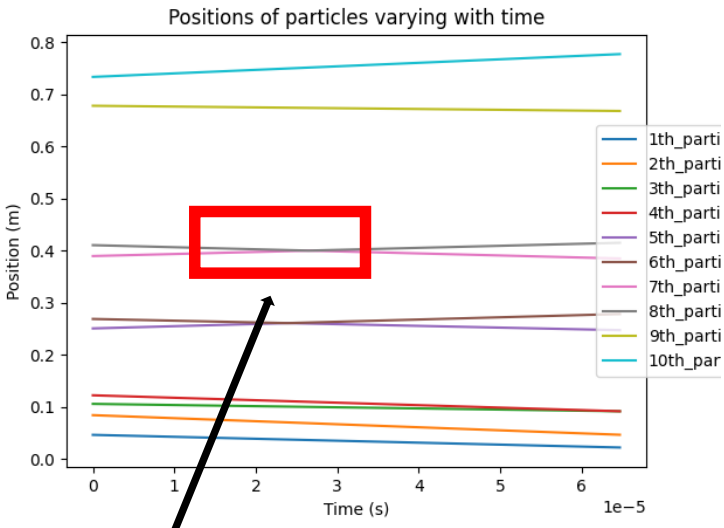


Figure 4: Positions of particles with respect to time for second collision

Next Collisions

Similarly all collisions possible in 0.0005 s are executed and the positions of all particles at different times are plotted. The final trajectory of the simulation is:

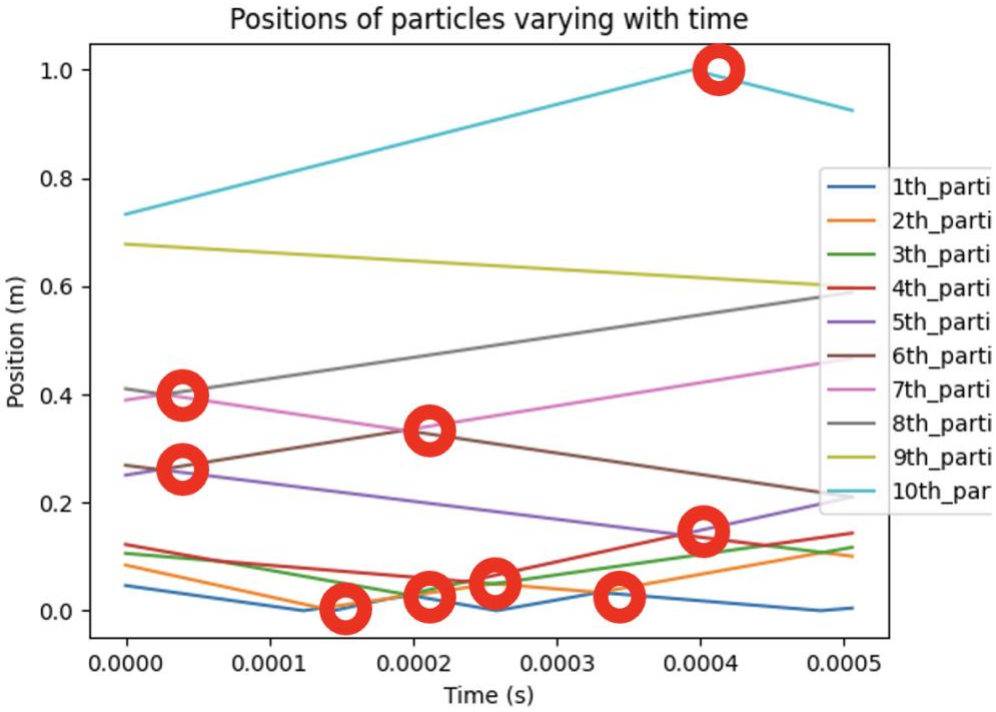


Figure 5: trajectory of all particles post during the simulation. The red circles in the above circle indicate all the points in time when collision has occurred.

Another significant observation of these trajectories is that after every collision the slopes of the two intersecting lines are exchanged. This is intuitively the exchange of velocities of particles (which is also the slope of position vs time graph for a particle).

8.2 Experiment 2

In our research, we employ a sophisticated simulation program with the primary objective of analysing the behaviour of tagged particles under varying conditions—specifically, during initialization and at saturation. Each iteration of our experiment involves defining the following parameters:

1. Number of Particles
2. Density
3. Number of Realizations
4. Simulation Time
5. Temperature

Following are some plots which were encountered while conducting some random experiments.

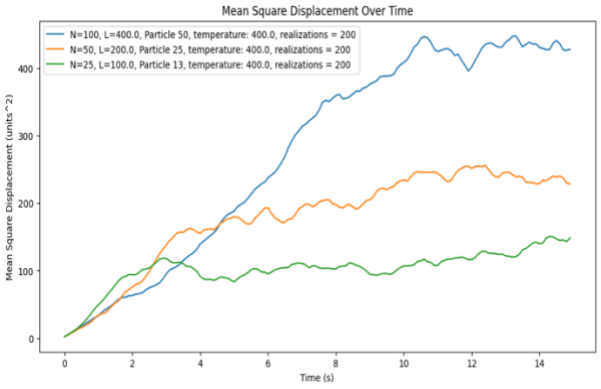


Figure 6: MSD vs time(s) for different sizes systems (temperature: 400 K, density: 0.25)

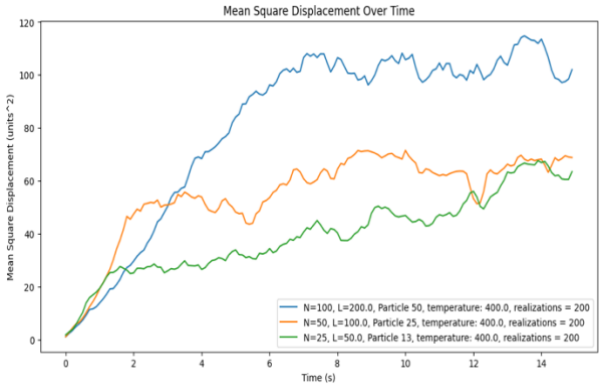


Figure 7: MSD vs time(s) for different sizes systems (temperature: 400 K, density: 0.5)

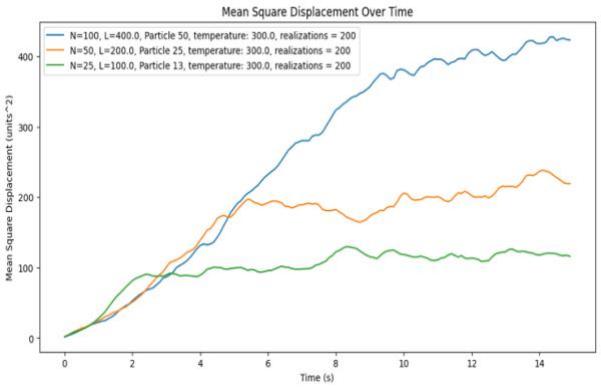


Figure 8: MSD vs time(s) for different sizes systems (temperature: 300 K, density: 0.25)

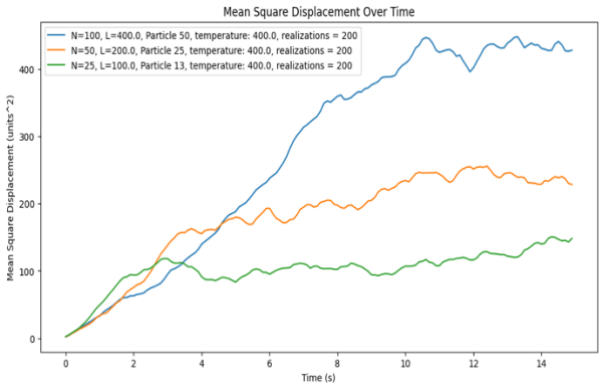


Figure 9: MSD vs time(s) for different sizes systems (temperature: 300 K, density: 0.25)

To ensure robust results, we maintain consistency in parameters such as the number of particles, density, realizations, simulation time, and temperature across multiple iterations. The variables that differ between realizations are the initial positions and velocities of particles. We also now plot *log(msd) vs log (t)*, instead of *msd vs t* to understand post saturation behaviour as well. We also increase the number of realizations to smoothen our curves. We systematically explore the influence of parameters on the relation using the below lists of values:

- **Number of Particles:** {25, 50, 100}
- **Density:** {0.5, 1.0}
- **Number of Realizations:** 300
- **Temperature:** {300 K, 400 K}

This comprehensive approach allows us to gain insights into how various factors impact the behaviour of tagged particles, facilitating a thorough understanding of their dynamics during both the initialization and saturation phases of the experiment.

Case I: Constant Density, constant temperature, varying the number of particles

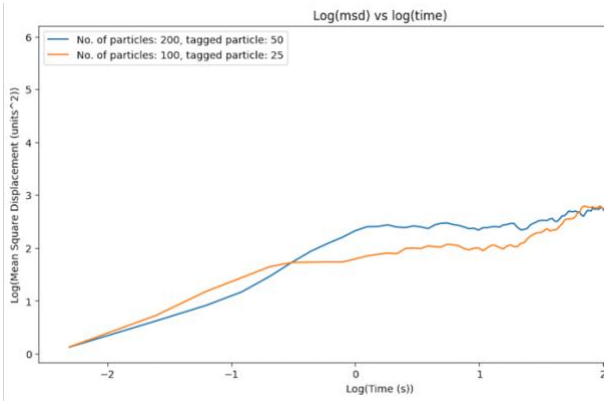


Figure 10: Log (MSD) vs Log (time) for different number of particles

(density: 0.5, temperature: 300 K)

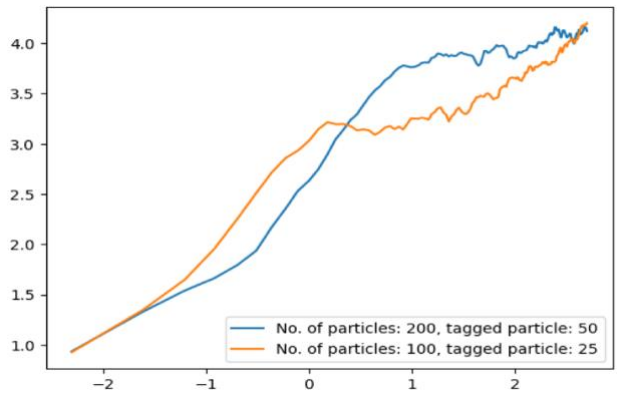


Figure 11: Log (MSD) vs Log (time) for different number of particles

(density: 0.5, temperature: 400 K)

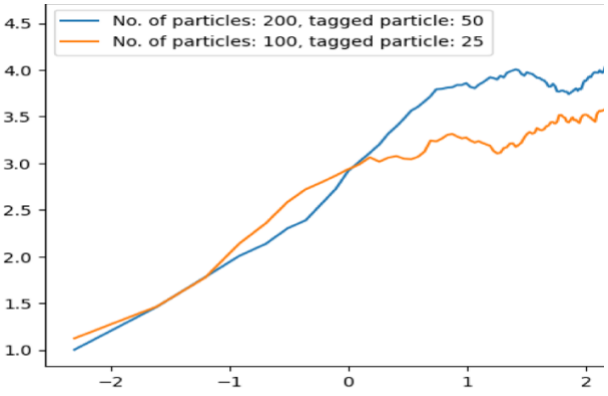


Figure 12: Log (MSD) vs Log (time) for different number of particles

(density: 1.0, temperature: 300 K)

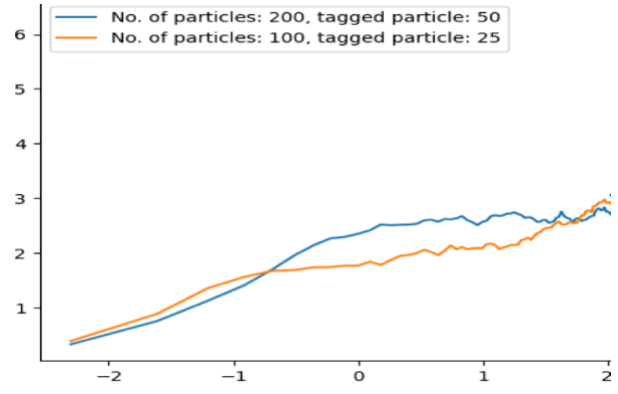


Figure 13: Log (MSD) vs Log (time) for different number of particles

(density: 1.0, temperature: 400 K)

Case II: Constant density, Constant number of particles, varying the temperature

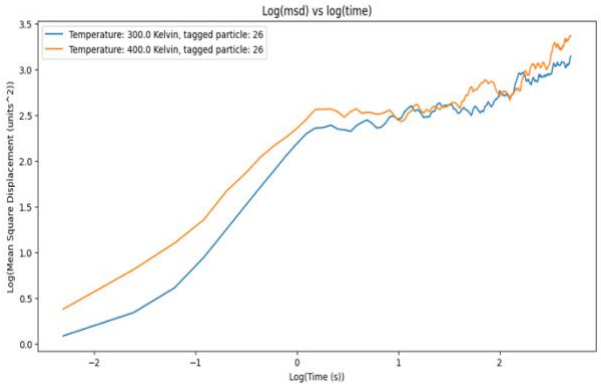


Figure 14: Log (MSD) vs Log (time) for different temperatures

(density: 1.0, #particles: 100)

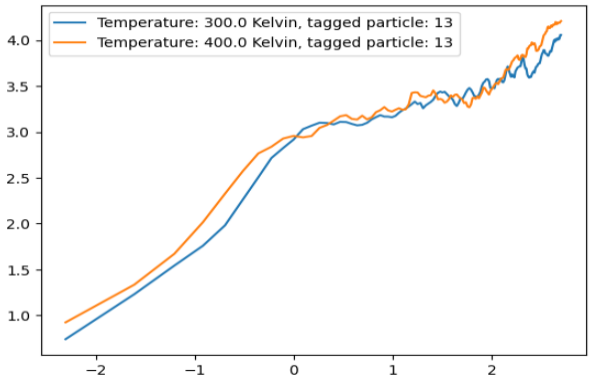


Figure 15: Log (MSD) vs Log (time) for different temperatures

(density: 1.0, #particles: 50)

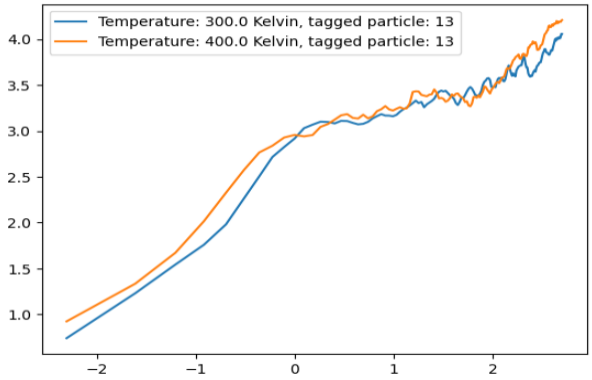


Figure 16: Log (MSD) vs Log (time) for different temperatures

(density: 0.5, #particles: 100)

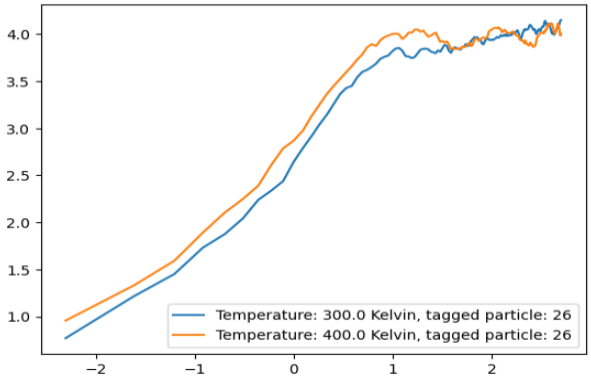


Figure 17: Log (MSD) vs Log (time) for different temperatures

(density: 0.5, #particles: 50)

Case III: Constant temperature, constant number of particles, varying the density

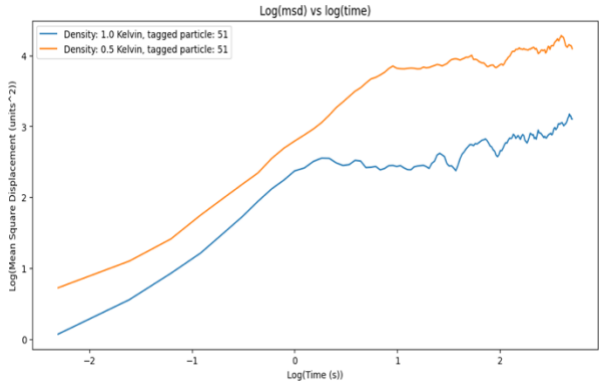


Figure 18: Log (MSD) vs Log (time) for different densities (temperature: 300 K, #particles: 100)

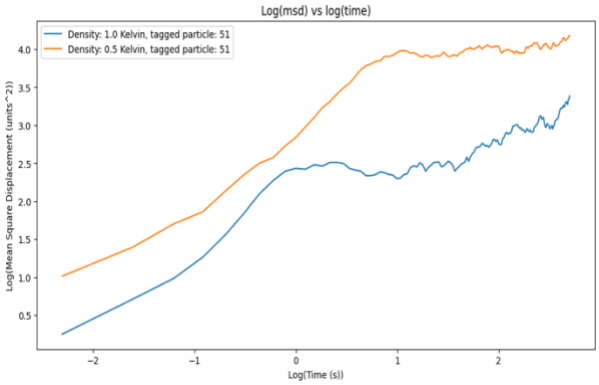


Figure 19: Log (MSD) vs Log (time) for different densities (temperature: 400 K, #particles: 100)

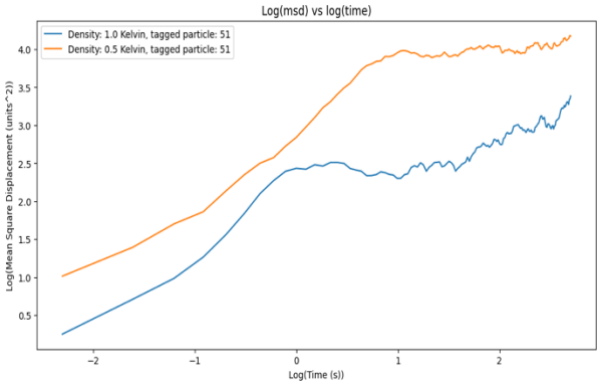


Figure 20: Log (MSD) vs Log (time) for different densities (temperature: 300 K, #particles: 50)

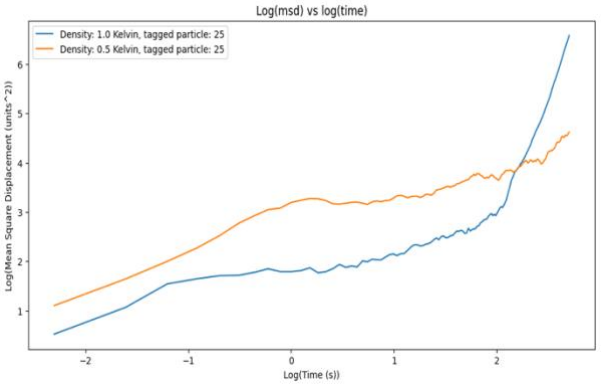


Figure 21: Log (MSD) vs Log (time) for different densities (temperature: 400 K, #particles: 50)

9. Conclusions

The experimental results from our study provide insightful observations about the relationship between the mean square displacement (MSD) of a tagged particle and time, particularly in two distinct temporal regimes. During the initial phase, there is a clear linear relationship between MSD and time. This linear behaviour is indicative of a deterministic process, where the movement of the particle can be predicted based on initial conditions and governing physical laws.

As time progresses, we observe a notable transition in this behaviour. After a specific finite period, the MSD reaches a saturation point. Beyond this point, the MSD ceases to increase significantly and stabilizes around a constant value. This saturation of the MSD is a critical observation, suggesting a shift from deterministic to stochastic behaviour in the particle's movement.

The accompanying graphical representation in the report illustrates this transition effectively. The initial linear phase, characterized by a steady increase in MSD with time, is followed by a plateau phase where the MSD levels off. This pattern reaffirms findings from previous experiments, substantiating the theoretical models that predict such behaviour in particle dynamics.

In the pre-saturation phase, the deterministic nature implies that the particle's motion is largely governed by predictable and regular patterns. This could be attributed to the dominant influence of factors like initial momentum and external forces. In contrast, the post-saturation stochasticity indicates a regime where the particle's motion becomes more influenced by random fluctuations, possibly due to interactions with other particles or environmental variables.

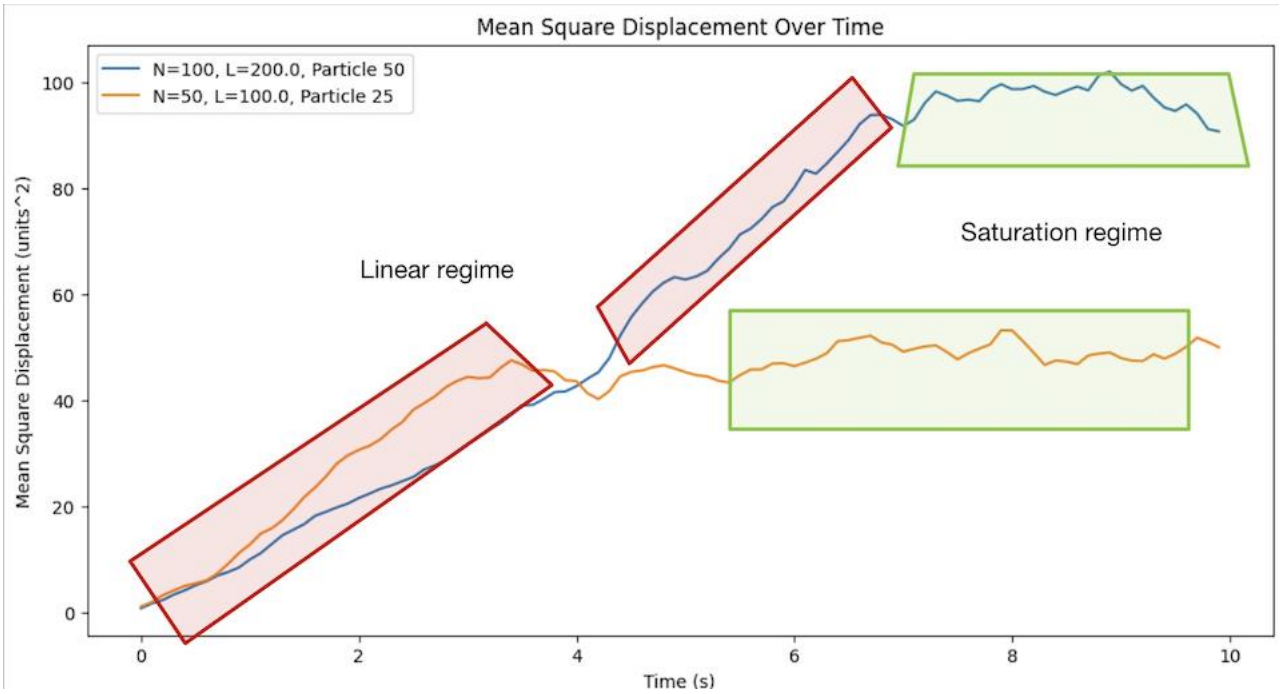


Figure 24: Schematics of linear and saturated regime in single file diffusion

10. References

1. Jepsen, D.W.: J. Math. Phys. **6**, 405 (1965)
2. Roy, A., Narayan, O., Dhar, A., & Sabhapandit, S. (2012, December 28). Tagged Particle Diffusion in One-Dimensional Gas with Hamiltonian Dynamics. *Journal of Statistical Physics*, *150*(5), 851–866. https://doi.org/10.1007/s10955-012-0673-9
3. Bannerman, M. N., Sargant, R., & Lue, L. (2011, August 29). DynamO: a free ${\cal O}$(N) general event-driven molecular dynamics simulator. *Journal of Computational Chemistry*, *32*(15), 3329–3338. https://doi.org/10.1002/jcc.21915
4. Paul, A., Laurila, T., Vuorinen, V., & Divinski, S. V. (2014). Fick's Laws of Diffusion. *Thermodynamics, Diffusion and the Kirkendall Effect in Solids*, 115–139. https://doi.org/10.1007/978-3-319-07461-0_3
5. Wong, K. S., & Wong, H. (2022, February 1). *Understanding the Law of Conservation of Momentum in One-Dimensional Collisions Between Two Objects Using the Velocity Space Approach*. The Physics Teacher; American Association of Physics Teachers. https://doi.org/10.1119/5.0024603
6. Ooshida, T., Goto, S., & Otsuki, M. (2018, August 2). Collective Motion of Repulsive Brownian Particles in Single-File Diffusion with and without Overtaking. *Entropy*, *20*(8), 565. https://doi.org/10.3390/e20080565
7. Narayan, O., Ramaswamy, S.: Phys. Rev. Lett. **89**, 200601 (2002)
8. van Beijeren, H.: Phys. Rev. Lett. **108**, 180601 (2012)
9. Dhar, A.: Adv. Phys. **57**, 457 (2008)
10. Hurtado, P.I.: Phys. Rev. Lett. **96**, 010601 (2006)
11. The Gaussian Distribution. (2019, March 15). *Probability, Random Variables, Statistics, and Random Processes*, 201–238. https://doi.org/10.1002/9781119300847.ch7
12. Harris, C.R., Millman, K.J., van der Walt, S.J. et al. *Array programming with NumPy*. Nature 585, 357–362 (2020). DOI: 10.1038/s41586-020-2649-2. (Publisher link).
13. Maxwell–Boltzmann distribution. (2023, September 3). In *Wikipedia*. https://en.wikipedia.org/wiki/Maxwell%E2%80%93Boltzmann_distribution