Molecular Dynamics Simulation using the Hard Sphere Potential to examine Diffusion

By: Fang Jiunn Ewe, Scott J McCormack and Seth Thor

Introduction

Molecular Dynamics (MD) is a computational simulation technique used to study the movement of objects based off of Newton's equations ($\mathrm{F_i} = \mathrm{m_i a_i}$) [1]. Objects can be made to resemble physical particles by defining a potential landscape. By differentiating this potential landscape ($\mathrm{F_i} = -\nabla_\mathrm{i}[V(r_1, r_2, \dots, r_N)]$), the force applied to each object in the system can be determined. By integrating forward (and backwards) using a reliable integrator (such as the Velocity-Verlet algorithm) one can predict atomic positions and velocities of system over all time [1].

The force on each object must be calculated at each time step based off of the defined potential of the system. This can be computationally intensive, resulting in large simulations. There are, however, simple potentials which do not have to be recalculated, allowing for efficient code. An example of one these potentials is the hard sphere potential (Equation 1).

Equation 1: Hard Sphere Potential [2] [3] [4].

$$\mathrm{V}(\mathrm{r_i}, \mathrm{r_j}) = \begin{cases} 0 \text{ if } |\mathrm{r_i} - \mathrm{r_j}| \geq \sigma \\ \infty \text{ if } |\mathrm{r_i} - \mathrm{r_j}| \leq \sigma \end{cases}$$

Here, $\mathrm{r_i}$ and $\mathrm{r_j}$are the position vectors of objects *i* and *j* respectively. σ is the separation between the position vectors at which the potential felt between the two objects will be infinite over an infinitesimal time. By setting σ to the sum of radii of two spheres in contact, the hard sphere potential can model the interactions between any spherical objects where the interactions are defined as ideal. An ideal interaction is one in which there is no attraction or repulsion felt between the objects, and the system depends entirely only on conservation of momentum and kinetic energy. Examples of physical systems which have (nearly) ideal interactions include noble gases (Neon, Argon etc…), billiard balls, marbles, and non-interacting colloids. A plot of the hard sphere potential is shown in Figure 1.

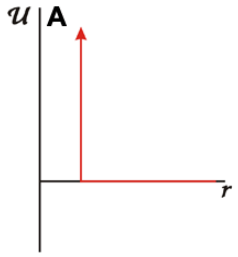


Figure 1: Hard Sphere Potential [2].

There are multiple physical phenomena which occur in non-interacting colloids which have proven very difficult to elucidate. One of which is the distinct regions of diffusion within a colloidal system

based on the observation time. By using molecular dynamics with a hard sphere potential, one can show that these regions of diffusion come about by physical space constraints within the system, rather than by favorable interactions.

The aim of this paper is to show that these regions of diffusion can easily be created and quantified using a simple hard sphere potential. The same code will be used to show that physical phenomena such as solidification can occur in non-interacting systems, an idea which has only recently been observed in the literature.

Hard Sphere Algorithm

Hard Sphere Dynamics and Kinematics

As mentioned above, the hard sphere potential is one which results in an infinite potential applied over an infinitesimal time when two object reach a length σ of each other. This results in the two objects repelling each by means of a finite impulse (product of force and time). The impulse imparts an equal and opposite change in momentum of the two objects. By applying conservation of energy, the impulse between the two objects can be defined according to Equation 2 and Figure 2 [4].

Equation 2: Change in Momentum based on Enskog Theory [5].

$$\Delta \mathrm{p} = \frac{2\mathrm{m_i m_j}}{\mathrm{m_i + m_j}} \frac{\mathrm{v_{ij}}.\widehat{\mathrm{r_{ij}}}}{\sigma} \widehat{\mathrm{r_{ij}}}$$

Where $\Delta \mathrm{p}$ is the change in momentum (a vector) from the collision, $\mathrm{m_i}$ and $\mathrm{m_j}$ are the masses of the colliding particles, v_{ij} and r_{ij} is the velocity vector and position vector respectively.

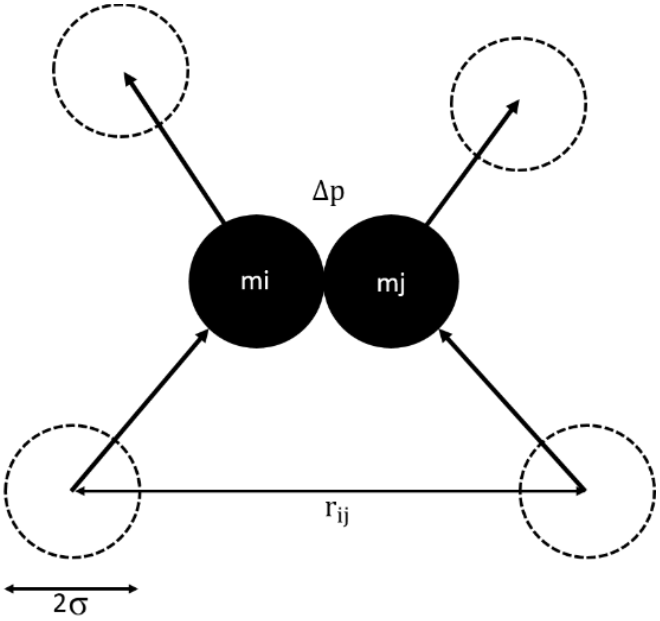


Figure 2: Depiction of equation 2

It should be noted that Equation 2 only takes into account the collisions of two particles. This is a result of Enskog theory, where the assumption is made that only two particles will collide at any one time. While

this is not always true, the probability of attaining more than two particles within a single collision is extremely low. This assumption greatly increases computation speed, as three-body or even multi-body calculations are extremely time intensive to solve.

When the objects are not colliding, they move based on their unchanging velocity vectors. Since there are no forces on acting on them, they do not undergo any acceleration. The change in position is thus determined from Equation 3.

Equation 3: Position [2] [3] [4].

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

Equation 4: Position vector related to particle diameter, σ [2] [3] [4].

$$\left|r_j(t + \Delta t) - r_i(t + \Delta t)\right|^2 = \sigma^2$$

By using Equation 3 and Equation 4 and expanding yields the quadratic equation shown in Equation 5.

Equation 5: Quadratic Equation used to solve for time step between collisions [2] [3] [4].

$$r_{ij}^2(t) - \sigma^2 + 2r_{ij}(t).v_{ij}(t)\,\Delta t + v_{ij}^2(t)\Delta t^2 = 0$$

where $r_{ij} = r_j - r_i$ and $v_{ij} = v_j - v_i$. Solving Equation 5 for Δt will give the time elapsed until the next collision in the system (Equation 6).

Equation 6: Time between collisions [2] [3] [4].

$$\Delta t_{ij} = \frac{-r_{ij}(t).v_{ij}(t) - \sqrt{\left(r_{ij}^2(t) - \sigma^2\right)v_{ij}^2(t) + \left(r_{ij}(t).v_{ij}(t)\right)^2}}{v_{ij}^2(t)}$$

Hard Sphere Molecular Dynamics Implementation

Since the particles move under constant velocities unless a collision occurs, we can calculate all the future collisions of any particle from the start. We do not even need to use any integration scheme, as we can simply move to the next collision instant directly. Thus, the core of the hard sphere algorithm only involves keeping track of the collision times and the colliding partners of all particles, and then selecting the smallest Δt to step to.

When solving for collisions, we only need to observe the discriminant to determine whether a pair will collide. If the discriminant is positive, then the two particles are approaching each other, but may not collide. We must then observe whether r_{ij} and v_{ij} are acute or obtuse to each other: the particles will only collide if they are acute, else the particles will miss. These to criteria are captured in Equation 7 and Equation 8:

Equation 7: The discriminant is positive [2].

$$\left(r_{ij}^2 - \sigma^2\right)v_{ij}^2 + \left(r_{ij}(t).v_{ij}(t)\right)^2 > 0$$

Equation 8: [2].

$$r_{ij}(t).v_{ij}(t) < 0$$

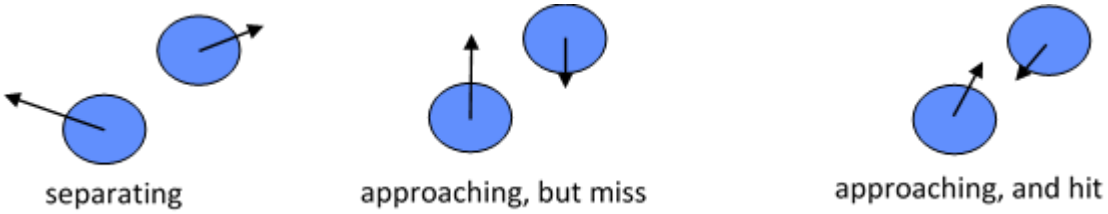


Figure 3: Particle collision pairs based on relative velocities and relative positions.

All particle pairs that satisfy these two criteria will collide, have their time remaining till collision, Δt_{ij}, calculated and stored, as well as their identity. The system is then progressed forward in time by the smallest Δt_{ij}. In other words, all particles undergo free-flight dynamics until the time of the next collision. After this single collision, all collision pairs that were computed in the previous iteration will have changed. Therefore, all possible particle pairs must be rechecked against the above criteria to determine future collision pairs and the next collision time. This process is repeated for as many steps as desired. It is important to note that, when implemented correctly, the algorithm runs quickly because it will only yield snapshots of the system at collision times. Each step in the simulation is actually Δt_{ij} instead of an even time step that can be set. Viewing the trajectory for this event-based simulation will thus show a choppy simulation with rapidly moving hard sphere particles.

In order to capture the real-time motion of the particles through even time-steps, the algorithm requires a much more complex setup that keeps track of an overall simulation timeline in addition to the collision time intervals. After computing the earliest collision time between a pair of particles, the system is progressed forward at a specified time step, dt, until the collision occurs. At the collision time, the new system dynamics and the next collision time is computed as normal. However, the difficulty with a real-time hard sphere simulation arises when a collision time occurs in between two simulation time-steps. In order to account for these cases, the algorithm is set to check after each dt forward if the simulation time has exceeded the any collision time Δt_{ij}. If yes, then the system is brought back to the previous simulation time step and progressed only up to the collision point. The collision dynamics and the next collision time are computed, then the system is progressed forward to either the next simulation time step or the next collision time point, whichever occurs first. Figure 4 illustrates this implementation. By running a hard sphere simulation in this fashion, the real-time dynamics will be captured in much higher resolution. If a small enough simulation time-step is used, many collisions occur at the simulation time steps, and the algorithm avoids having to backtrack. Although collisions that still occur between simulation time steps are not captured in the trajectory output, this real-time simulation approach allows for better estimates of system properties such as diffusion coefficients, pair correlation functions, etc.

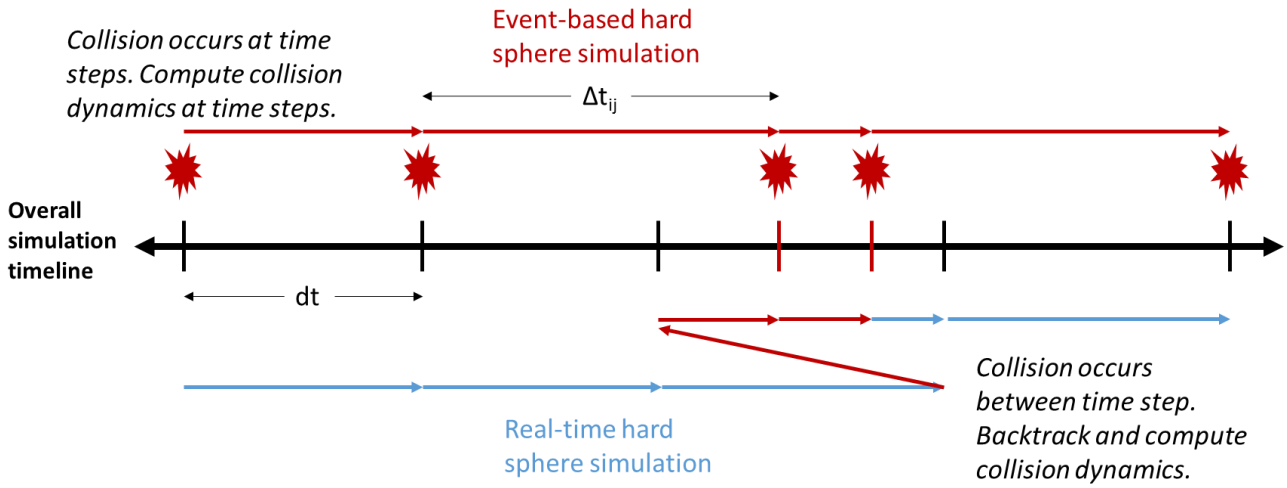


Figure 4: Illustration of hard sphere implementation [2].

Diffusion Determination and Physical Phenomena

Diffusion and Molecular Dynamics

In Molecular dynamics, the diffusion constant (D) can be calculated form the Mean Squared Displacement (MSD) and/or the velocity autocorrelation function (C_v) (Equation 9).

Equation 9: Mean Squared Displacement (MSD) [6].

$$\mathrm{MSD(t)} = \langle \left(\mathrm{r_i(t) - r_i(0)}\right)^2 \rangle$$
$$\mathrm{MSD(t)} = \langle \mathrm{r_i(t)^2} \rangle + \mathrm{r_i(0)^2} - 2\mathrm{r_i(0)}\langle \mathrm{r_i(t)} \rangle$$

Where $\mathrm{r_i(t)}$ is the position of an atom at time t, $\mathrm{r(0)}$ is the position of an atom at t=0. The average is taken over all atoms at each time step. The MSD is related to the diffusion coefficient by Equation 10.

Equation 10: Diffusion constant (D) from mean squared displacement (MSD) [6].

$$\mathrm{MSD(t)} = 6\mathrm{Dt}$$

Where D is the diffusion constant and t is the time. The diffusion constant can also be calculated form using the velocity autocorrelation function.

The velocity autocorrelation function essentially compares the initial velocity of each atom, with the velocity of the atom at a later time step (Equation 11).

Equation 11: Velocity autocorrelation function (C_v) [7].

$$\mathrm{C_V(t)} = \langle \mathrm{v_i(t).v_i(0)} \rangle$$
$$\mathrm{C_V(t)} = \frac{1}{\mathrm{N}} \sum_{\mathrm{i=1}}^{\mathrm{N}} (\mathrm{v_i(t).v_i(0)})$$

Where $v_i(t)$ is the velocity of particle *i* at time t, $v_i(0)$ is the initial velocity of particle *i*. The average is taken over all atoms at each time step. A typical plot of the velocity autocorrelation function is shown in Figure 5.

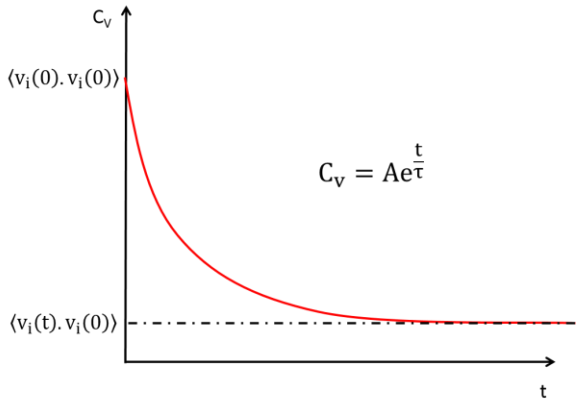


Figure 5: A typical plot of the velocity autocorrelation function .

Initially, the system will show high correlation as the particles continue in their free flight. As time progresses, the particles collide and lose their initial velocity, such that the correlation goes to zero. The velocity autocorrelation function averages over all the particles in the system, and can be used to determine the diffusion constant of the particles (Equation 12).

Equation 12: Diffusion constant (D) from velocity autocorrelation (C_v) [7].

$$D = \frac{1}{3}\int_0^\infty \langle v_i(0).v_i(t)\rangle \, dt$$

Diffusion Dynamics in Non-Interacting Colloids

The Diffusion constant for a system varies depending on the time scale of observation. For a non-interacting colloidal system, there are 3 distinct regimes. The short time diffusion (D_S), the intermediate diffusion (D_I) and the long time diffusion (D_L) [5].

The short time diffusion is the diffusion on a time scale where the atom does not come into contact with any other atoms. It is typically very small and very difficult to observe experimentally. Figure 6 shows a depiction of short time diffusion. Once the observation time exceeds a critical time, the diffusion constant will change to the intermediate diffusion constant. This time is known as the short characteristic diffusion time τ_S, and typically occurs when the atom comes into contact with another atom.

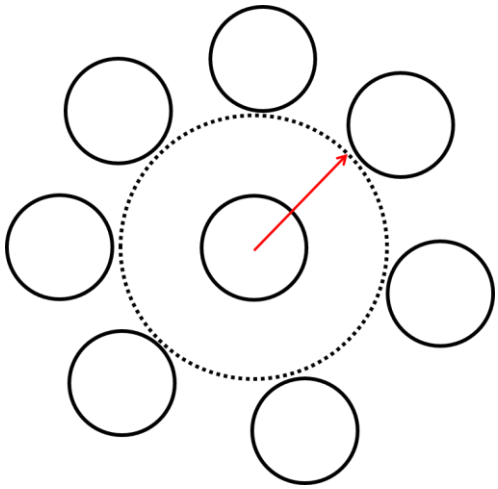


Figure 6: Depiction of the short time diffusion. The red arrow represents the time-length scale over which the atom can explore. Over this time scale, the center atom does not have enough time to reach its nearest neighbors. During this time there will be no collisions [5] [8].

The intermediate diffusion (D_I) constant is typically the smallest of the three. It is the smallest due to a phenomena known as caging. It is called this because the atom being observed becomes trapped by its surrounding atoms. Figure 7 shows a depiction of the intermediate diffusion constant. Once the observation time increases above the relaxation time (τ_L), the time it takes for the atom to escape the "cage", the diffusion constant will change to the long time diffusion constant.

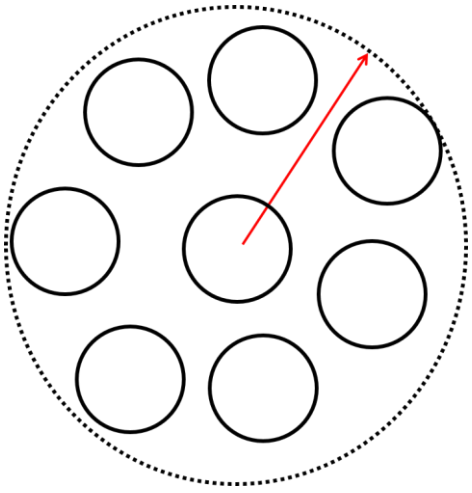


Figure 7: Depiction of the intermediate diffusion (D_I). The red arrow represents the time-length scale over which the atom can explore. Over this time scale, the center atom will have enough time to reach its nearest neighbors, but will not have enough time to pass them. When the center atom comes in contact with the nearest atoms, it will be pushed back to its center position. This phenomenon has been termed caging. In this regime the diffusion constant will be the lowest of the three regimes [5] [8].

The longtime diffusion constant is the diffusion constant which arises when the atoms are able to diffuse throughout the material. Figure 8 shows a depiction of the longtime diffusion constant. It is this

value which is typically measured experimentally. Ironically, the longtime diffusion constant is the hardest to measure using MD simulations. This is because the time the relaxation time τ_L is typically too large for most simulations. This is why most long time diffusion constants calculated using simulation tend to be slightly smaller than the measured long time diffusion constant. The longtime diffusion constant is typically measured using the autocorrelation method, as it involves an integral over all time.

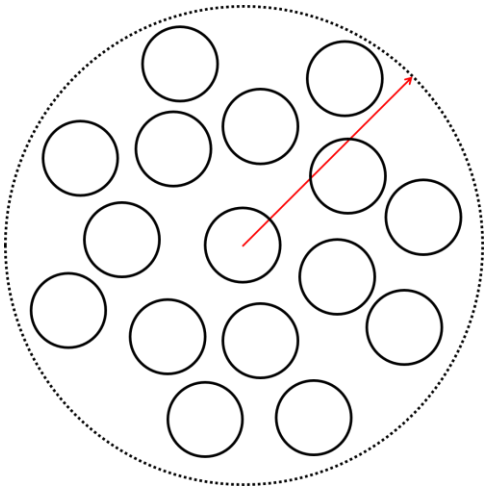


Figure 8: Depiction of the longtime diffusion (D_L). The red arrow represents the time-length scale over which the atom can explore. Over this length scale, the center atom is able to explore the entire system. The longtime diffusion constant is typically used for bulk diffusion calculations [5][8].

These different diffusion constants can be seen more clearly when observing a log MSD versus log time plot. Here we can see the short diffusion regime, labeled as (I), the intermediate diffusion regime, labelled as (II) and the long time diffusion regime, labelled as (III) (Figure 9).

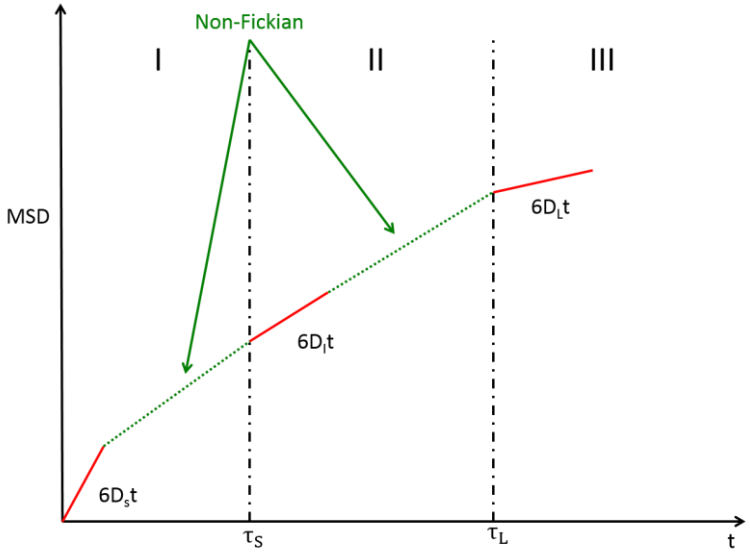


Figure 9: Plot of MSD versus time showing the different diffusion constants over different time scales [5] [8].

These three regimes will appear in all non-interacting spherical colloidal systems as a function of volume fraction. If the volume fraction is small, the intermediate regime will not be present. This is because there are not enough surrounding particles to cause “caging”, resulting in a nice transition between short time and longtime diffusion. The critical volume fraction for caging to occur has been reported in the literature is as low as 0.05 for some systems [5].

It is worth noting that these three regimes are observed exclusively in non-interacting spheroidal collides. If colloids with interacting potentials/non-spherical are simulated, other regimes may appear.

Simulations

Molecular Dynamic Simulation using Hard Sphere Potential

Aim

To determine the optimal parameters for implementing the molecular dynamic hard sphere potential simulation to study diffusion phenomena.

Methodology

The hard sphere molecular dynamics code was used to perform two different simulations: (i) one with periodic boundary conditions and the other with (ii) reflective boundary conditions. The initialized parameters are summarized in Table 1.

Table 1: Initialized parameters

Parameter	**Value**
Number of atoms (N):	64
Number of steps	50,000
Atom diameter	$71 \times 10^{-12}\ m$
Volume fraction, ϕ	0.01
Temperature (used to initialize velocities from a maxwell-boltzman distribution)	20 K
Recorded Time step (the actual time step is based off the number of collisions)	$1 \times 10^{-14} s$
Atomic array	Simple cubic

The total energy, the velocity-velocity autocorrelation function, Mean Squared Displacement and Pair distribution function were computed for both simulations.

Results and Discussion

As this is the first time the hard sphere MD simulation was run, the total energy of the system was monitored to ensure that energy was conserved throughout the simulation. This is shown in Figure 10 and Figure 11.

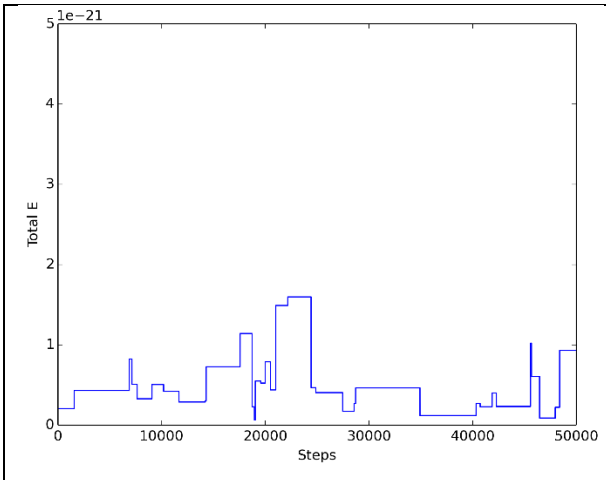


Figure 10: Plot of Energy versus number of steps to show that the molecular dynamics

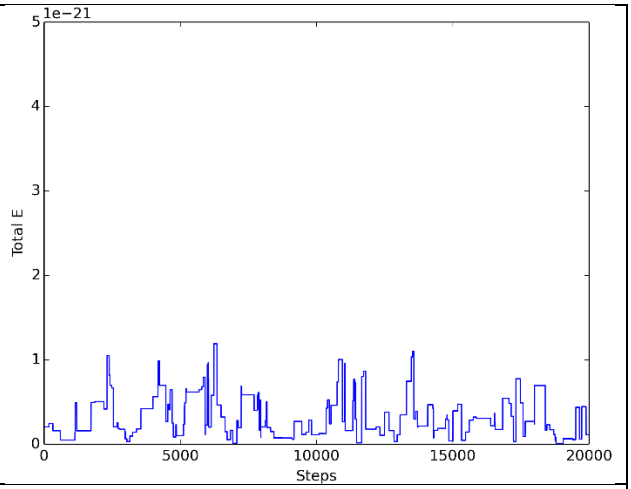


Figure 11: Plot of Energy versus number of steps to show that the molecular dynamics

hard sphere simulation with periodic boundary conditions is stable.	hard sphere simulation with reflective boundary conditions is stable.

The total energy for both is relatively constant throughout the entire simulation, proving that both systems are stable.

The MSD was calculated for both the periodic boundary simulation and the reflective boundary condition simulation. This is shown in Figure 12 and Figure 13.

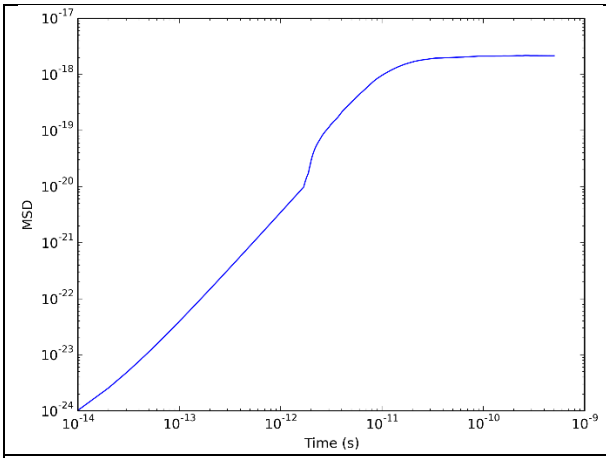


Figure 12: MSD versus time for periodic boundary conditions.

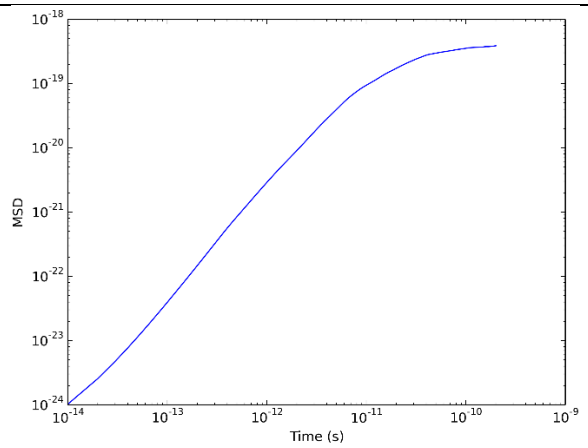
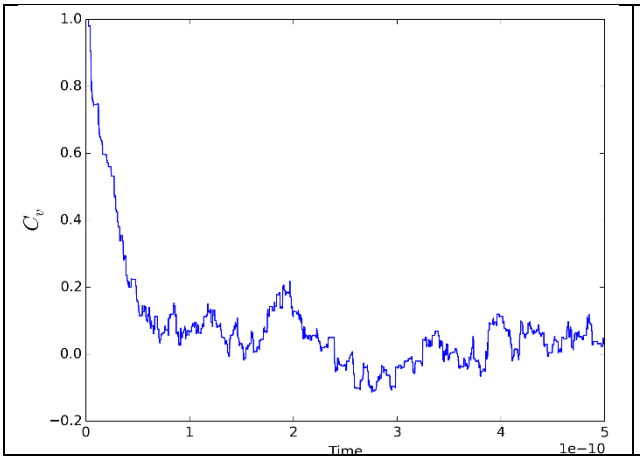


Figure 13: MSD versus time for reflective boundary conditions.

The MSD for the periodic boundary condition (Figure 12) does not reflect that of which is expected for MD modelling of diffusion. The reason for this is because the periodic boundary conditions are effecting the set volume fraction of the system. This prevents the observation of the 3 expected regimes. When compared to the reflective boundary conditions (Figure 13), two of the expected diffusion regimes are clearly visible. The first, the short time diffusion regime is not clearly visible due to the time step.

The Velocity-velocity autocorrelation function was calculated for both the periodic boundary simulation and the reflective boundary condition simulation. This is shown in Figure 14 and Figure 15.



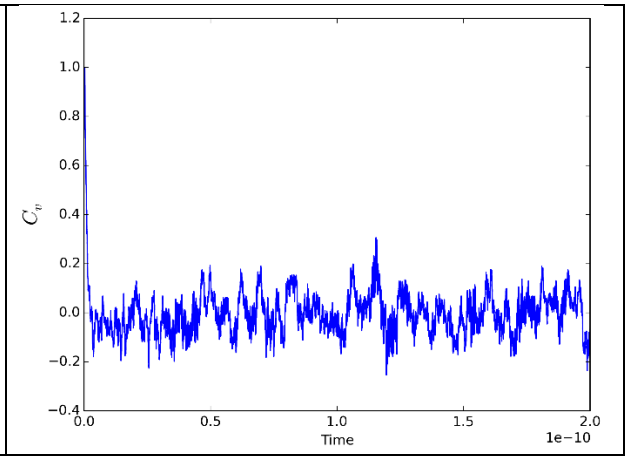


Figure 14: Velocity-Velocity autocorrelation function for periodic boundary conditions	Figure 15: Velocity-Velocity autocorrelation function for reflective boundary conditions

It can be seen that the reflective boundary condition randomizes at a much faster rate than the periodic boundary condition. This is because the velocities in the reflective boundary system will change more frequently, as they change when the particles come into contact with the wall of the box.

Conclusion

In conclusion, the periodic boundary conditions hard sphere simulation did not capture the physical phenomena which was trying to be modelled. All further computations were performed using the reflective boundary conditions.

Diffusion and characteristic times when initialized in simple cubic arrangement

Aim

To determine the diffusion coefficients, the characteristic times and the phases when the simulation is initialized in a simple cubic arrangement and reflective boundary conditions.

Methodology

The particles were initialized evenly spaced in a simple cubic array within a cubic box of size L. The size of the box was determined based on the volume fraction. The molecular dynamic hard sphere code was ran with the initialization parameters shown in Table 2.

Table 2: Initialized parameters

Parameter	**Value**
Number of atoms (N):	64
Number of steps	50,000
Atom diameter	$71 \times 10^{-12}\ m$
Volume fraction, ϕ	0.1, 0.3, 0.49, 0.5, 0.52
Temperature (used to initialize velocities from a maxwell-boltzman distribution)	20 K
Recorded Time step (the actual time step is based off the number of collisions)	$1 \times 10^{-14} s$
particle array	Evenly Spaced Simple cubic

The initial velocity and the first collision time was used to determine the short time diffusion, $\mathrm{D_S}$ and the short characteristic time, τ_S. The MSD was used to calculate the intermediate diffusion, $\mathrm{D_I}$ and the long characteristic time, τ_L. The $\mathrm{C_v}$ was used to determine the long time diffusion $\mathrm{D_L}$. $\mathrm{g(r)}$ was used to determine the volume fraction at which the system began to solidify.

Results/Discussion

It is worth noting that the simulation fails when the volume fraction is set to 0.52. This is because the reflective boundary is pressing against the particles which are in face center cubic arrangement. When the code tries to compute the next collision time, it finds that each particle is touching each other. Since the code can only a two-body collision (Enskog theory) at any one time, it gets caught in a loop where it continually tries to compute the next collision time, which is 0 as the atoms already colliding. Although the simulation could not be run, it could be visualized to see that the particles were sitting on the correct lattice sites.

The mean squared displacement prior to the first collision and the first collision time were used to calculate the short time diffusion as shown in Table 3.

Table 3: Short time diffusion

ϕ	$\langle \mathbf{x} \rangle^2$	$\mathbf{t_{collisions} = \tau_S}$	$\mathbf{D_S}$
0.1	4.74×10^{-22}	6.35×10^{-13}	7.42×10^{-10}
0.3	4.33×10^{-23}	1.78×10^{-13}	7.95×10^{-10}
0.49	1.66×10^{-24}	1.54×10^{-14}	8.32×10^{-10}
0.5	1.66×10^{-24}	1.02×10^{-14}	8.23×10^{-10}
0.52	-	-	-

The mean squared displacement prior to a collision and the collision time are small and decrease with increasing volume fraction. This is expected, as the probability of a collision is dependent on the amount of free space in the system. If the volume fraction increases, this amount of free space decreases. D_S is relatively constant with respect to volume fraction. This is expected as D_S is defined as the diffusion prior to a collision. Hence it is dependent on the parameters of the particle and the velocity of the particle. In this simulation, the initial velocity, the size and mass of the particles are held constant. The mean squared displacement and collision times could not be calculated for a volume fraction of 0.52 as all the particles are touching.

The MSD versus time is shown in Figure 16, 17, 18 and 19.

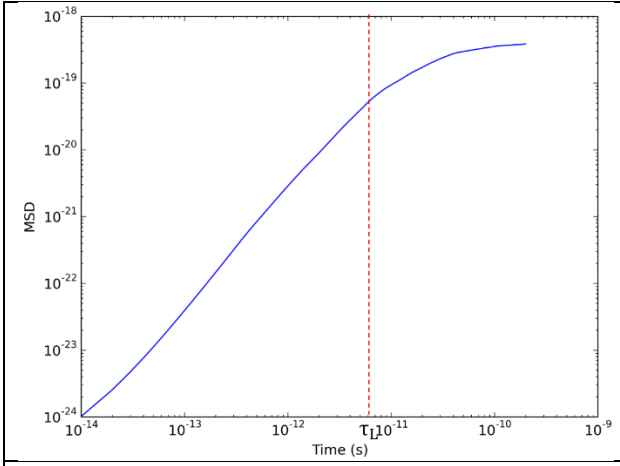


Figure 16: MSD versus time for $\phi = 0.1$ showing the long time characteristic time

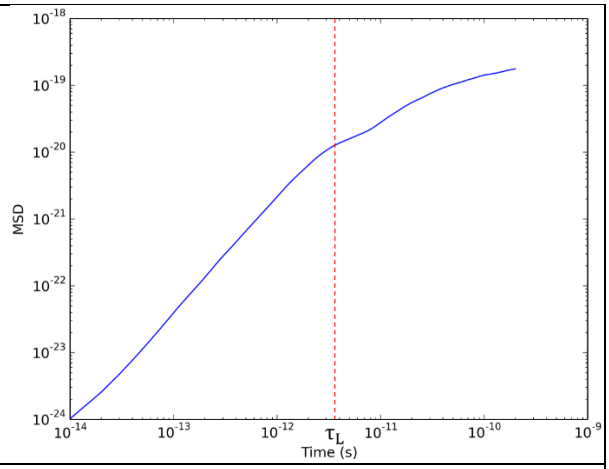


Figure 17: MSD versus time for $\phi = 0.3$ showing the long time characteristic time

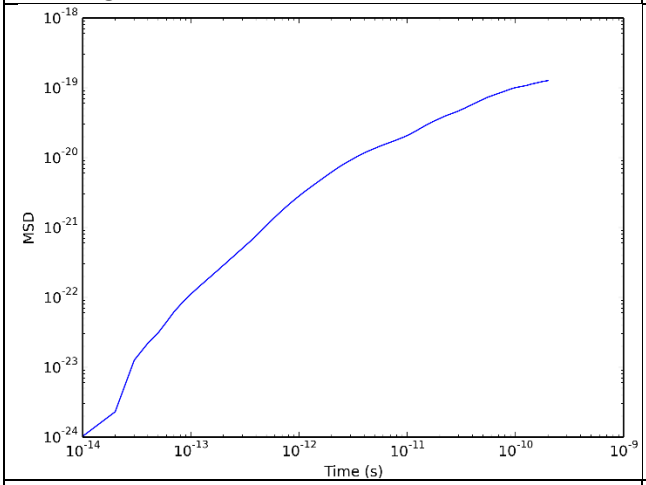


Figure 18: MSD versus time for $\phi = 0.49$. Due to the spike, the characteristic time could not be accurately determined

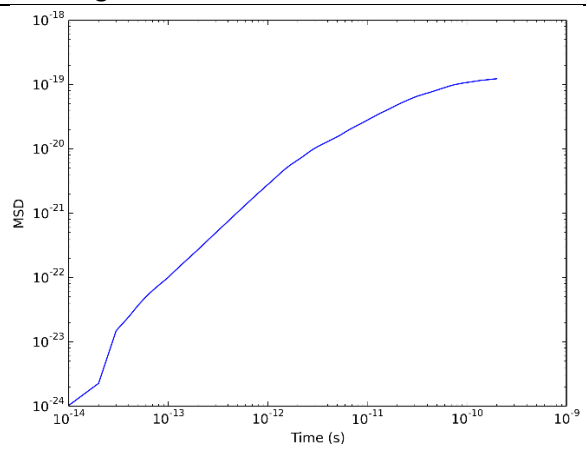


Figure 19: MSD versus time for $\phi = 0.5$. Due to the spike, the characteristic time could not be accurately determined

From the MSD, the D_I has been calculated, and is summarized in Table 4. Here, it can be seen that the D_I (the gradient) decreases with increasing volume fraction. This again is expected as there are now more particles to impede the motion of particles, resulting in a reduction in the D_I. The τ_L can also be seen to decrease with an increase in the volume fraction.

Figures 20, 21, 22 and 23 show a plot of C_V versus time for a series of volume fractions. It is difficult to see any key relations from the C_V plots other than the fact that the system is equilibrating. This is significant because it signifies that particles are escaping the "cage" (the intermediate regime) and achieving their D_L.

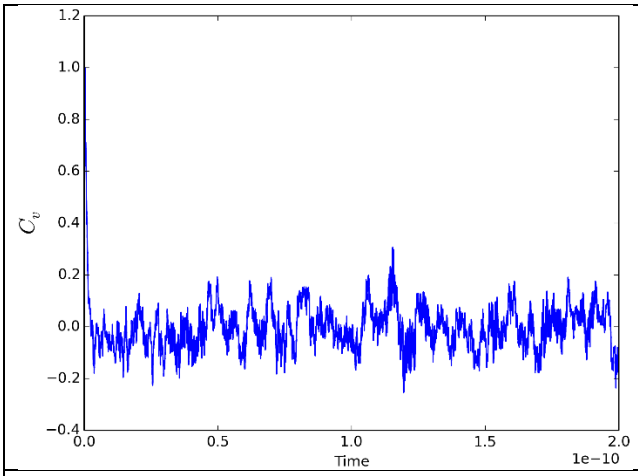


Figure 20: Velocity-Velocity autocorrelation function for $\phi = 0.1$

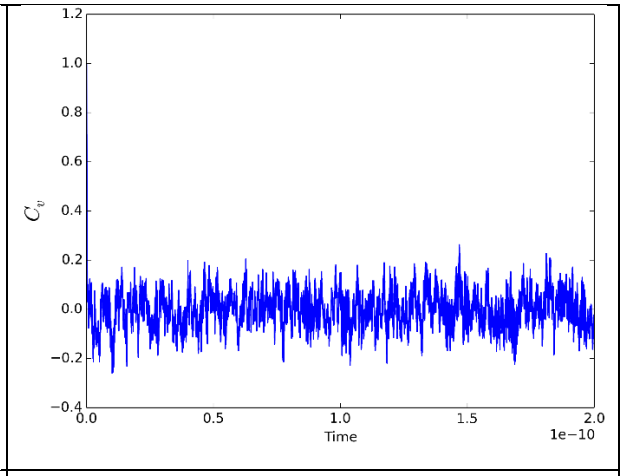


Figure 21: Velocity-Velocity autocorrelation function for $\phi = 0.3$

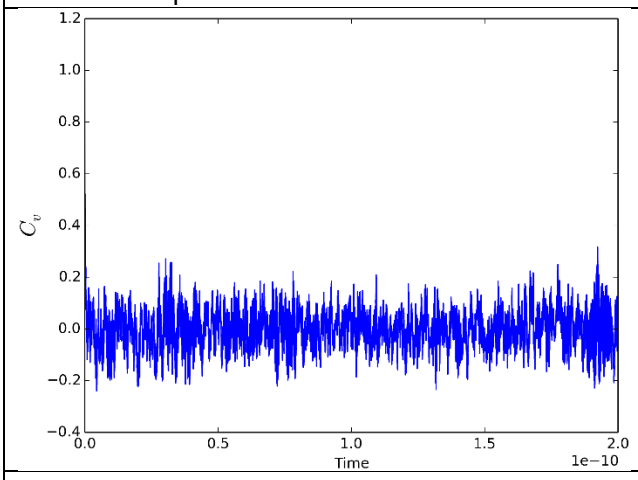


Figure 22: Velocity-Velocity autocorrelation function for $\phi = 0.49$

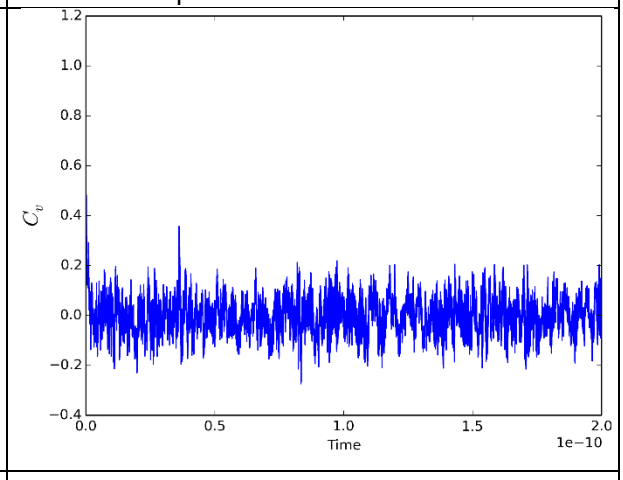


Figure 23: Velocity-Velocity autocorrelation function for $\phi = 0.5$

D_L has be calculated from the Velocity-Velocity autocorrelation function and is summarized in Table 4. D_L decreases with an increase in volume fraction. This is expected as the path of diffusion becomes more tortuous if there are more particles in the same volume.

Table 4: Summary of all diffusion and characteristic times determined for the simple cubic initialized simulation.

ϕ	D_s	τ_S	D_I	τ_L	D_L
0.1	7.42×10^{-10}	6.35×10^{-13}	1.21×10^{-9}	4×10^{-12}	3.61×10^{-9}
0.3	7.95×10^{-10}	1.78×10^{-13}	5.63×10^{-10}	2×10^{-12}	3.50×10^{-9}
0.49	8.32×10^{-10}	1.54×10^{-14}	-	-	-
0.5	8.23×10^{-10}	1.02×10^{-14}	-	-	-
0.52	-	-	-	-	-

Conclusion

In conclusion, the D_S was shown to be constant with changes in ϕ, the τ_S decreased with increasing ϕ, D_I increased with increasing ϕ, τ_L decreased with increasing ϕ.

Diffusion and characteristic times when initialized in face center cubic arrangement

Aim

To determine the diffusion coefficients, the characteristic times and the phases when the simulation is initialized in a face center cubic arrangement to allow for a higher volume fraction simulation

Methodology

The particles were initialized in face centered close packed array within the center of cubic box of size L. The size of the box was determined based on the volume fraction. All particles were touching at the beginning of the initialization. The parameters used in the initialization of the simulation are summarized in Table 5.

Table 5: Initialized parameters

Parameter	**Value**
Number of atoms (N):	108
Number of steps	50,000
Atom diameter	$71 \times 10^{-12}\ m$
Volume fraction, ϕ	0.1, 0.3, 0.49, 0.52, 0.6 0.74
Temperature (used to initialize velocities from a maxwell-boltzman distribution)	20 K
Recorded Time step (the actual time step is based off the number of collisions)	$1 \times 10^{-14} s$
Particle array	Face Centered Cubic

Unfortunately the same technique used to calculate the $\mathrm{D_S}$ and τ_S for the simple cubic initialization cannot be performed on the face center cubic initialization. This is because the simple cubic initialization was designed to have even spacing between the particles based on the box size. The face center cubic initialization was set up with all with all particles touching. The MSD was used to calculate the intermediate diffusion, $\mathrm{D_I}$ and the long characteristic time, τ_L. The $\mathrm{C_v}$ was used to determine the long time diffusion $\mathrm{D_L}$. $\mathrm{g(r)}$ was used to determine the volume fraction at which the system began to solidify.

Results and discussion

It is worth noting that the simulation fails when the volume fraction is set to 0.74. This is because the reflective boundary is pressing against the particles which are in face center cubic arrangement. When the code tries to compute the next collision time, it finds that each particle is touching each other. Since the code can only a two-body collision (Enskog theory) at any one time, it gets caught in a loop where it continually tries to compute the next collision time, which is 0 as the atoms already colliding. Although the simulation could not be run, it could be visualized to see that the particles were sitting on the correct lattice sites.

Figures 24, 25, 26, 27 and 28 show the MSD for each volume fraction. It can be clearly seen that as the volume fraction increases, the τ_L is decreasing. This too is expected.

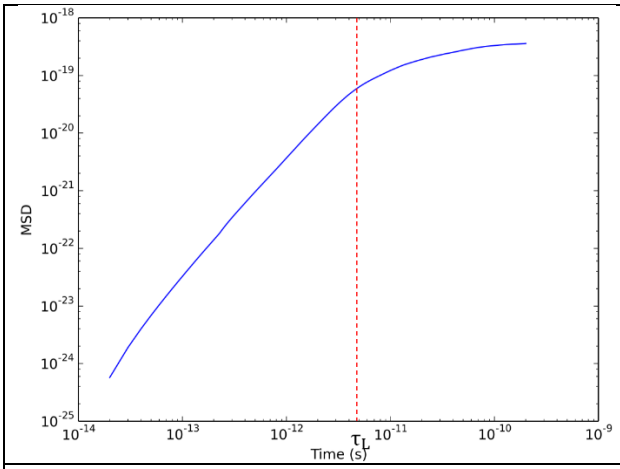


Figure 24: MSD versus time for $\phi = 0.1$ showing the long time characteristic time

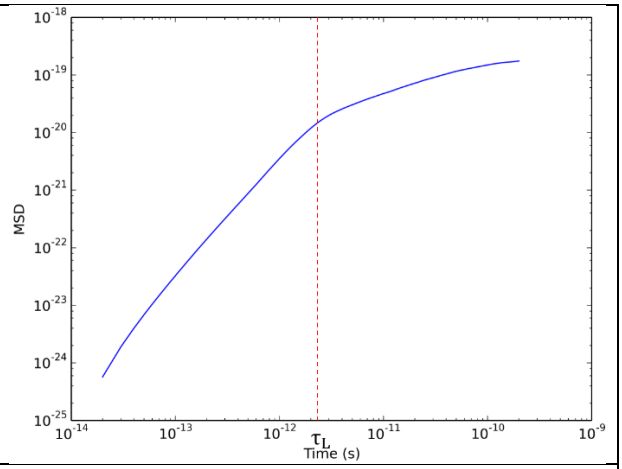


Figure 25: MSD versus time for $\phi = 0.3$ showing the long time characteristic time

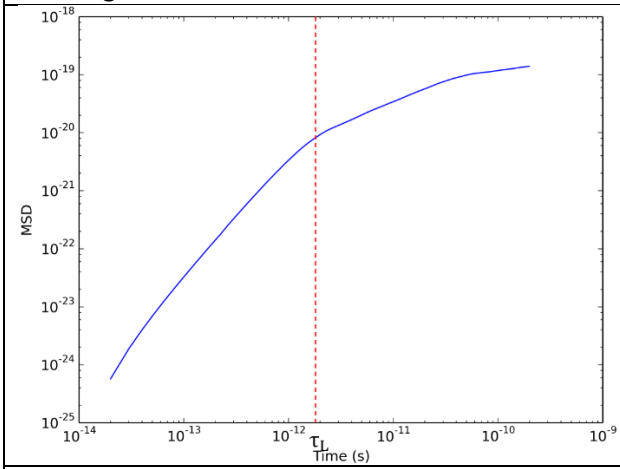


Figure 26: MSD versus time for $\phi = 0.49$ showing the long time characteristic time

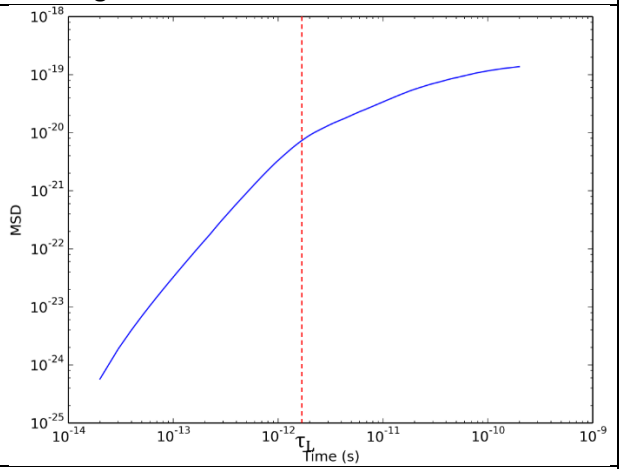


Figure 27: MSD versus time for $\phi = 0.52$ showing the long time characteristic time

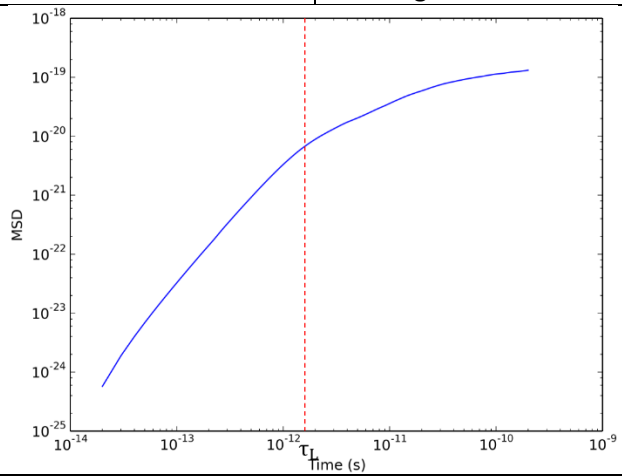


Figure 28: MSD versus time for $\phi = 0.6$ showing the long time characteristic time

From the MSD, the D_I has been calculated, and is summarized in . Here, it can be seen that the D_I (the gradient) decreases with increasing volume fraction. This again is expected as there are now more particles to impede the motion of particles, resulting in a reduction in the D_I.

Figures 29, 30, 31, 32 and 33 show a plot of C_V versus time for a series of volume fractions. It is difficult to see any key relations from the C_V plots other than the fact that the system is randomizing on the observed time scale. This is significant because it signifies that particles escaping the "cage" (the intermediate regime) and achieving their D_L.

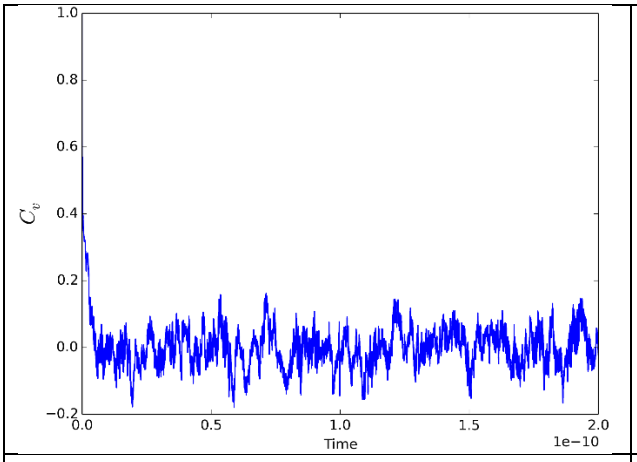


Figure 29: Velocity-Velocity autocorrelation function for $\phi = 0.1$

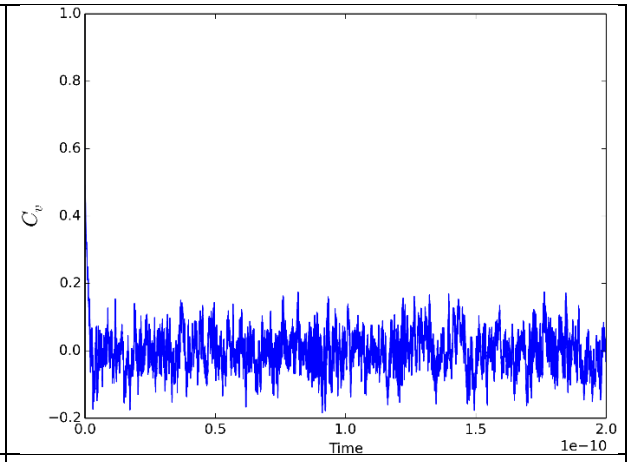


Figure 30: Velocity-Velocity autocorrelation function for $\phi = 0.3$

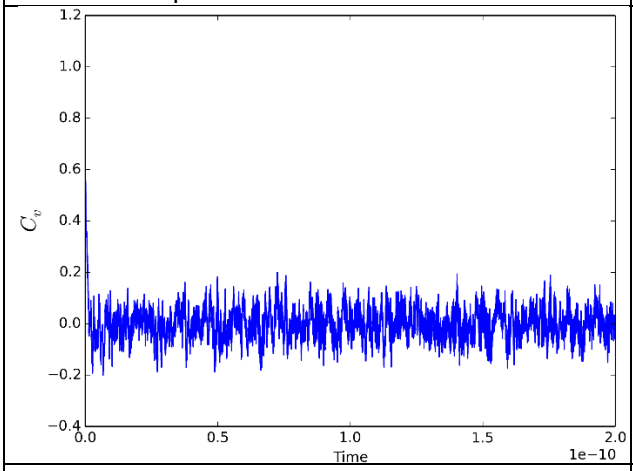


Figure 31: Velocity-Velocity autocorrelation function for $\phi = 0.49$

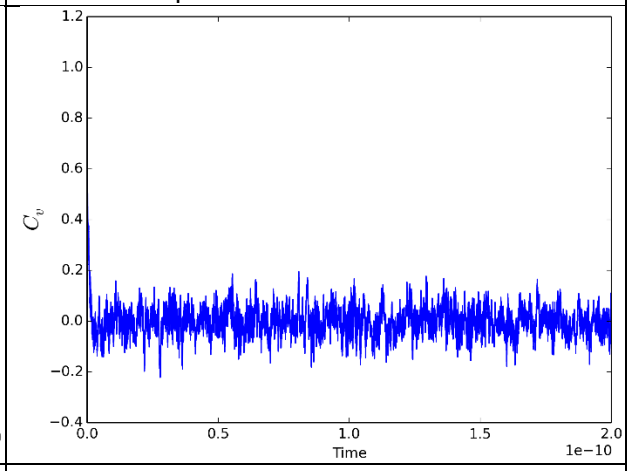


Figure 32: Velocity-Velocity autocorrelation function for $\phi = 0.52$

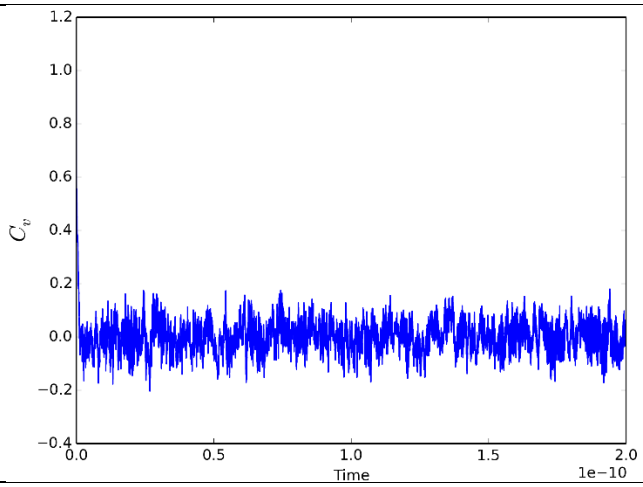


Figure 33: Velocity-Velocity autocorrelation function for $\phi = 0.6$

D_L has be calculated from the Velocity-Velocity autocorrelation function and is summarized in Table 6.

Table 6: Summary of all diffusion and characteristic times determined for the face center cubic initialized simulation.

ϕ	D_s	τ_S	D_I	τ_L	D_L
0.1	-	-	1.69×10^{-9}	3×10^{-12}	1.52×10^{-9}
0.3	-	-	8.54×10^{-10}	1.5×10^{-12}	1.51×10^{-9}
0.49	-	-	5.81×10^{-10}	1.2×10^{-12}	1.27×10^{-9}
0.52	-	-	5.80×10^{-10}	1.1×10^{-12}	3.24×10^{-9}
0.6	-	-	5.75×10^{-10}	1.0×10^{-12}	1.43×10^{-9}
0.74	-	-	-	-	-

Issues

The only issue that was encountered when implementing the code was that particles would overlap when the volume fraction was increased above 0.3. This issue has been reported in literature and is supposedly very difficult to remove. The only real way to remove this error is to increase the precision of the machine or increase the accuracy in the equations used to solve the physics of the hard sphere model.

Conclusions

The diffusion constants over different time regimes and the characteristic times describing these regimes were able to be simulated using a MD hard sphere model.

References

[1] D. F. a. B. Smit, Understanding Molecular Simulation, Academic Press, 2001.

[2] Princeton University, "Computer Science 226 Algorithms and Data Structure," 2015. [Online]. Available: http://introcs.cs.princeton.edu/java/assignments/collisions.html.

[3] N. V. Shokhirev, "Hard-sphere dybamics," 2009. [Online]. Available: http://www.shokhirev.com/nikolai/abc/physics/hard_spheres.html.

[4] D. Kofke, "Molecular Simulations," 2014. [Online]. Available: :www.eng.buffalo.edu/~kofke/ce530/Text/HardSphereMD_2.doc molecular dynamics hard sphere.

[5] R. G. Larson, The Structure and Rheology of Complex Fluids, Oxford University Press, 2007.

[6] DEMOCRITUS, "Mean Squared Displacement," 2007. [Online]. Available: http://www.ccp5.ac.uk/DL_POLY/Democritus/Theory/msd.html.

[7] University of Vienna, "Autocorrelation Functions," 2008. [Online]. Available: http://homepage.univie.ac.at/franz.vesely/simsp/dx/node23.html.

[8] J. Dhont, An Introduction to Dynamics of Colloids, Elsevier, 1996.

[9] B. J. A. A. T. E. WAINWRIGHT, "Phase Transition for a Hard Sphere System," *The Journal of Chemical Physics,* vol. 27, p. 1208, 1957.

[10] P. C. W. R. W. M. J. Z. R. R. R. O. Z. Cheng, "Phase diagram of hard spheres," *Materials and Design,* vol. 22, no. 7, p. 529, 2001.

[11] M. Micoulaut, "Calculating Properties of Space Correlation functions," 2011. [Online]. Available: http://www.lehigh.edu/imi/teched/AtModel/Lecture_5_Micoulaut_Atomistics_Glass_Course.pdf.