Nucleation and Crystallization of the Metastable Hard Sphere Fluid

by Wilkin Wöhler

Physics - Master Thesis

at the Albert-Ludwigs University of Freiburg

May 2021

Elaborated within the
Research group for complex systems and soft matter
supervised by
Prof. Dr. Tanja Schilling

Abstract

Nucleation and cluster development in the metastable hard sphere fluid are studied in this thesis. To this purpose an event driven molecular dynamcis simulation code is written and thouroughly tested by measuring well known quantities like diffusion coefficients or radial distribution functions at various packing fractions. Its performance is well suited for systems of about one million particles enabling the measurement of cluster growth rates and shape descriptors for clusters with sizes up to a hundred thousand particles without significant spatial influence of the cluster on to itself due to the periodic boundary conditions.

During the cluster growth a constant attachment rate to the cluster surface is measured, suprisingly unaffected by the packing fraction of the surrounding mestastable liquid. But the attachemnt rate may vary between single clusters by about 50% leading to uncertainties that do not exclude a dependence on the diffusion time.

For the shape descriptors based on the Tensor of Gyrationa tendency towards more spherical clusters is observed up to sizes of about a thousand particles. Clusters including more particles seem to conserve their almost spherical proportions and approach the complelty spherical shape only slowly.

Also the nucleation rate at volume fractions of $\eta \in [53.1\%,53.4\%]$ are measured at high precision compared to earlier measurements of these, but also confirming the discrepancy between real world and simulation experiments. Beyond that the memory kernels of nucleation for smaller systems are investigated finding a rather featureless Gaussian kernel. The width of the Gaussian kernel therby is comparable to the width of the phase transition time for a single trajectory.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Hard sphere system

The hard sphere system is the simplest model of a fluid, going beyond the ideal gas only by including interactions between the particles in the form of an occupied volume. Its well known potential between particles i and j is given in eq. 1.1.1.

$$V(r_{ij}) = \begin{cases} \infty & r_{ij} \leq \sigma \\ 0 & r_{ij} > \sigma \end{cases} \tag{1.1.1}$$

In this equation $r_{ij} = r_j - r_i$ denotes the distance between two particles and σ is the diameter of a hard sphere.

While the ideal gas model without pair interactions already makes it possible to derive the famous equation of state $pV = NkT$, it does not include phase transitions yet. These can be observed when granting the particles space to occupy, in the simplest case by defining hard spheres of the kind in eq. 1.1.1. As it is the simplest model and it is efficiently accessible for computer simulations the hard sphere system is very well suited to study basic properties of first order phase transitions.

Compared to experiments where similar systems are realizable and extensively studied, general properties of the system at hand can be varied effortlessly and information about each single particle can be extracted easily as they are naturally required for the simulation.

On the downside computer simulations are much more constraint in their size, but with today's computational possibilities systems of the order of one million particles become tractable, and hence computer simulations are becoming an ever more powerful tool to study phase transitions of simple systems.

The first of such simulations date back to the beginning of electronic computer technology with first studies by Alder and Wainwright in 1959 [1]. Since then more algorithms to increase efficiency have been elaborated, and technology advanced to the point where virtual studies of large scale systems are possible.

1.2 Memory effects and nucleation rates

Nucleation by itself can be characterized as a metastable state that, by crossing a first order phase transition, ends in a stable and qualitatively different state. Because nucleation processes are found in many circumstances, like atmosphere physics or metallurgy, people from various subjects have worked on understanding it.

Most descriptions are based on classical nucleation theory (CNT) which in a simple form is shown in section 1.4. CNT is capable of qualitatively capturing the behaviour of nucleations, but often fails a quantitative comparison to experiments or numerical findings, sometimes by orders of magnitude. Models of this kind often include modifications to circumvent field specific problems but no broadly applicable framework has found a consensus to fully describe nucleations today[2].

include Markovian embedding?

There are other theoretical works beyond the classical nucleation theory that not only tailor CNT to a specific problem but actually are based on more fundamental ideas. These take into account memory effects and non stationarity, where the latter is obviously important for phase transitions.

In the 1960's Mori and Zwanzig used their projection operator formalism to derive the Generalized Langevin equation while Grabert later also used a time dependent formalism introducing non stationarity. Based on these earlier works Meyer et al. derived the non stationary Generalized Langevin Equation (nsGLE)[3]. While the framework is too broad to cover at this point we may show the nsGLE in eq. 1.2.1 to understand the memory kernel that is evaluated in section 3.10.

$$\frac{dA_t}{dt} = \omega(t)A_t + \int_0^t K(\tau,t)A_\tau d\tau + \eta(0,t) \quad , \tag{1.2.1}$$

In the equation A_t denotes an observable depending on time on a single trajectory, $\omega(t)$ is the time dependent friction coefficient, $\eta(0,t)$ is a time dependent noise term and $K(\tau,t)$ is the memory kernel depending on two times. As can be seen the memory kernel is integrated over, which means that it holds the information about how the history of the observable's trajectory influences its future. As the kernel depends on two times this impact is time dependent. Further we may note that Markovian processes exhibit a Dirac delta distribution, as they do not include memory, in which case eq. 1.2.1 is reduced to the usual Langevin equation.

Quantifying the actual impact of memory effects in different systems is necessary for studying the use of the above mentioned ideas. For example Kuhnbold et al.[4] have previously studied the nucleation process of a metastable Lennard-Jones fluid concluding that memory effects can not be neglected for

an accurate description. One aim of this thesis therefore is to extend this picture by a study of memory effects in the nucleation of the metastable hard sphere fluid, done in section 3.10.

An other question concerning the hard sphere system is to measure nucleation rates which summarize by some definition how fast the phase transition occurs.
An other major aim is to help understand the huge discrepancy between nucleation rates of the hard sphere system measured in experiments on the one hand and in computer simulations on the other hand. To explain the difference spanning order of magnitude, multiple attempt have been made but it could not be resolved until now. To this purpose a detailed analysis and characterization of the hard sphere nucleation process is done, leading to a speculation on the origin of the discrepancy.

1.3 The phase diagram and the meta stable fluid

The equation of state for the monodisperse hard sphere system has various approximations [5]. The most common of these approximations due to its simplicity is the Carnahan-Starling approximation[6]

$$Z = \frac{1 + \eta + \eta^2 - \eta^3}{(1-\eta)^3} \ . \tag{1.3.1}$$

It approximates the compressibility factor Z as a function of the packing fraction η for the hard sphere fluid.

For the stable solid branch a common approximation is given by the Almarza equation of state[7]

$$\frac{p(v - v_0)}{k_B T} = 3 - 1.807846y + 11.56350y^2 + 141.6y^3 - 2609.26y^4 + 19328.09y^5 \ . \tag{1.3.2}$$

where p is the pressure, v is the volume per particle $v_0 = \sigma^3/\sqrt{2}$ is the volume per particle at close packing, including the diameter of the spheres σ and $y = p\sigma^3/(k_B T)$, with k_B being the Boltzmann constant and T the temperature of the crystal.
The inverse of the volume per particle corresponds to the number of particles per volume $v^{-1} = \rho$. The relation to the corresponding packing fraction η is given by $\rho = \frac{6}{\pi}\eta$, which can be easily shown by extending $\rho = \frac{N}{V}$ by the single particle's volume $V_s = \frac{4}{3}\pi\left(\frac{\sigma}{2}\right)^3 = \frac{\pi}{6}\sigma^3$.
Within the thesis mostly but not only the volume fraction is used as it is the most common parameter for describing the system, but it can always be interchanged by the density.

A first order phase transition occurs when switching between the two stable branches of the system, described by the two equations of state, in between volume fractions of $\eta_{freeze} = 0.494$ and $\eta_{melt} = 0.55$.

They correspond to first solidifying clusters when approaching the transition from the liquid branch and melting of the crystalline phase when approaching the transition from the solid branch. Within this volume fraction interval the systems tends towards a coexistence state that in its equilibrium only varies the fraction of solid to liquid volume.
This can be understood in the following way: The liquid may follow its branch to pressures above the coexistence pressure. As it becomes unstable the particles may arrange into the crystalline phase as each single particle can access a larger free volume in the structured lattice than it would be possible in the unordered fluid.
By comparing the volume fractions of random close packing $\eta_{RCP} \approx 64\%$ with the one of a face centered cubic or hexagonal close packing fraction of $\eta_{HCP} \approx 74\%$ this becomes evident. Within the crystalline phase each particle still has free volume accessible while the randomly packed particles are already confined at exactly one place.
This additional accessible volume translates into a larger number of possible states for the particle or in terms of thermodynamics a larger entropy, that acts as a driving force for the metastable fluid into the solid phase. As the particles in the crystal are packed more densely with a volume fraction of $\eta_{melt} = 0.55$, the pressure is reduced and not all fluid transforms into the solid phase, but both phases may coexist.
The overall phase diagram is shown with the coexistence pressure in fig. 1.3.1.

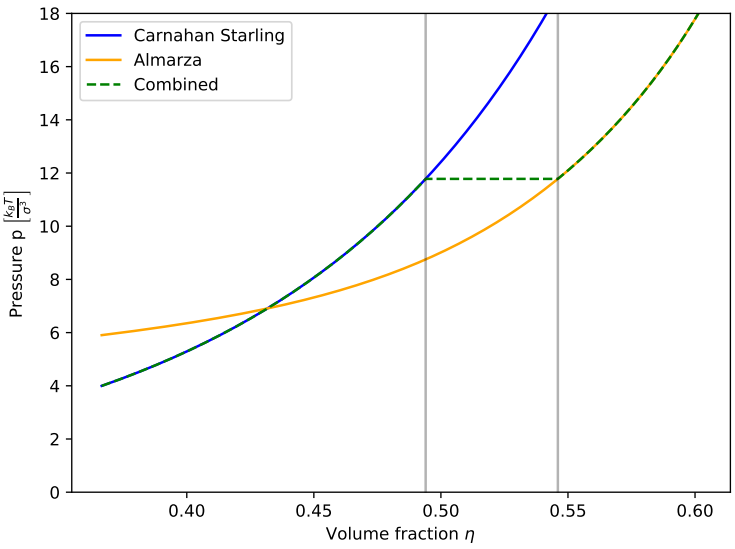


Figure 1.3.1: Phase diagram of the hard sphere system with freezing and melting volume fraction shown as shaded lines and the green dashed line indicating the equilibrium stable branch. Where liquid and solid branch do not coincide with the stable branch these are unstable and tend towards the stable branch.

The solid fraction in terms of volume for the system $x_s = \frac{V_s}{V}$ with V_s the solid volume and V the total volume can be described within the coexistence region in the equilibrium state by **??**.
For the derivation it is necessary to use that in the stationary state the density of the solid phase is given by the melting density and that the liquid density is equal to the freezing density, i.e $\rho_s = \rho_{\text{melt}}$ and $\rho_l = \rho_{\text{freeze}}$ respectively.
When further using the trivial equations

$$\begin{aligned} V &= V_s + V_l \ , \\ N &= n_s + n_l \ , \\ N_i &= \rho_i V_i \ , \end{aligned} \tag{1.3.3}$$

with $n_{s/l}$ the number of solid/liquid particles we may write

$$\rho V = \rho_s V_s + \rho_l V_l \tag{1.3.4}$$

leading under the assumption of equilibrium within a few lines of calculation to

$$\frac{V_s}{V} = \frac{\rho - \rho_{\text{freeze}}}{\rho_{\text{melt}} - \rho_{\text{freeze}}} \ . \tag{1.3.5}$$

As the solid fraction below ρ_{freeze} vanishes and above ρ_{melt} is 1, we can conclude that the equilibrium solid fraction of the system is given by eq. 1.3.6.

$$x_s(\rho) = \begin{cases} 0 & \rho < \rho_{\text{freeze}} \\ \frac{\rho - \rho_{\text{freeze}}}{\rho_{\text{melt}} - \rho_{\text{freeze}}} & \rho_{\text{freeze}} < \rho < \rho_{\text{melt}} \\ 1 & \rho > \rho_{\text{melt}} \end{cases} \ . \tag{1.3.6}$$

Evaluating the above result at feasible volume fractions for nucleation in between $\eta \in [0.53,0.55]$ leads to coexistence fractions of $x_s \in [0.7,1]$. This means that we are expecting nucleated systems to consist mostly of the solid phase after enough time for complete crystallization.

As pointed out earlier the phase transition takes place as it reduces the pressure in the liquid. This means that already during the growth of clusters the volume fraction of the metastable liquid is reduced, potentially altering its behaviour significantly. For closer inspection of this the particle density of the metastable liquid depending on the solid fraction x_s is evaluated in eq. 1.3.9. For this purpose first the liquid volume V_l and the number of liquid particles N_l are expressed in terms of the

solid fraction x_s:

$$V_l(x_s) = V(1 - x_s) \tag{1.3.7}$$

$$N_l(x_s) = N - n_s(x_s) = N - \rho_m V x_s = N(1 - \frac{\rho_m}{\rho} x_s) \tag{1.3.8}$$

Combining eq. 1.3.7 and eq. 1.3.8 to the expression for the particle density in the remaining liquid leads to

$$\rho_l(x_s) = \frac{N_l(x_s)}{V_l(x_s)} = \frac{N}{V} \frac{1 - \frac{\rho_m}{\rho} x_s}{1 - x_s} = \rho \frac{1 - \frac{\rho_m}{\rho} x_s}{1 - x_s} \tag{1.3.9}$$

Some examples of eq. 1.3.9 are depicted in fig. 1.3.2 for moderate solid fractions of the system at regular volume fractions used for nucleation.

What can be seen is that for crystalline fractions of a few percent the remaining liquid is not altered

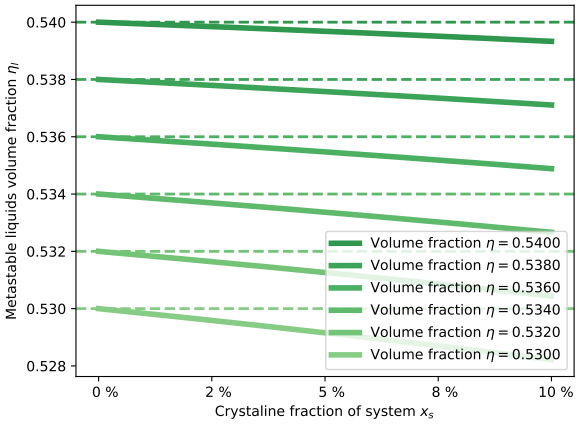


Figure 1.3.2: Visualization of eq. 1.3.9. The volume fraction of the remaining liquid decreases for all shown initial volume fractions only little up to crystalline ratios of a few percent.

significantly. Especially for system sizes of about 1 million particles it already corresponds to cluster sizes of a few ten thousand particles, where stable growth of clusters takes place which is rather insensitive to changes of the volume fraction as shown in section 3.5. This means that during the highly sensitive cluster forming processes the volume fraction of the liquid can be assumed to be globally stable.

1.4 Classical nucleation theory

Classical nucleation theory (CNT) has been proposed by Becker and Döring in 1935[8] and since then used and modified multiple times to suit various types of systems. Modifications of it are necessary to account for deviations off experimental results. Still it provides some reference or expectation to compare with the simulation data. Still its framework seems not to encompass the full process.

The simplest version of CNT assumes that a spherical crystallite may form in the liquid with properties of the bulk crystal while the fluid remains with the properties of the bulk liquid. The difference in the free energy landscape is given by a surface and a volume term. The first arises from the surface tension γ between the fluid and the solid bulk phase. The latter is caused by the difference in chemical potential $\Delta\mu$. The whole expression for the free energy is given by

$$\beta\Delta G(R) = 4\pi R\gamma - \frac{4}{3}\pi R^3\rho\Delta\mu \,, \tag{1.4.1}$$

where ρ is the particle density of the solid phase and further R is the radius if the crystallite.

For the difference of the chemical potential $\Delta\mu$ we first derive the free energy difference between the metastable liquid branch and the stable coexistence branch. To calculate the free energy we employ its differential relation

$$dF = -S\,dT - P\,dV + \mu\,dN \tag{1.4.2}$$

at a constant number of particles and constant temperature. By reformulating dV using $dN = dV\,\rho + V\,d\rho$ and $dN = 0$ we find $dV = -d\rho\frac{N}{\rho^2}$. Under this transformation eq. 1.4.2 becomes

$$\frac{dF}{N} = \frac{P(\rho)}{\rho^2}d\rho \,. \tag{1.4.3}$$

The pressure $P(\rho)$ is given by the equation of state and approximated by the Carnahan-Starling approximation where $\eta = \frac{6\rho}{\pi}$ and $Z = \frac{pV}{NkT} = \frac{p(\rho)}{\rho kT}$. Integrating eq. 1.4.3 between two densities $\rho_{1/2}$ leads to

$$\frac{\Delta F}{N} = \int_{\rho_1}^{\rho^2} \frac{kT}{\rho}\frac{1 + \left(\frac{6\rho}{\pi}\right) + \left(\frac{6\rho}{\pi}\right)^2 - \left(\frac{6\rho}{\pi}\right)^3}{\left(1 - \frac{6\rho}{\pi}\right)^3}d\rho \,, \tag{1.4.4}$$

with the analytical solution

$$\int_{x_1}^{x_2} \frac{1 + (ax) + (ax)^2 - (ax)^3}{(1-ax)^3\,x}dx = \frac{3 - 2ax}{(ax-1)^2} + \log(x)\Big|_{x=x_1}^{x_2} \,. \tag{1.4.5}$$

Dropping the lengthy notation for $\eta = \left(\frac{6\rho}{\pi}\right)$ we end up with

$$\frac{\Delta F}{N} = kT\left(\frac{3-2\eta_2}{(\eta_2-1)^2} - \frac{3-2\eta_1}{(\eta_1-1)^2} + \log\left(\frac{\eta_2}{\eta_1}\right)\right) \tag{1.4.6}$$

The analytical solution is compared in fig. 1.4.1 with numerically found results which have been calculated before the analytical solution was found. The free energy difference is in the following identified with the difference in chemical potential $\Delta\mu$ as it is the driving force of the nucleation. is this justified like this?

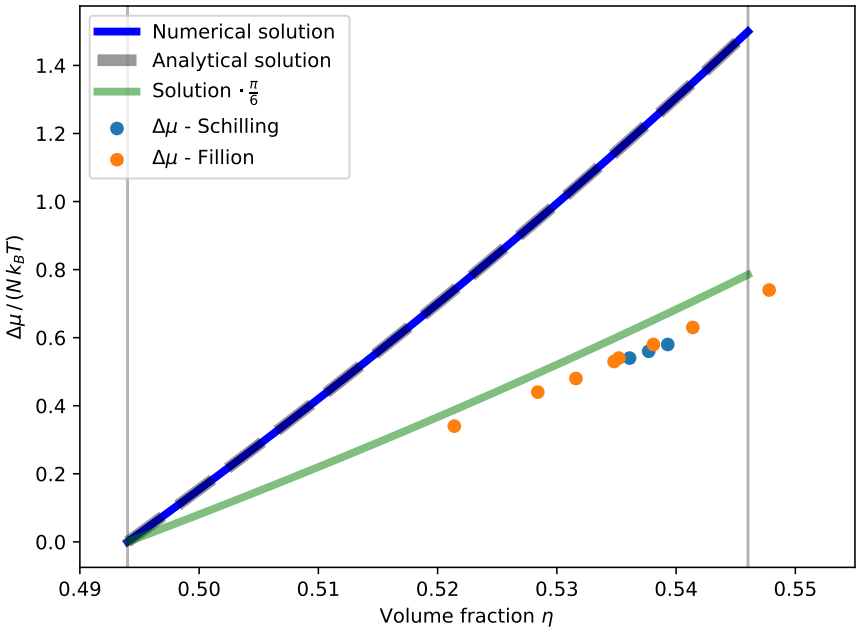


Figure 1.4.1: Free energy difference per particle between the metastable liquid phase and the coexistence phase. Values found in the literature deviate from the shown result, but we assume that a factor of $\frac{\pi}{6}$ in the calculations is missing in either this or their calculation, as the modified green curve collapses rather accurately on the literature values when choosing $\eta_{freeze} = 0.5$.

Coming back to the free energy landscape of eq. 1.4.1 we see that it exhibits a maximum at a radius called R_{crit}. The interpretation of this radius is that if a cluster surpasses the critical radius it is likely to continue to grow until the system settles at the equilibrium solid fraction. Cluster in this sense is defined as a structure having a crystalline like ordering locally. The critical radius, simply calculated by setting the derivative of eq. 1.4.1 to zero, is given by eq. 1.4.7.

$$R_{crit} = \frac{2\gamma}{\rho\Delta\mu} \tag{1.4.7}$$

Furthermore the height of the barrier can be calculated to be

$$\beta\Delta G(R_{crit}) = \frac{16\pi\gamma^3}{3\rho^2(\Delta\mu)^2} \ . \tag{1.4.8}$$

The classical critical radius depending on the volume fraction is depicted in fig. 1.4.2 for a first impression of the cluster sizes that we are expecting for nucleation. The interfacial surface tension is often given by $\gamma \approx 0.6 k_B T \sigma^{-2}$ but it's precise value is under debate. Thus we may stick to one of the recently calculated values of $\gamma = 0.589 k_B T \sigma^{-2}$[9].

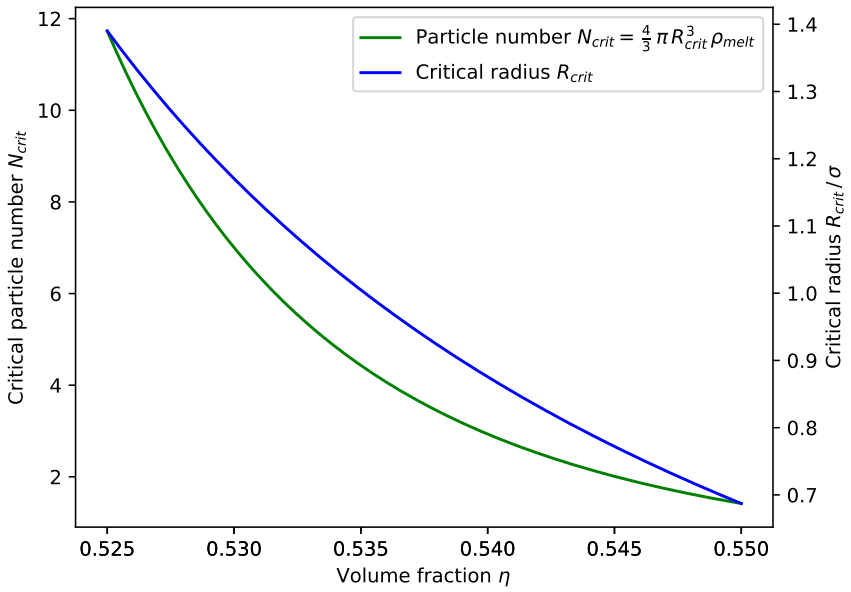


Figure 1.4.2: Critical radius R_{crit} calculated from CNT depending on volume fraction η. As it can be seen the critical radii are rather small. When using the chemical potential calculated by Fillion and Schilling the critical clusters sizes become of the order $N \approx 50$ at intermediate metastable volume fractions, which is much more in agreement with typical largest cluster fluctuations found in simulations.

1.5 Computer Precision

The finite floating point precision of computers impacts the outcome of a single simulations as the simulation itself constitutes a many body problem with chaotic behaviour. In this section it is shown that even smallest variations of positions for example, lead to radical changes of the simulation after a certain number of steps. It is used to emphasize the importance to rigorously save the simulation state if it is supposed to be restarted from file, or with changing measurement intervals.

The exponential growth of induced variations in a chaotic system can be visualized by comparing a reference simulation with a pertubred one. In fig. 1.5.1 the mean of the squared displacements of all particles is observed between such a pair of simulations. The perturbation consists of a slight push of $10^{-10}\sigma$ to one particle's position, comparable with missing some floating point precision during saveing and loading.

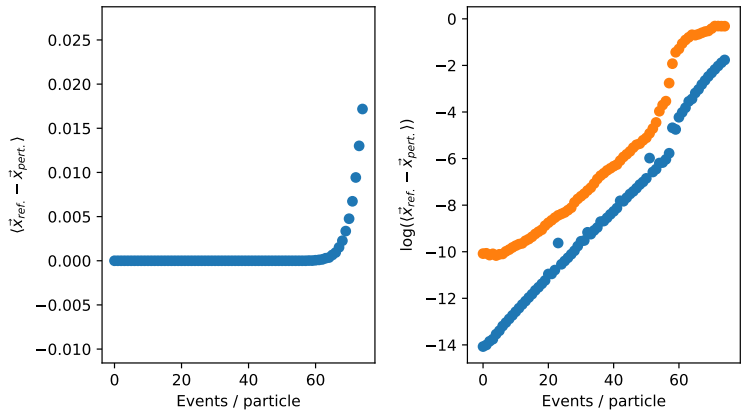


Figure 1.5.1: Mean difference of particle positions in the reference and perturbed simulation. The blue lines show the same data while the orange curve shows the maximum deviation present at each step. For comparison with datasets using the system time δt as units, a rough conversion is given by $T \approx (\#\text{steps}) \cdot \frac{1\delta t}{60\text{steps}}$ where a step is defined as one event per particle.
The maximum deviation first consists only of the initial perturbation but then increases similar like the mean deviation.

Only observing the left side leads to the assumption that the simulations remain the same to a certain point and then suddenly diverge. But when looking at the logarithmic representation we see that actually the perturbation grows exponentially as long as it is small and deviates from this exponential growth at some point when reference and perturbed simulation become more or less independent of each other.
The small bumps at first sight seemed to be an artifact of the periodic boundary conditions but this seems not to be the case. What causes these deviations therefore remained hidden.

The challenge that this behaviour poses is that any perturbation leads to a completely different simulation. In the context of EDMD simulations we can for example look at the case when a measurement of some quantity is performed. For this purpose all particles have to be propagated to the global time. To not perturb the system with this extra calculations, an exact copy of the particle positions has to be saved prior to the propagation. After the measurement this copy is then used to restore the

unperturbed system.
Similarly recalculating an event for the FEL at some point of time is not possible as the outcome will vary in the last floating point digits. For this reason it becomes necessary to save all precalculated events of the simulation to be able to restart it from a file.
Facing this challenge makes it possible to for example resimulate some interesting part of a trajectory from some saved checkpoint with a higher measurement frequency to resolve more details.

1.6 Comparison to real world experiments

Starting in 1986 with the experiments by Pusey and Megen [10] hard spheres have been synthesized in the lab. Today a large variety of systems is known to show hard sphere like behaviour, but still further systems are developed to better controll stability, sphere size or also to reduce the possible impact of charges on top of the spheres as the Coulomb interaction alters the behaviour of the system strongly. All of these system have in common, that the hard spheres are in a bath of a fluid, which surrounds them. Even though nucleation experiments have been done without gravity in space[11] usually the fluid's mass density has to be matched to the mass density of the hard spheres to prevent sedimentation. Further it is necessary for optical measurements to match the refractive index of the fluid and the spheres as otherwise it becomes opaque .

The absence of the bath in simple hard sphere simulations constitutes a large difference to the hard sphere systems in the laboratory. On the one hand it has been argued that it only introduces a difference of the diffusion time scale and is regularly compensated by normalizing all times with a characteristic diffusion time. On the other hand a discussion on the possibility of hydrodynamic effects changing the behaviour of the laboratory system compared to simulations is ongoing at the moment,citation of the pro and contra hydrodynamics? and also the mode spectrum of the suspending fluid within the cavities between the dispersed spheres might have a more important role than expected.
Even though it is desirable to include the suspending fluid into simulations, the proliferation of particles often is not feasible as calculation times increase by orders of magnitude.

A further difference is given by the spatial extent and geometry of the simulation. The geometry is often defined by periodic boundary conditions (PBC) in simulations to circumvent surface effects, but it is a rather unphysical setup.
Concerning the spatial extent, simulations are mostly confined to very small systems in comparison to experimental setups leading to a further difference between the measurement geometries. While the experimentalists usually probe a continuous volume of hard spheres in a suspending fluid, in simulations many disjunct volumes are used as each subvolume can be processed by one CPU. The expected

behaviour of the disjunct volumes under the assumption of a constant nucleation rate is discussed in section 3.8.1. On the other side when not looking at ratios of nucleated and not nucleated boxes, but rather a quantity describing the overall solid fraction of a volume in the thermodynamic limit we may expect a different behaviour that is inspected in the following.

As is shown in section 3.5 the cluster growth rate in simulations is more or less independent of the volume fraction. When making the assumption that this is the case not only in the small region in which it was tested, we may approximate the number of particles in a cluster N at time t as

$$N(t) = c^3(t - t_0)^3 \tag{1.6.1}$$

with t_0 the time where the cluster emerged from the fluid. Furthermore approximating the stochastic nucleation events by a constant rate at which new clusters are added to the system $\Delta t = (\kappa V)^{-1}$, we can write the total number of solidified particles N in a Volume V at time step m with the corresponding time $t_m = m\Delta t$ as the sum of all previously nucleated cluster sizes N_i. Reformulating this leads to

$$\begin{aligned}
N(t_m) &= \sum_{i=1}^{m} N_i && \overset{V\to\infty}{\Leftrightarrow} \quad N(t) = \kappa V \int_0^t c^3(t-t')^3 dt' \\
\Leftrightarrow N(t_m) &= \sum_{i=1}^{m} N_i \frac{\Delta t}{\Delta t} && \Leftrightarrow \quad \frac{N(t)}{V\rho_{\text{melt}}} = \frac{\kappa c^3}{\rho_{\text{melt}}} \frac{1}{4} t''^4 \Big|_{t''=0}^{t} \\
\Leftrightarrow N(t_m) &= \kappa V \sum_{i=1}^{m} N_i \Delta t && \Leftrightarrow \quad \frac{V_{solid}}{V} = t^4 \frac{\kappa c^3}{4\rho_{\text{melt}}} \\
\Leftrightarrow N(t_m) &= \kappa V \sum_{i=1}^{m} c^3(t_m - t_i)^3 \Delta t && \Leftrightarrow \quad x_s(t) = t^4 \frac{\kappa c^3}{4\rho_{\text{melt}}}
\end{aligned} \tag{1.6.2}$$

The solid fraction is not the equilibrium solid fraction, but rather the expected solid fraction of an infinitely large system at a time t after some quench that suddenly takes the system into the metastable regime.
For the derivation of eq. 1.6.2 the thermodynamic limit $V \to \infty$ is used to obtain the definition of an integral. Further it neglects any interference between different clusters. This assumption is justified for $x_s \ll 1$ if no long range interference are present and heterogeneous nucleation is assumed to be part of the cluster growth process.

If the aforementioned assumptions also hold we can calculate a characteristic nucleation time t^* at which x_s is not negligible anymore. As in simulations with periodic boundary conditions clusters begin to interfere with each other at a filling fraction of about $x_s = \frac{1}{8}$ this is also chosen as a threshold where interferences can not be neglect any longer in the macroscopic system. Under this definition t^*

becomes

$$t^* = \sqrt[4]{\frac{\rho_{\text{melt}}}{2\kappa c^3}} \tag{1.6.3}$$

As we see the time t^* actually depends only on the fourth root of the induction time $\tau_{nucleation} = \kappa^{-1}$. This might be an explanation for the huge discrepancy between experiment and simulation studies.

A first try is taken in fig. 1.6.1, where a nucleation time of the type defined in eq. 1.6.3 is calculated and plotted together with other nucleation rates. This can not be taken as a proof but as a hint that

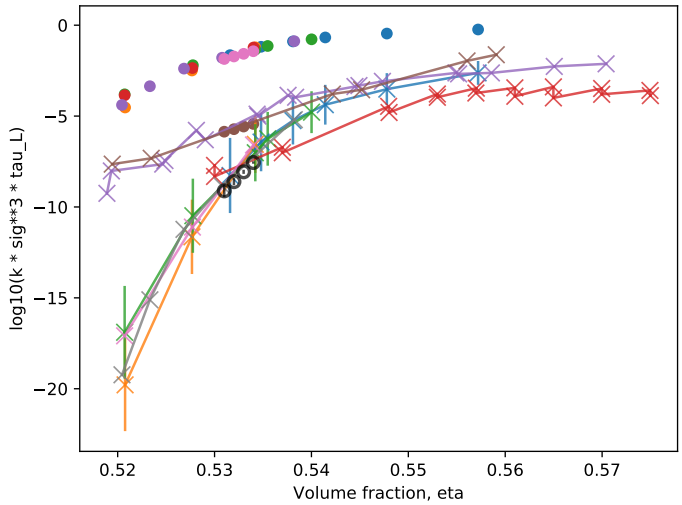


Figure 1.6.1: Diagram with modified nucleation rates calculated by eq. 1.6.3. The growth rate is set by the measurement shown in section 3.5. Furthermore a modified version with a factor 10000 is plotted to visualize the matching of the slopes. t_fill denotes t^* as it is the time until a significant fraction of volume is filled by clusters.

experimentalists might be measuring more cluster growth than nucleations. It has to be discussed with experimentalists if the assumptions leading to this result hold under close inspection or if measures against this behaviour have been taken like using many disjunct cavities.

2 Simulation details

During the course of the master thesis an event driven molecular dynamics (EDMD) simulation code has been developed. The EDMD approach is chosen because interest in the actual dynamics of the system are desired. This means that simulations probing the phase space of the system instead of the dynamics, like Monte Carlo (MC) simulation schemes, are not suited.
Furthermore the discontinuous potential of the hard spheres is an obstacle not easy to face in regular molecular dynamics (MD) schemes, where the Newtonian equation of motion for the particles are numerically integrated. As the EDMD approach even requires these discontinuities, as discussed in this section, it is very well suited for the purpose at hand.
Further key points of the program, possible extensions and a thorough testing are presented.

2.1 Algorithm and Simulation details

In this section we will highlight the main differences to regular MD simulations, as they are a main tool to probe the dynamics of molecular systems. Furthermore we will stick to the hard sphere example when discussing the EDMD simulations, but it can be kept in mind that the EDMD approach allows to simulate the dynamics of all systems governed only by potentials made of step functions.

The decisive difference between EDMD simulations and regular MD schemes is that, instead of evaluating all pair and external forces on each particle and then evolving the whole system to the next time step, EDMD simulations do not have a predefined time step, but the system is evolved from one event to the next one. An event in this context is defined as the time where the next collision in the whole system takes place.

The event prediction algorithm follows closely the approach proposed by Bannerman et al. [12] which is discussed in the next section.

2.1.1 Event driven molecular dynamics (EDMD)

For the prediction of events in EDMD simulations an overlap function $f_{ij}(t)$ between particles i and j is defined, where the squared quantities are used merely because they are easily accessible.

$$f_{ij}(t) \coloneqq |\vec{r}_j(t) - \vec{r}_i(t)|^2 - \sigma^2 \tag{2.1.1}$$

$$\left|\begin{array}{ll} \text{with} & \vec{r}_i(t) = \vec{r}_i(t_0) + (t - t_0)\ \vec{v}_i(t_0)\,, \\ & \Delta t \coloneqq t - t_0\,, \\ & \vec{v}_{ij}(t) \coloneqq \vec{v}_j(t) - \vec{v}_i(t)\,, \\ & \vec{r}_{ij}(t) \coloneqq \vec{r}_j(t) - \vec{r}_i(t)\,, \\ \Leftrightarrow & \vec{r}_{ij}(t) = \vec{r}_{ij}(t_0) + \Delta t\ \vec{v}_{ij}(t_0) \end{array}\right. \tag{2.1.2}$$

$$f(t) = (\vec{r}_{ij}(t_0) + \Delta t\ \vec{v}_{ij}(t_0))^2 - \sigma^2 \tag{2.1.3}$$

$$f(t) = |\vec{r}_{ij}(t_0)|^2 + \Delta t^2\ |\vec{v}_{ij}(t_0)|^2 - 2\Delta t\ \vec{r}_{ij}(t_0) \cdot \vec{v}_{ij}(t_0) - \sigma^2 \tag{2.1.4}$$

The overlap function has the property that it is negative for two particles being closer than their diameter, 0 at collision and positive if neither overlapping nor touching. The task to calculate the next collision thus is to calculate the roots of eq. 2.1.4.

Solving for Δt with $rr \coloneqq |\vec{r}_{ij}(t_0)|^2$, $vv \coloneqq |\vec{v}_{ij}(t_0)|^2$ and $rv \coloneqq \vec{r}_{ij}(t_0) \cdot \vec{v}_{ij}(t_0)$ has the solution eq. 2.1.7.

$$0 = rr + vv\ \Delta t^2 - 2rv\ \Delta t - \sigma^2 \tag{2.1.5}$$

$$\Leftrightarrow \quad 0 = \Delta t^2 - \frac{2rv}{vv}\ \Delta t + \frac{rr - \sigma^2}{vv} \tag{2.1.6}$$

$$\Leftrightarrow\ \Delta t = -\frac{rv}{vv} \pm \sqrt{\left(\frac{rv}{vv}\right)^2 - \frac{rr - \sigma^2}{vv}} \tag{2.1.7}$$

But a caveat when executing on a floating point machine is present as can be seen when considering which solution is of interest. For a possible collision it is necessary that the two particles move towards each other or mathematically $rv < 0$ as the relative velocity is required to be opposite to the relative position.

Further the quadratic formula has two solutions, corresponding to the beginning and the ending of the overlap. Because the entry is prior to the exit, we further conclude that we are interested in the smaller solution, that is:

$$\Delta t = \frac{-rv - \sqrt{(rv)^2 - vv(rr - \sigma^2)}}{vv} \tag{2.1.8}$$

Now for the case where the distance of the spheres is already close to the diameter of the spheres we find $(rv)^2 \gg (rr - \sigma^2)$, which results in a cancellation of two large numbers leaving a small number. Floating point number operations are inherently badly suited because they tend to large inaccuracy in this case. But rewriting eq. 2.1.8 by making use of the third binomial formula leads to the mathematically identical expression

$$\Delta t = \frac{(rr - \sigma^2)}{-rv + \sqrt{(rv)^2 - vv(rr - \sigma^2)}} \; . \tag{2.1.9}$$

Comparably eq. 2.1.9 does not contain a cancellation of the type seen before and hence is better suited for the use in a computer simulation [13].

The event prediction algorithm proposed by Bannerman[12] works by differentiating 4 cases:

1. If $rv > 0$ the particles move away from each other leading to a collision time of $\Delta t = \infty$.

2. If $rr < \sigma^2$ an overlap is present resulting in an immediate collision time of $\Delta t = 0$

3. If $(rv)^2 - vv(rr - \sigma^2) \leq 0$ the two particles miss each other, including touching without momentum transfer, resulting in a collision time of $\Delta t = \infty$

4. If none of the before is given the particles collide and Δt is calculated by eq. 2.1.9.

All collision times for a particle are then stored in a queue that is sorted by the event time and is called particle event list (PEL). From the PEL the first entry is then passed to the global future event list (FEL).
This procedure initially takes place for all particles to set up the system and later on only for those particles involved in the execution of an event.

In section 2.1.2 the implementation of some widely used measures to reduce redundant calculations and using a cell system to reach $\mathcal{O}(N)$ computation time are discussed.

A further detail to take care of is the possibility of scheduled events which have become invalid due to an earlier collision of one of the particles. This is handled by assigning an interaction count to each particle that is stored at precalculation time with the event. When the event is drawn from the FEL and the interaction count of one of the particles has increased in the meantime, the event is said to be invalidated. Depending on which particle had an event in the meantime the invalidation either causes no action or a recalculation of new events.

2.1.2 Details of the Implementation

As the simulation code is based on an earlier Monte Carlo code for hard spheres a complete walk through the whole program would become quite extensive. Hence we will focus on key points to understand the details of the simulation.

Event struct

We start with the basic *Event* struct which includes 6 entries as shown in tab. 2.1.1. The type of

Datatype	Name of entry
(timeType)	time
(int)	event_type
(Particle*)	particle
(void*)	partner
(int)	particle_count
(int)	partner_count

Table 2.1.1: Content of the *Event* struct.

time (timeType) is usually set to double and actually is more or less static as it depends on the h5md routines that have been written for the double type. The *time* variable holds the time for when the event is scheduled.
The *event_type* variable is either set to 0 or 1 and indicates if the event is a cell transfer or a collision of two particles, respectively. To include hard walls or other elements further types of events can be defined.
The *particle* variable is a pointer to the particle for which the event has been precalculated, while the *partner* variable is set to be a void pointer, allowing it to either be interpreted as a particle pointer for the collision type event or as an integer pointer to the index in the current cells' neighbours list for transfer events.
In the last two rows the interaction counts for particle and partner are listed as well. As the destination cell in a transfer event does not require an interaction count, the *partner_count* variable is only used for collision events.

The *event* struct is used for all events throughout the simulation. For read and write operations with the HDF5 file format, the struct *event_data* is available which uses only indexes instead of pointers.

Particle class

The *Particle* class is comparable to the one from the MC code basis. Its MC related variables have been removed and additional key variables and concepts will be discussed in the following.

First a vector storing events called *backupEvents* has been added to make it possible to store events from the precalculation for the case of the first event being invalidated. The idea of reusing events is discussed in many publications, for example that the memory cost increases only moderately with more backup events while the speedup does not increase much for more than two stored events [14]. It also has been argued that the added complexity can not account for the increase in efficiency[15]. Even though in the own simulations a decrease in calculation time of more than 10% was observed and the cost of complexity and memory was seen as moderate. The difference might be explained by the fact that the systems under consideration in this thesis have a rather large particle density, leading to more invalid collisions.

In the context of reusing precalculated results, it should also be mentioned that after a cell transfer the recalculation of events can be reduced to possible partner particles of only the new neighbouring cells, leading to only 1/3 of the calculation time in this case. But as mentioned systems under consideration are very dense resulting in little transfer events, often constituting below 5% of all executed events. Thus the increase in efficiency was assumed to be to costly on the complexity side, and not implemented. But for sparse systems, it might make sense to include an *updatePEL()* routine.

Also key differences to the former MC Particle type are the variables *total_interactions* and *particle_delayed_time*. The first is the variable for book keeping of interactions, while the second represents the event driven character of the simulation. Because each particle only moves on purely ballistic trajectories until an event occurs, it is not necessary to keep all particle positions and velocities synchronized in time. Quite on the contrary it would mean executing extra operations with extra calculation time and extra rounding errors.

But as it sometimes is desired to have the whole configuration at one point of time, the *transferToTime()* function of the particle provides the possibility to move the particle to some point of time. This is necessary soon as measurements are performed on the system, including snapshots.

As mentioned before the system behaves chaotic even under slightest changes like a rounding error from an extra floating point operation. A result of this is that measuring at different rates during a simulation changes the simulation trajectory quite a bit. It is observed in section 1.5 that such a system may keep close to the undisturbed trajectory for about 50-100 events/particle. As it is of desire to measure quantities and take snapshots without disturbing the simulation, the program employs

copies of the configuration being costly in terms of memory but making simulation resets or higher sampling rates at interesting points possible within a well defined trajectory.

The measurement without perturbing the system is implemented by making a backup copy of the working configuration just before a measurement is taken. The working trajectory then is disturbed by the measurement, and afterwards replaced with its state before the measurement from the backup configuration.
A second copy is carried throughout the simulation including the full simulation state, while the first only includes the particle configuration. To save a state during the simulation and reset to just the same point at any later time might be useful for example to do a committer analysis, where a cluster at different stages is sampled multiple times with different perturbations.

The *Box* class

The box of the simulation stayed mostly the same as in the previous MC code. A new element is the *neighbours_lookup* table. It contains the indices for the cells' *neighbours* array pointing to cells that share their surface. It is used to identify which cell a particle has to be transferred to during a cell transfer event.

Furthermore the *Update()* routine now takes care of all quantities depending on the length of the box, and the *rescale()* routine is a simple rescaling of the edge lengths with an additional *Update()* call.

The *Scheduler* class

While the aforementioned elements of the program are also required for the EDMD simulation, the *Scheduler* class certainly contains the most distinct parts of the program. It keeps track of all events to come, predicts new events and orchestrates the execution of the events. The essential functions are discussed in the following subsections while some basic properties are shortly highlighted here.

First of all the *Scheduler* holds the future event list (FEL) in which at least one event per particle is stored. As discussed within section 2.1.2 the simulation is capable of saving the complete state of a trajectory, including all precalculated events. For this purpose the *reset_FEL_array* is available.
Furthermore the *Scheduler* includes the *global_time* variable that holds the latest event execution time.
Important for the efficiency is the pre allocation of all arrays used during the predictions, as the number of executions for the collision prediction routine is about $\frac{30}{\text{particle}\cdot\text{step}}$ easily accounting to a few

billion function calls during a small simulation.

Scheduler::predictTransfer()

As the name suggests this function predicts the next cell transfer of a particle due to its movement. For this it calculates the position of the particle at global time, which for a valid state of the simulation always lies within its cell. By transforming the momentary position of the particle from the global coordinate system to the coordinate system of the cell and taking into account the periodic boundary conditions, we can write for each dimension i the equations

$$t_{i1} = -\frac{r_i}{v_i} \quad \text{and} \quad t_{i2} = \frac{l_i - r_i}{v_i} \tag{2.1.10}$$

which describe the times when the particle pierces the cell's left and right boundaries. A negative time corresponds in this case to a boundary crossing in the past, a time comparable to 0 means that the particle is on the edge of its cell and a positive time means that the boundary crossing lies in the future. By going through the different possible cases for t_1 and t_2 we find the resulting next crossing time for each case as shown in tab. 2.1.2.

t_1	t_2	Result	Case
$>$	$>$	invalid	-
$>$	$=$	$t_{\text{crossing}} = t_1$	0
$>$	$<$	$t_{\text{crossing}} = t_1$	1
$=$	$>$	$t_{\text{crossing}} = t_2$	2
$=$	$=$	invalid	-
$=$	$<$	$t_{\text{crossing}} = t_1$	3
$<$	$>$	$t_{\text{crossing}} = t_2$	4
$<$	$=$	$t_{\text{crossing}} = t_2$	5
$<$	$<$	invalid	-

Table 2.1.2: Possible results for left and right crossing time with resulting choice of next crossing time. $>$, $=$ and $<$ are to be read as for example $t_1 > 0$. The case indicates the case number within the simulation code.

By collecting the next crossing times for each dimension and taking the minimum of these times the exit time of the particle from its cell is determined.

The return value of the routine is an *Event* where the partner is a pointer to the corresponding entry in the box' *neighbours_lookup* table. The index lies between zero and five for to the six possible neighbour cells sharing a surface with the current cell of the particle. Each valid case represents a distinct neighbour cell and its index within the cell's *neighbours* array is clearly defined by the

cell setup routines. The indices within the neighbours array are matched with the defined cases is tab. 2.1.3.

dimension	boundary	case	index
x	front	0	12
	back	1	13
y	front	2	10
	back	3	15
z	front	4	4
	back	5	21

Table 2.1.3: Overview of the cells' *neighbours* indices directly sharing a surface for 3 dimensions. As the indices hardly follow any simple pattern they are explicitly noted at this point. Obviously the cell consists of a front and a back boundary in each dimension. The corresponding case numbers are identical to the ones from tab. 2.1.2.

Scheduler::predictCollision()

The prediction of collision times has already been discussed in section 2.1.1. The implementation in the program first calculates all necessary scalar products while accounting for the periodic boundary conditions, and in a second step returns the collision time depending on the case at hand.

The presented algorithm is only valid for single sized particles but can be extended to polydisperse systems as is shown insection 2.5.1.

As this routine is executed through out the simulation very often it has been tried to optimize its efficiency as far as possible. For example calculating only necessary results for the next case differentiation has been tried but without significant increase in efficiency and for better readability the prior version has been used. In either case if more efficient calculations are found it is useful to implement them at this point.

Scheduler::setupFEL()

This routine fills the FEL of the simulation. For this purpose it iterates through all particles and calls *setupPEL* for each of them. The PEL in turn is set up by predicting the next cell transfer as well as the next collisions with all particles within the 3^d cells in d dimensions directly surrounding the particle. From all predicted events only such with finite times are then written to the *backupEvents* vector that is the PEL of the particle.
For the FEL only the top event of each particles PEL is then used. But because other events from the PEL might move on to the FEL at later times the top event that was pushed to the FEL has to

be erased from the PEL.

Scheduler::executeTransfer()

The execution of a transfer event is accomplished by the particles *MoveBetweenCells()* routine. The departure cell is taken as the event particles own cell, while the information about the destination cell is contained in the event's *partner* variable. It points to an address within the look up table of the box where the index of the destination cell in the departure cell's neighbour array is deposited.

Scheduler::executeCollision()

The outcome of a collision between particle 1 and 2 with corresponding position and velocity can be derived by momentum and energy conservation too be

$$\vec{v}_1' = \vec{v}_1 + \left(\frac{\vec{r}\cdot\vec{v}}{\vec{r}\cdot\vec{r}}\right)\vec{r}_{12} \tag{2.1.11}$$

While eq. 2.1.11 is not directly depending on the radius, an indirect dependence by the collision time and configuration at which eq. 2.1.11 is evaluated persists. The generalization to arbitrary masses is also given in section 2.5.1.

Scheduler::executeEvent()

The execution of an event works in multiple steps. At first the topmost *Event* is copied from the FEL where it is deleted. Next the validity of the interaction counts of both particle (*cond1*) and its partner (*cond2*) are evaluated. The validation is nothing else than a comparison of the interaction counts when the event was scheduled with the present interaction counts. As the conditions are used in the following flow statements they are stored in boolean variables. Furthermore in the case of a transfer event the validation of the partner is not necessary but for better readability performed either way. It follows a distinction between 5 cases which are given by:

Valid transfer (*event_type*==0 and *cond1*)
: The transfer is executed, the global time is evolved to the event time, the particle's PEL is rebuilt and it's next event pushed to the FEL.

Valid collision (*event_type*==1 and *cond1* and *cond2*)
: The collision is executed, the global time is evolved to the event time, for both participating particles new PEL's are built and each top event is pushed to the FEL.

Invalid transfer (*event_type*==0 and not *cond1*)

The particle must have had an interaction previously where a new event for it was scheduled, thus no action is taken.

Invalid collision due to particle (*event_type*==1 and not *cond1*)

The particle must have had an interaction previously where a new event for it was scheduled, thus no action is taken.

Invalid collision due to partner (*event_type*==1 and not *cond2*)

Only the partner had an interaction previously where a new event for it was scheduled, thus a new event for the particle is required. As the particle had no further interactions, the *backupEvents* are still valid and its first entry is pushed into the FEL. In case no backupEvents are stored the PEL is rebuilt and its first entry pushed to the FEL instead.

The order of the cases might be exchanged, except for the last two. This is because the last one indirectly assumes *cond1* to be true, which is guaranteed only by the case before.

Furthermore the routine counts the number of each case, to monitor numbers of collisions, transfers and invalidated cases by type. This is not required by the simulation but can be helpful for understanding the system and simulation.

2.1.3 Simulation periphery

Well either make it a little nicer or maybe leave it out For the simulation to work also some more surrounding is required. The corresponding parts of the simulation are only briefly discussed.

Inout and ch5md

As suggested by the names, the first comprises the read and write routines of the simulation, while the later one holds routines dealing with the h5md format. The used data types are mostly single figures, arrays, variable length arrays and tables.

Setup

The setup routines are called mainly at the beginning of a simulation to either set up a simulation from a file or to completely start a new simulation.

Tools

This is the toolbox of the simulation holding routines to measure quantities like mean squared displacement, radial distribution functions and also the cluster finding routine. It also contains an overlap

and minimal distance function used within the simulation during compression.

2.2 Probe of simulation code

To probe the simulation dynamics we measured the longtime diffusion constant and the radial distribution function of the stable hard sphere liquid, as there are many measurements available in the literature to compare with.

2.2.1 Diffusive behaviour

The diffusive behaviour of particles in a liquid usually can be separated in three distinct parts. First the short time diffusion which can be understood as the random movement of the particles within their momentary cage within the fluid. Second a sub diffusive phase in which the particles are repelled for the first time by their nearest neighbours. And third the long time diffusion to describe the random propagation of the particle through the fluid over time.

As the ballistic hard sphere system enters into the long time diffusion almost at once only this is really measurable. By assuming a diffusive process we have the expectation that the average mean squared displacement (MSD) of a particle can be well described by

$$\langle x^2 \rangle(t) = 2\, d\, D\, t\,, \tag{2.2.1}$$

where $\langle x^2 \rangle$ is the expectation value of the MSD, d the number of spatial dimensions, D the characteristic diffusion constant, and t the time by which the system has evolved.

By measuring $\langle x^2 \rangle(t)$ and making a linear regression to the data points we can find the Diffusion constant D.
To probe the simulation code, systems as characterized in tab. 2.2.1 have been used. The equilibration phase has been carried out already at the final volume fraction up to $\eta_f = 50\%$, while above this an initial volume fraction of $\eta_i = 45\%$ has been used to obtain a fluid rather than a solid during the equilibration phase. As some measurementes are within the metastable regime, it has been checked that no clusters were present in the box during the measurement as they would reduce the averaged diffusion.

The resulting diffusion constants depending on the fluid volume fraction are shown in fig. 2.2.1 alongside values obtained from the literature, for a similar system.

As it can be seen the EDMD simulation is very well capable of reproducing the diffusion constant for

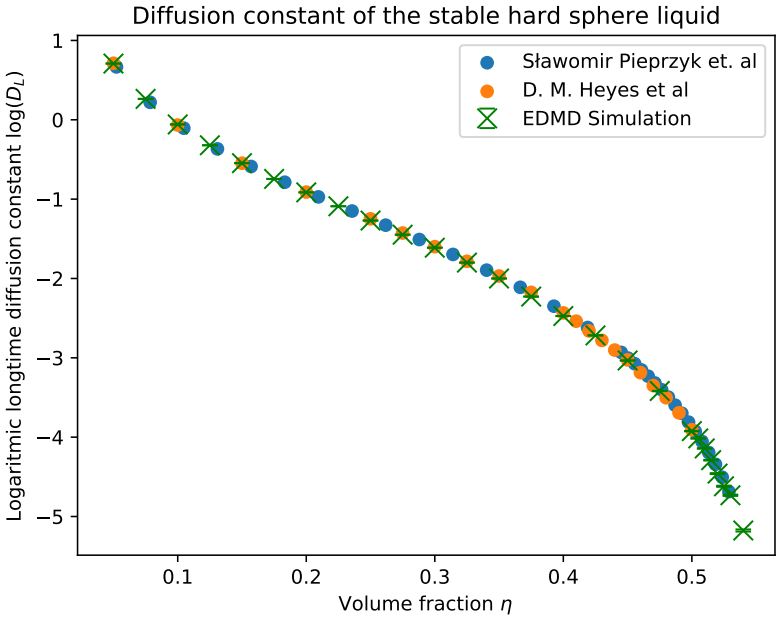


Figure 2.2.1: Logarithmic plot of long time diffusion constant of the hard sphere liquid as measured in our own simulations as well as measurements from the literature[16],[17].

the hard sphere liquid, and therefore we expect the dynamics of it to accurately represent the purely ballistic hard sphere system.

Parameter	Value
N	16384
eq_steps/particle	5000
pr_steps/particle	20000
η_i	5% ... 50 %
η_f	5% ... 54 %

Table 2.2.1: Input parameters of diffusion test systems.

2.2.2 Radial distribution function

A further well known quantity for the hard sphere system is the radial distribution function. As a theoretical prediction the Percus-Yevick approximation can be used to compare with, also it would be possible to compare with Monte Carlo simulations of the hard sphere system. In fig. 2.2.2 an overview for a range of volume fractions is shown from the same simulations used in section 2.2.1. Clearly visible is that no particles enter within the diameter of the spheres. Further for higher volume fractions the

liquid shells become very well visible. At very high volume fraction we also find that new peak arises below $r = 2\sigma$.

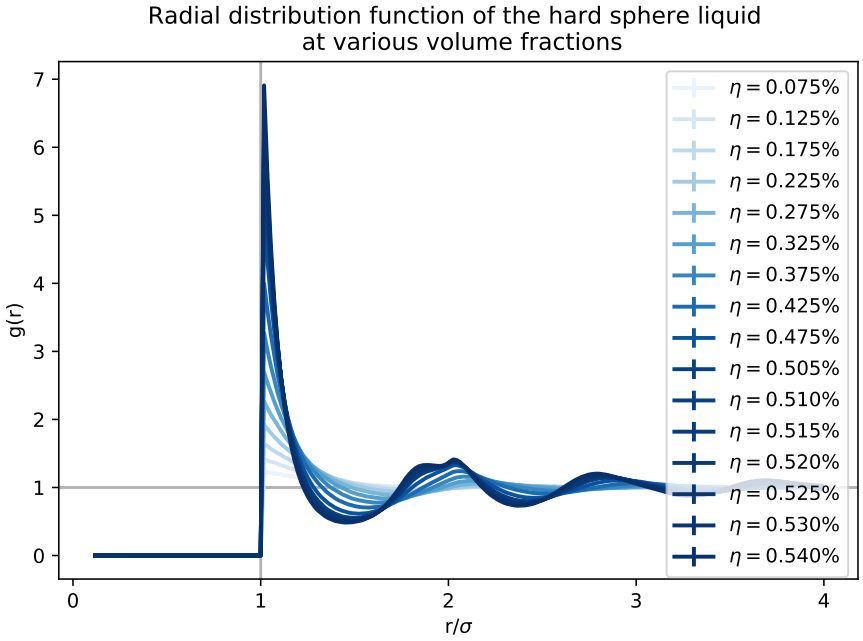


Figure 2.2.2: Radial distribution functions for a range of volume fractions. The colouring corresponds to the used volume fraction.

To compare with Percus-Yevick approximation the radial distribution function for two single volume fractions is shown with the corresponding theoretical solution in fig. 2.2.3.
As highlighted for example in [18] the theoretical approximation has some flaws as can be seen with $g(r)|_{r=1\sigma}$ being too low for the Percus-Yevick approximation. But overall the two radial distribution functions follow each other rather closely giving confidence that the developed simulation code is capable of producing accurate data in other contexts as well.

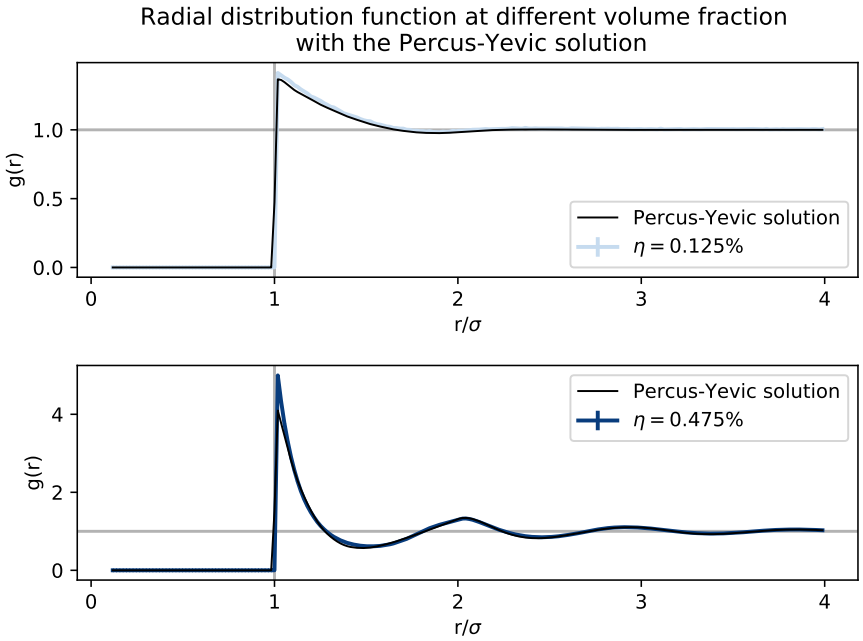


Figure 2.2.3: Radial distribution function for the hard sphere system at a low and at a high volume fraction of the liquid together with the theoretical prediction from the Percus-Yevick approximation.

2.3 Estimate of required resources

To choose system parameters reasonable, calculation times and file sizes of the simulation have been characterized. This was of interest as the program was supposed to run on the NEMO high performance computing cluster which puts hard boundaries on calculation times which when trespassed can cause tremendous loss of data if not correctly caught by the program.

2.3.1 Calculation time estimates

The calculation time of the program was tested for a large range of different system sizes up to almost 9 million particles in the fluid state. As can be seen in fig. 2.3.1 the calculation time increases proportional to the system size for the execution of a step as well as for a measurement of the fluid system. The calculation cost being of $\mathcal{O}(N)$ enables the study of large systems. Furthermore from the slope an expectation for the execution time of a single event can be deduced, as well as an expectation for the time necessary for a measurement. As discussed on the example of fig. 2.3.2 the dependence of the measurement routines on the largest cluster size were not seen here, as possible clusters remained rather small during these simulation times.

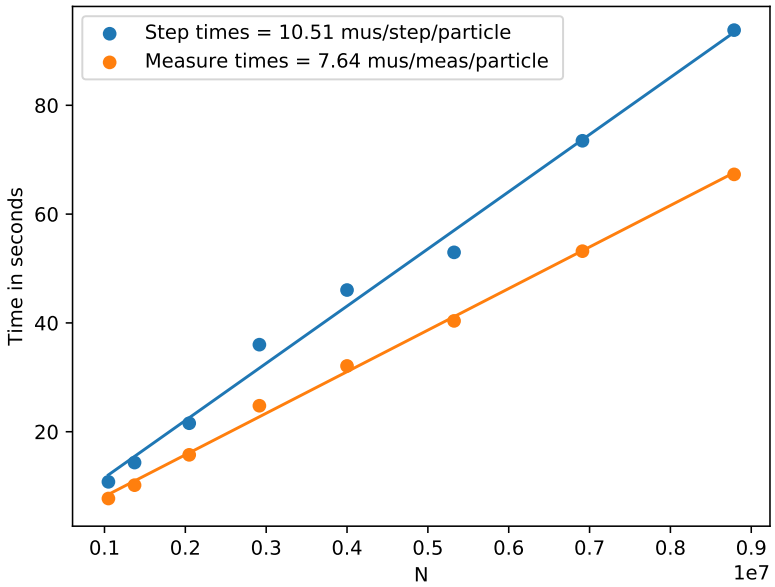


Figure 2.3.1: Overview of CPU time required for calculating a simulation step, consisting of an event for each particle, and a measurement of relevant quantities for the system. As assumed for a simulation algorithm with $\mathcal{O}(N)$ calculation effort, the data points can be described by a line rather well. As the CPU time is clearly related to the further workload of the CPU during the calculation it is also expected to find fluctuations if the other workload of the machine is not strictly controlled.

The effect of larger clusters was only investigated after problems with the runtime of the programs were traced back to these. The q6q6-order parameter routine was tested for larger clusters in a nucleating simulation with about 1 million particles within the box. As can be seen in fig. 2.3.2 the calculation cost of the cluster finding routine can be described with a quadratic dependence on the largest cluster. For an impression what this means we can use the calculation costs of a simulation step from fig. 2.3.1 being about $t_{step} \approx 10\,\mu$s/particle. Therefore the execution of one step takes about 10 s for 1 million particles. If a measurement is performed every 10th step, the calculation cost of the measurements without a large cluster remain below 10%. But as the largest cluster grows to a few hundred thousand particles in size, the measurements can make up 30 % and more of the calculation cost, or for a fixed number of steps, increase the calculation time by about 50 %. This previously unseen effect lead to actual data loss as the combination of NEMO cluster policy and EDMD simulation program did not result in a save shutdown of the program after breaching the wall time limit of the NEMO cluster.

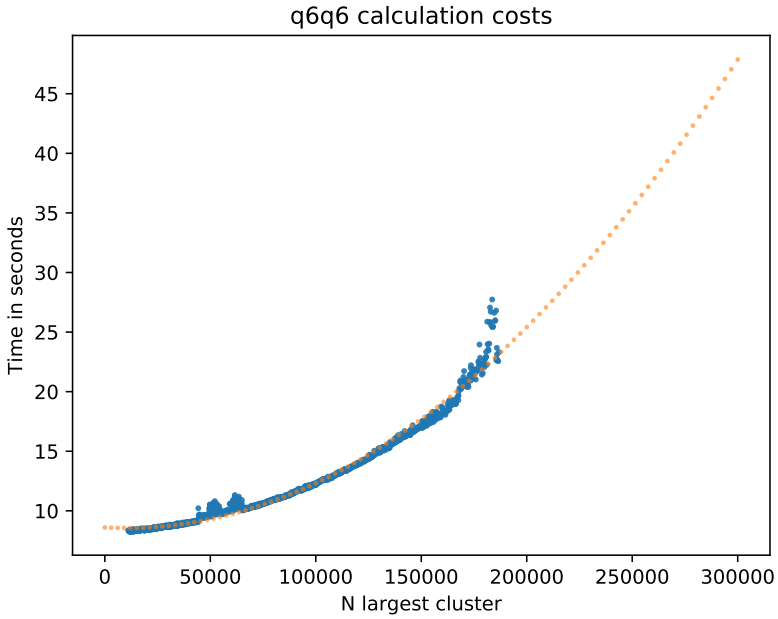


Figure 2.3.2: Calculation time of the q6q6 order parameter at an increasing largest cluster size during one nucleation, together with the quadratic best fit indicating that the q6q6 routine calculation effort can be approximated by $\mathcal{O}({N_{\mathrm{lc}}}^2)$ where N_{lc} is the size of the largest cluster.

2.3.2 File sizes estimates

A further important constraint for the simulations are the produced amount of data. To get an impression of the file sizes, the required memory for snapshots, reset steps and other measurements were measured prior to the actual simulations. The results for a single snapshot containing all positions and velocities of all particles as well as the size of a single simulation reset step containing all positions, velocities, the FEL, all PEL's and all delayed times is shown in fig. 2.3.3. It can be seen that the file size is directly proportional to the system size which clearly expected as each particle adds a further set of positions, velocities etc. to the saved data.
The memory costs of other measurements have been left out of fig. 2.3.3 as these only amount to substantial file sizes if measurements at about each step for long simulations are done.

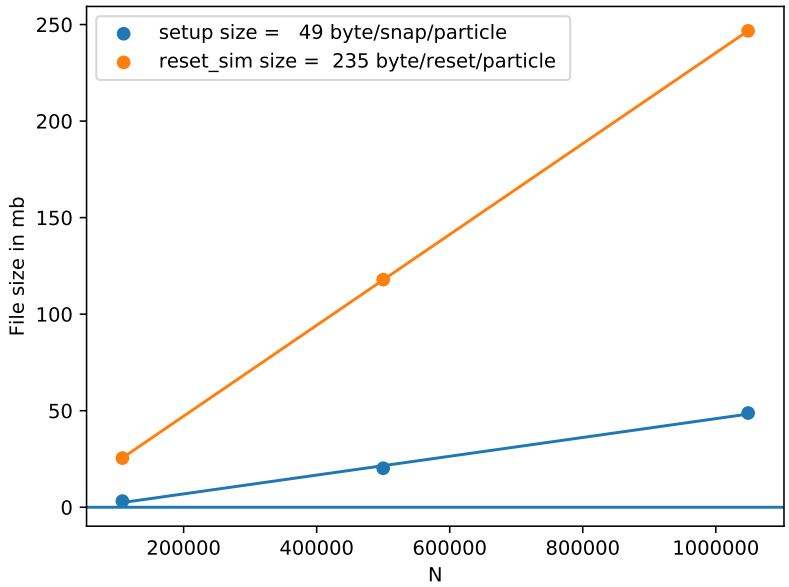


Figure 2.3.3: Overview of file sizes when a single setup on the one hand and a single full simulation on the other hand is saved for comparison reasons together with their corresponding linear regression. While the linear regression for 3 points is statistically not exceedingly meaningful it still remains a useful tool to extract the slope which corresponds to the required memory per particle and snapshot or reset simulation.

2.4 Preliminary data for testing equilibration

The motivation for the simulation code is based on the interest in nucleation rates of the hard sphere system at varying volume fractions. To observe a nucleation the volume fraction of hard spheres has to be changed rapidly from lower ones where the system is in the stable fluid phase to higher ones where a meta stable fluid-solid phase exists. If this metastable phase is evolved in time nucleations can be observed as stochastic distributed events. To measure those without effects originating from the handling of the simulation, some parameters were tested within reasonable ranges prior to the data production.
For this simulation the equilibration steps as well as the initial density before the volume quench seemed like they could introduce unwanted artifacts, and thus we performed some smaller data series to evaluate if and when these effects might come into play.

The used test system is characterized by the figures in tab. 2.4.1.

The general behaviour of the systems is analysed by inspecting the cluster distribution over time. The mean cluster distribution is shown in fig. 2.4.1 together with the same data smoothed by a Gaussian

Parameter	Value
N	16384
eq_steps/particle	100 ... 20000
η_i	5% ... 49 %
η_f	54 %

Table 2.4.1: Input parameters of test systems probing the dependence on equilibration steps and initial density.

filter matrix. The smoothing is used because in a next step the difference between the mean cluster distribution and the cluster distributions with varying simulation parameters is compared, and without smoothing at low count rates only fluctuations are visible.

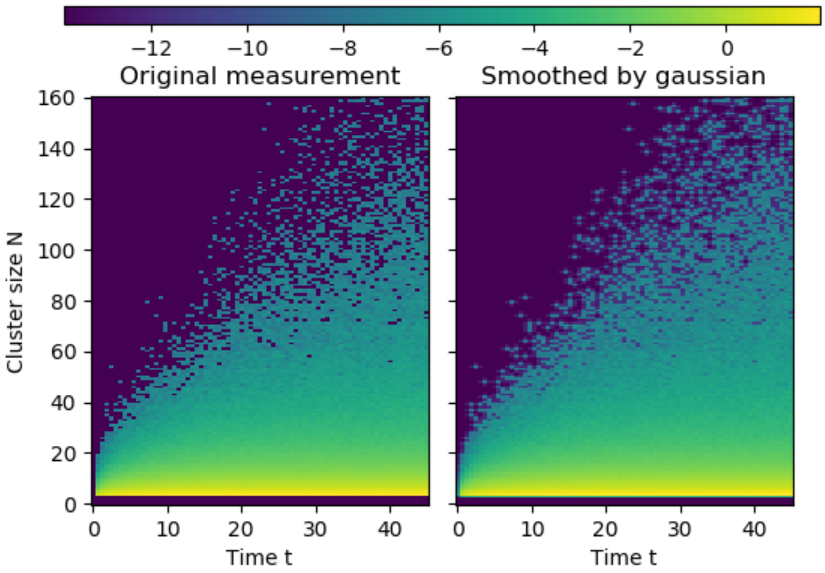


Figure 2.4.1: Heat map of the mean cluster distribution over time. The diagram encompasses 800 trajectories of 16384 particles each. The colouring indicates the logarithm of the mean cluster occurrence corresponding to a probability in the stationary case.

From fig. 2.4.1 we can see how the system behaves after a volume quench into the metastable region. In the liquid rarely any clusters are present and thus directly after the quench no clusters are present either as the spatial configuration requires time to rearrange into clusters. In the later evolution we see how clusters form, and soon after begin to nucleate leaving the range of the diagram.

To compare simulations with varying parameters the quantity defined in eq. 2.4.1 is used, where complications with zero values are circumvented by fixing these values below the regular signal.
Three samples of this comparison are shown in fig. 2.4.2 and fig. 2.4.2. The colouring indicates $\Delta_{p(N,t)}$

defined in eq. 2.4.1. As mentioned above the quantities $p_i(N,t)$ and $\langle p(N,t)\rangle$ have been smoothed by a Gaussian filter, because the number of samples included, with 100 trajectories per series, were not sufficient to produce smooth distributions at the given sampling rate. Thus without smoothing only fluctuations would be visible.

$$\Delta_{p(N,t)} = \log\left(\left|\frac{p_i(N,t)}{\langle p(N,t)\rangle} - 1\right|\right) \tag{2.4.1}$$

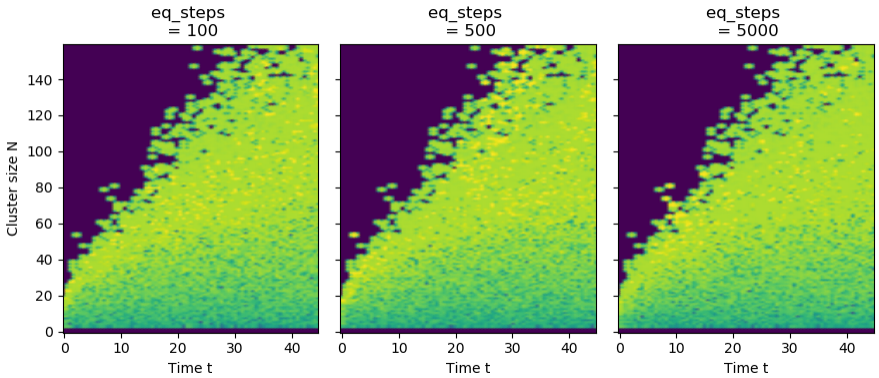


Figure 2.4.2: Heat map of differences between the cluster distributions within simulations carried out with varying the length of the equilibration phase. The quantity used for colouring is defined in eq. 2.4.1, where yellow indicates a large difference while blue indicates a small difference. Providing a legend of the colouring is omitted as $\Delta_{p(N,t)}$ has no further use than to indicate differences and actual values do not add any use.

On first sight none of them differ in their general behaviour. Because at t=0 after the quench no clusters have formed yet and also no clusters were present in the stable liquid, the difference between all simulations is zero, indicated by the blue region in the top left corner. The features visible on the edge between the zero region and the nonzero region on the other side are the same, because they are features of the mean distribution carried through. Actual differences not due to fluctuations can only be seen within the green and yellow non-zero region, but none such differences is observed.

While it seems like the initial volume fraction of $\eta = 0.4$ and $eq\_steps = 5000$ include less irregular fluctuations, dramatic effects from choosing the simulation parameters can be excluded. Interesting in this context are especially the simulations with $eq\_steps = 100$ because after executing 100 events/particle on average, the initial perfect crystal configuration is only on the edge of not being detected anymore. For this reason one could expect that a significant part of these configurations might directly crystallize again, but instead we do not detect any significant difference.

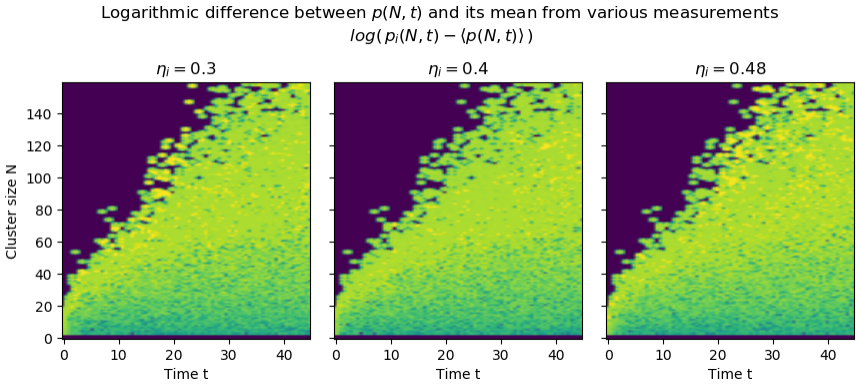


Figure 2.4.3: Heat map of differences between the cluster distributions within simulations carried out with varying the volume fraction of the liquid during the equilibration phase. The quantity used for colouring is defined in eq. 2.4.1, where yellow indicates a large difference while blue indicates a small difference. Providing a legend of the colouring is omitted as $\Delta_{p(N,t)}$ has no further use as to indicate differences and actual values do not add any use.

A more detailed analysis of the differences is given by calculating the mean nucleation rates assuming classical nucleation theory. This is done for the data shown in fig. 2.4.4. The calculations of the rates have been carried out as described in section 3.8.
As we see, no significant difference in the nucleation rates can be observed even for the extreme short equilibration phase of 100 events per particle. For this bold setting the rate is a little higher, but still in accordance with the other measurements within its statistical uncertainty.

Overall we conclude in this chapter that as long as parameters are set within reasonable boundaries, we expect not to have systematic influences of simulation parameters.

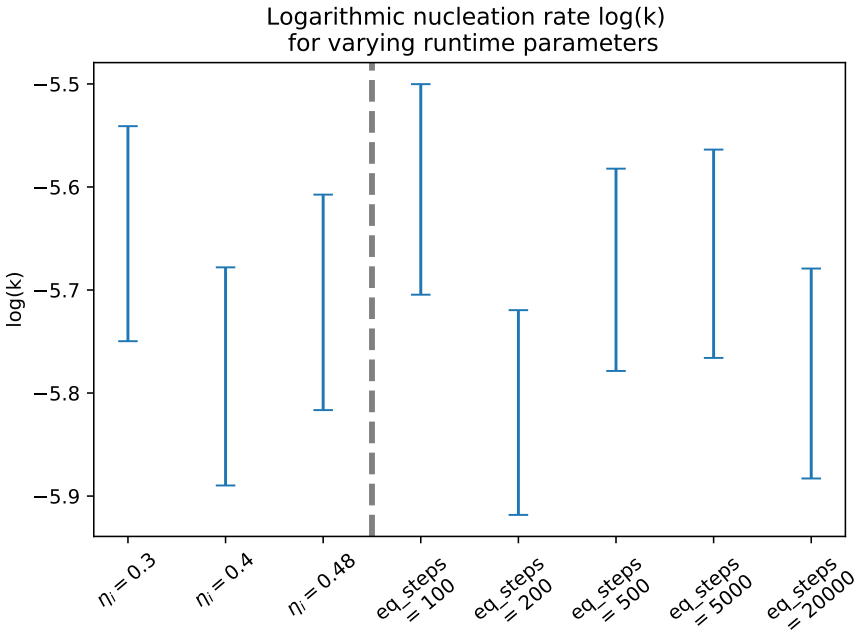


Figure 2.4.4: Comparison of nucleation rates under CNT assumptions for different initial densities during equilibration with eq_steps fixed at 5000, as well as varying eq_steps with η_i fixed at 0.45.

2.5 Extensions

The program at this state is capable of simulating large systems including compression and relaxation. While it has been used to study the nucleation of the monodisperse hard sphere fluid in this thesis, further features have been developed to suit the code for further studies. Polydispersity in the sense of radii and masses have been implemented and roughly tested, as well as individual cluster tracking, to enable detailed study of spatial information about the clusters. These two features are described and their use summarized in the following section.

2.5.1 Varying radius

Polydispersity has been included in the simulation to make comparison with the real world simpler, as in actual experiments monodisperse spheres are practically not archived. For the implementation the prediction of collisions has to be adjusted. When looking at the derivation of eq. 2.1.9 it is found that σ being the former diameter of a sphere in the monodisperse case has to be changed to $\sigma = R_i + R_j$. In the equations the same definitions of scalar products are used as before in section 2.1.1.

$$\Delta t = \frac{(rr - \sigma^2)}{-rv + \sqrt{(rv)^2 - vv(rr - \sigma^2)}} \tag{2.5.1}$$

For a physical model in which the particles are made of some matter with constant density the change of the radius is also accompanied by a change of the mass. This has to be taken into account when assigning the velocities after a collision as written in eq. 2.5.2.

$$\begin{aligned} \vec{v}_i\,' &= \vec{v}_i + \frac{2m_j\ (rv)}{(m_i + m_j)\sigma^2} \cdot (\vec{r}_j - \vec{r}_i) \\ \vec{v}_j\,' &= \vec{v}_j + \frac{2m_i\ (rv)}{(m_i + m_j)\sigma^2} \cdot (\vec{r}_j - \vec{r}_i) \end{aligned} \tag{2.5.2}$$

A small caveat is given by the fact that the simulations should be run within the center of mass frame, as otherwise unnecessary transition events have to be calculated.

2.5.2 Individual cluster tracking

Following trajectories of single metastable clusters within the fluid is useful as for example mean lifetimes of these fluctuations can be measured by it. Also the nucleation time can be measured with higher precision as the precursor can be tracked back to only a few particles.
Because the clusters themselves form out of the liquid and are not numbered and easily distinguishable as the particles in the simulation, they have to be identified for each measurement step. They are mostly characterized by the participating particles and their center of mass position, of which the latter one is easier comparable and accessible in our case. An algorithm based on a maximum of expected movement from one time step to the other is tested and already yields reasonable results as can be seen in fig. 2.5.1.
Information about the lifetime and size of the fluctuations derived from the analyzed example trajectory shown in fig. 2.5.1 are depicted in fig. 2.5.2. First we note that both the maximum size and the mean size can be used as a measure for the scale of the flucutations, as the results do not vary by a lot. Further we can read from the diagram that at a volume fraction of $\eta = 53.4\%$ there are a lot of small to medium sized clusters with short lifetimes up to about $10\delta t$, and some large clusters with lifetimes of more than $15\delta t$. The overal impression is that the fluctuation distribution is compact with a small but very far reaching tails towards the large lifetimes as well as towards the large cluster sizes.

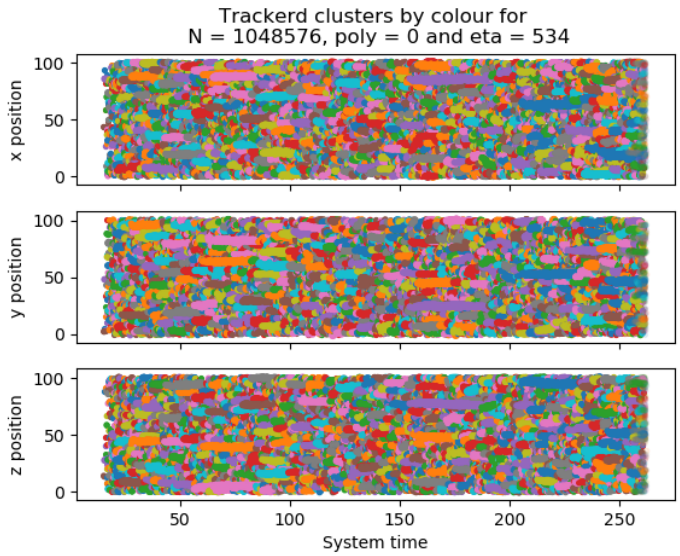


Figure 2.5.1: Example results of cluster tracking algorithm in a monodisperse simulation. The three plots are the projections of the box onto the three spatial dimensions over time. Each cluster is given a color which does not change over time. With it we can see for example that two clusters mingling in one projection are actually some distance apart from each other in an other dimension.
While in this example only small metastable clusters that dissolve after some time are present, also nucleation events can be seen in this kind of plot. These are easily identfied as the linewidht of the lines are drawn proportional to the diameter of a sphere with a volume corresponding to the clusters volume under the assumption of a spherical cluster.

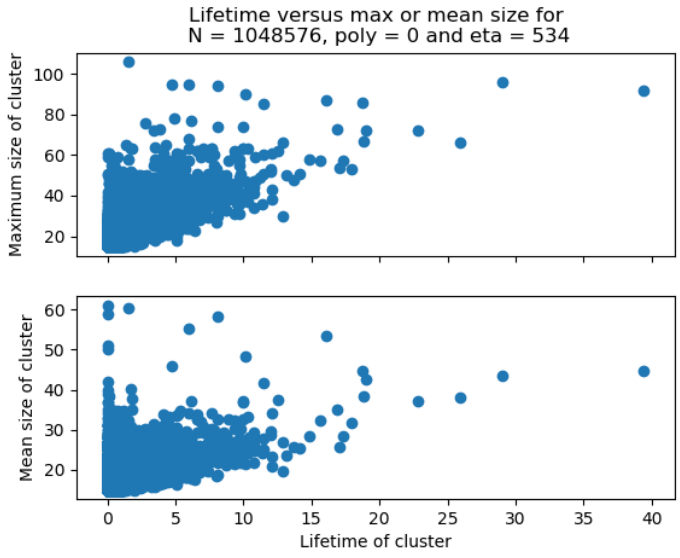


Figure 2.5.2: Example of lifetime depending on maximum (top) or mean (bottom) size of the metastable cluster.

3 Data Analysis

3.1 Parameter choice of the simulated system

As an integral part of this work large scale simulations have been executed on the NEMO High performance computation cluster. The input parameters of the simulated systems are given in tab. 3.1.1.

Parameter	Value
N	1048576
eq_steps/particle	1000
pr_steps/particle	20000 ... 60000
η_i	45.0 %
η_f	53.1% ... 53.4 %

Table 3.1.1: Input parameters of large scale simulations on the NEMO HPC cluster. The varying steps during production come by the fact, that 20000 steps were estimated to be calculated within 3 days leaving 1 day of buffer to the hard wall time limit of 4 days. Due to the increasing calculation cost of the q6q6 cluster routines for large clusters the wall time limit was still breached and without proper reset steps the datasets could not be restarted without large calculation overhead as all lost data has to be replaced, and the broken reset steps within the files would have to be removed prior to further simulations. Therefore the last proper version of the files were used resulting in varying simulation lengths but with usually a nucleation event in case of early breakdown.

The simulations consist of four series at volume fractions of $\eta = 0.531, 0.532, 0.533, 0.534$, where each series again consists of 500 trajectories. Therefore at each volume fraction a total number of about half a billion particles have been simulated in the metastable fluid.

The volume fractions have been chosen too probe nucleation to the lowest possible limit. As single nucleations have been observed down to volume fractions of $\eta = 53.2\%$, the lowest volume fraction was set to just below this value, as the large statistic of 500 trajectories was expected to still yield enough nucleation events to measure the nucleation rate.

The size of the systems was chosen comparably large with about 1 million particles. These large systems intuitively seem to be in conflict with the long induction times, even though using CNT as a guideline it can be shown that the computational effort for simulating nucleation events does not

significantly increase with increasing system size. The calculation time per unit of simulation time is proportional to N, it is at a given volume fraction also proportional to the volume V:

$$\frac{T_{CPU}}{\delta t_{Sim}} \propto N \propto V \tag{3.1.1}$$

Further we expect the nucleation time in terms of the system time $\langle \tau_{Nucleation} \rangle$ to be proportional to the inverse of the system volume if assuming a nucleation rate density independent of time:

$$\langle \tau_{Nucleation} \rangle \propto \frac{1}{V} \tag{3.1.2}$$

As the required CPU time for a nulcleation event is simply proportional to the product of $\langle \tau_{Nucleation} \rangle$ and the calculation time per unit of system time we can conclude:

$$\langle T_{CPU} \rangle \propto \frac{T_{CPU}}{\delta t_{Sim}} \cdot \langle \tau_{Nucleation} \rangle \propto \frac{V}{V} = \text{const.} \tag{3.1.3}$$

Thus the size of the system is only relevant to be chosen smaller if ordering processes are important for the system resulting in an induction time that is independent of the system size. This might be the case for polydisperse systems, but in the monodisperse case the above reasoning was found to hold true.
An other objective that has to be considered is that less configuration snapshots of the system can be stored, as these requrie a lot of space. If one is interested in quantites like $g(r)$ this is not a problem as the necessary statistics can be either derived from a large set of small snapshot or from a small set of large snapshots, but for example resolving and storing the dynamcs of a configuration for a growing cluster would require using smaller system sizes as files easily grow to many GB's in size.

3.2 Diffusion in the metastable liquid

Diffusion or more precisely self-diffusion, characterizes the movement of the single particles within the system. The diffusive behaviour for many system can be subdivided into different regimes with different physical meaning.

For the ballistic hard sphere system we have an extremely short period in which most particles are freely moving without constraint, the ballistic regime. This could be resolved in the simulation by taking measurements at extreme rates, like after every event, but is not as the result could if desired also be determined by measuring the velocity distribution.

The short time diffusion usually seen in many systems is not seen in the ballistic hard sphere system.

To understand this we can look at the physical interpretation of the short time diffusion. While it does not describe the movement of particles between different liquid cages, it only describes the Brownian motion of particles in the suspension within their momentary liquid cage. As the simulated system does not contain any suspension there is no short time diffusion.

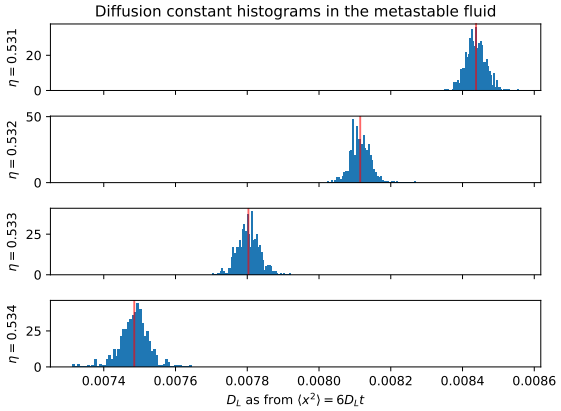
The long time diffusion on the other hand describes the movement of single particles in the simulated systems on long time scales. The interpretation of the long time diffusion is that particles are able to change their momentary cage by collisions and thus can diffuse throughout the whole system until finite size effects stop their further diffusion. If circumventing finite size effects by using unwrapped coordinates the long time movement of the particles is governed by the relation eq. 3.2.1 which was first described by Einstein[19].

$$D_L^S = \lim_{t\to\infty} \frac{\langle(\vec{r}(t) - \vec{r}(0))^2\rangle}{2dt} \tag{3.2.1}$$

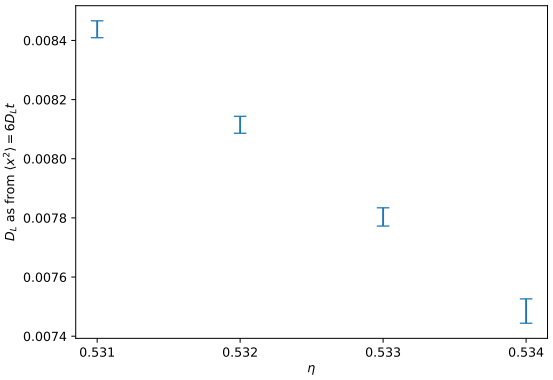
With D_L^S the longtime self-diffusion constant which will in the following be denoted only by D, $\vec{r}(t)$ the position of a particle at time t, d the number of spatial dimensions of the system and $\langle ... \rangle$ the expectation value of the ensemble.

The average is measured in the system by saving an reference position of all particles at one point, and further carrying a set of unwrapped positions through out the simulation. The average of all particles difference in their reference position with their unwrapped position is used as a measurement of the ensemble average. Especially for large system of 1 million particles, this quantity has only very small fluctuations as can be seen in fig. 3.2.1.
The diffusion coefficients are important to know for comparison between different systems. This importance is based on the idea that the fundamental mechanisms for nucleation and cluster growth do not vary between different hard sphere like systems, but are only scaled by the varying diffusion times. Furthermore there are theoretical predictions for the relationship of short time and long time diffusion, making it possible to compare experiments where the short time diffusion behaviour is better accessible with the ballistic simulations where only the long time diffusion constant is measurable.

As we see in fig.3.2.1 the diffusion constants can be measured with rather good precision with a relative standard deviation of $\sigma_D/D \approx 1\%$. Hence it does not introduce large uncertainties when normalizing time related quantities by the diffusion time $\tau_D = D^{-1}$.



(a) Histograms of the slopes for the linear regressions to the mean squared displacements. The histograms are for $\eta = 0.531, 0.532, 0.533, 0.534$.



(b) Mean of the histograms with the uncertainty on the mean given by $\sigma_{\langle D \rangle} = \sigma_D / \sqrt{n}$ with n being the number of measurements included in the average.

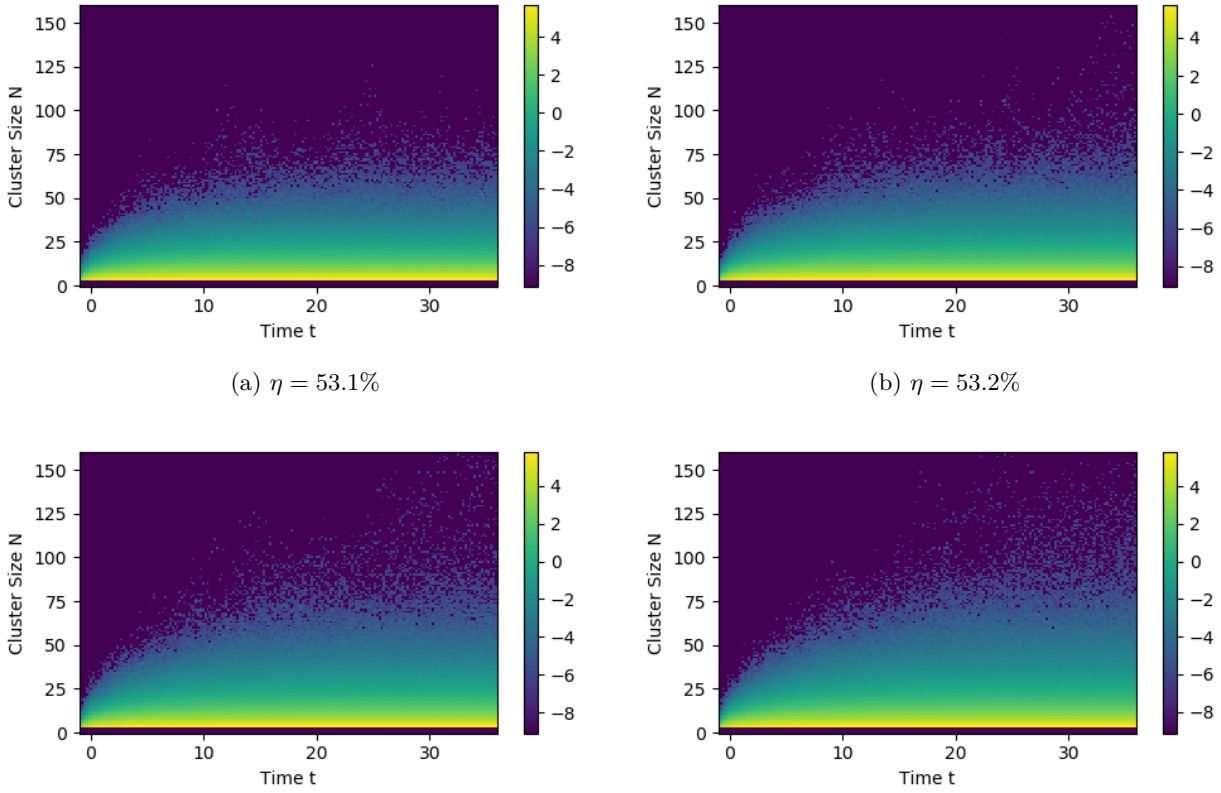
Figure 3.2.1: Comparison of long time self diffusion constants at different volume fractions as histograms and their means with uncertainty.

3.3 Cluster size distribution over time

The cluster size distribution of the system can be used to test the assumption of Markovian dynamics by trying to find a Fokker-Planck equation describing the time evoluation of the distribution. This has been done for the Lennard-Jones system by Kuhnbold et al.[4]. Testing the trajectories shown in **??** and **??** in a similar fashion would yield a good comparison but due to time constraints of this thesis it is not done. We still can illustrate some characteristics of the metastable fluid directly after and long after the quench as it compactly shows some main features of the system.

The cluster size distributions are the averages over all trajectories at a given volume fraction. While they are normalized by the number of included measurements they have not been normalized by the volume. The maximum cluster size is set to 160 as above this value only nucleating trajectories can be seen. Also a logarithm to the base of 10 is used, and cluster sizes not present at a given time step have been fixed to a value below the minimal signal as the logarithm requires non zero values.
The logarithmic representation is used as the mesurements span orders of magnitude and further can be interpreted as a quantity proportional to a free energy under the logarithm. This is justified by assuming that stationary states may fluctuate in a free energy landscape where the probability for a particular state with some energy δE follows a Boltzmann distribution $p \propto \exp\left(-\frac{\Delta E}{k_B T}\right)$.

In fig. 3.3.1 we can see the initial phase after the quench. As the fluid before the quench was at a volume fraction of $\eta = 45\%$ only very little local ordering is present directly after the quench. This



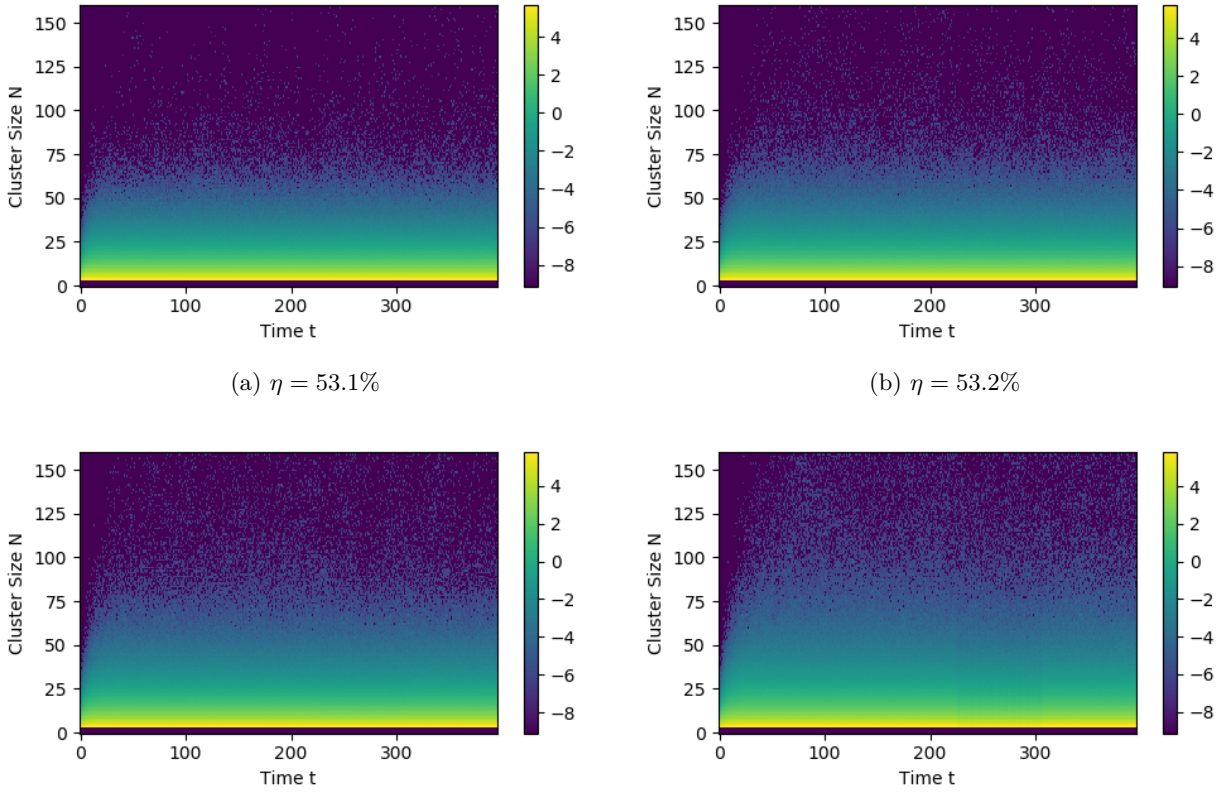
(a) $\eta = 53.1\%$ (b) $\eta = 53.2\%$

(c) $\eta = 53.3\%$ (d) $\eta = 53.4\%$

Figure 3.3.1: Cluster distributions at different volume fractions in the initial phase after the quench.

changes within the first $15-25\delta t$ after which the distribution becomes stable, where the interval lenght depends on the volume fraction. As this first time interval shows how long it takes for the system to build up the local ordering in the metastable liquid, and the unstable clusters tend to be larger for higher volume fractions the length of the interval might be explained by the larger number of particles required to find its ordering.
To further compare the system time with the more intuitive number of collisions each particle had on average we can use that at the given volume fraction we find $1\delta t \approx \frac{60 steps}{particle}$. When further using a collision probability of $\sim 40\%$ for each executed event, we find that $1\delta t \approx \frac{25 collisions}{particle}$. As a result we can conclude that it takes a few hundred collisions for each particle to build up the local ordering with unstable clusters.

The diagrams in fig. 3.3.2 show a zoomed out version of the same data depicted already in fig. 3.3.1. We see that the distribution that is reached at the end of the initial phase remains stable over prolonged periods of time. Only the nulceation events, which account for most of the probability at largest



(a) $\eta = 53.1\%$

(b) $\eta = 53.2\%$

(c) $\eta = 53.3\%$

(d) $\eta = 53.4\%$

Figure 3.3.2: Cluster distributions at different volume fractions during the waiting time.

cluster sizes, indicate that this is not a stable process.

3.4 Autocovariance functions of largest cluster in metastable fluid

The autocovariance function (ACF) of the largest cluster contains information about how long a single cluster persists as the largest cluster within the volume, as fluctuations of clusters at different points of the volume are expected to be independent of each other, only the size fluctuation of a distinct cluster should be correlated in time.

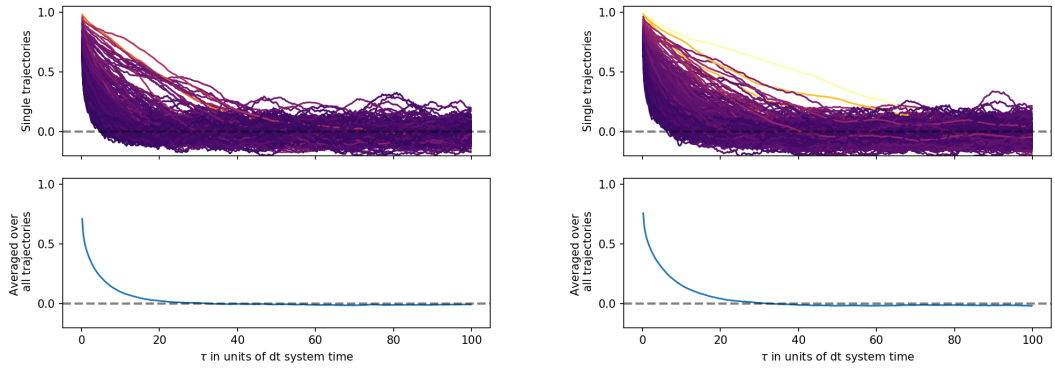
The autocovariance function is defined by eq. 3.4.1, where $N_{\mathrm{lc}}(t)$ is the number of particles in the largest cluster at time t, $\langle N_{\mathrm{lc}} \rangle_t$ is the corresponding average over time, thus $X(t)$ describes the deviations from the average. The autocovariance function furthermore is normalized by $\langle X^2 \rangle$, the variance of the data, such that $ACF(\tau = 0) = 1$.

$$ACF(\tau) = \frac{\langle X(\tau) - X(0) \rangle}{\langle X^2 \rangle} \tag{3.4.1}$$

$$\text{with } X(t) = N_{\mathrm{lc}}(t) - \langle N_{\mathrm{lc}} \rangle_t \tag{3.4.2}$$

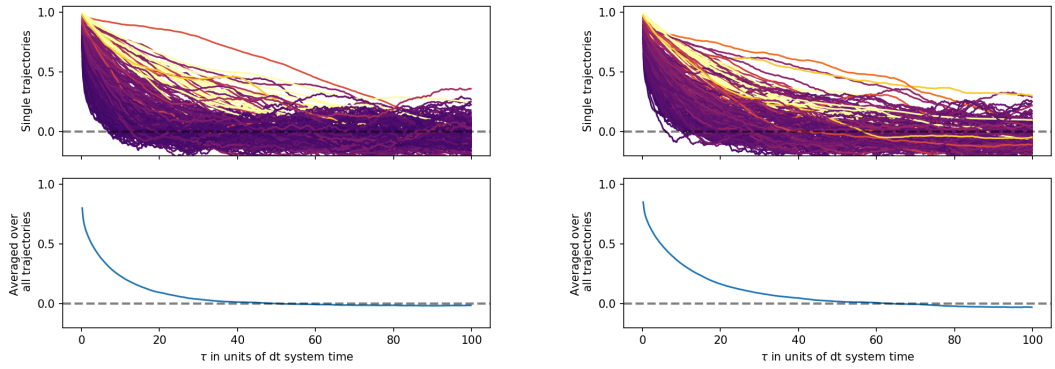
The ACF is calculated from the largest cluster measurement for each trajectory. Because soon as a nucleation event occurs the largest cluster will surely be correlated to its former size, only those parts of the measurements that did not involve strong cluster growth are used. Therefore the ACF shown in fig. 3.4.1 show the correlations of the largest cluster in the metastable fluid.

From the autocovariance functions we see that structural fluctuations persist for longer times at higher volume fractions. From the colouring as well as from the cluster distributions we can also conclude that the fluctuations tend to be larger at higher volume fractions and that for $\eta = 53.4\%$ a signal from nucleation events might not be completly negligible anymore. Still this behaviour was also seen in the cluster size distributions (section 3.3).
Also the time scale on which the ACF decays corresponds closely to the initial ordering time observed for the cluster distribution directly after the quench. And further it also correpsonds to the lifetimes of large clusters found in the single example of the individual cluster tracking algorithm (fig. 2.5.2). This leads to the conclusion that these three observations all show the same time scale of local ordering processes in the metastable fluid.



(a) $\eta = 0.531$

(b) $\eta = 0.532$



(c) $\eta = 0.533$

(d) $\eta = 0.534$

Figure 3.4.1: Comparison of autocovariance functions in the metastable fluid. The top of eache diagram depicts all trajectories with colouring indicating the largest cluster size within the used time interval. The lightest colour thereby indicates a largest cluster of more than 500 hundred particles which is a nucleation event, but these are rare in the given selection and therefore the data represents the metastable fluid still well. The bottom of each diagram shows the average of the above one with decay times of $15\delta t - 35\delta t$.

3.5 Cluster growth

Once the clusters reach a certain size they are expected to grow with new particles being attached to the surface at a constant rate leading to a growth with a proportionality of $N \propto t^3$ as shown in eq. 3.5.1 where k is the assumed constant attachment rate, N is the number of particles in a specific cluster, A is the surface of the cluster, R is the radius of the cluster and ρ_{solid} is the bulk density which is for large clusters a good approximation of the cluster density.

$$
\begin{aligned}
&\dot{N} = kA \\
&\quad \Big| \quad \text{with} \quad N = \frac{4}{3}\pi R^3 \rho_{solid} \\
&\qquad \Leftrightarrow R = \left(\frac{3N}{4\pi\rho_{solid}}\right)^{\frac{1}{3}}, \\
&\quad \Big| \quad \text{and} \quad A = 4\pi R^2 \\
&\qquad \Leftrightarrow A = \left(\frac{4\pi 3^2}{\rho_{solid}^2}\right)^{\frac{1}{3}} N^{\frac{2}{3}}, \\
&\frac{dN}{dt} = k\left(\frac{4\pi 3^2}{\rho_{solid}^2}\right)^{\frac{1}{3}} N^{\frac{2}{3}}
\end{aligned}
$$

From the bottom left side

$$
\begin{aligned}
&\Rightarrow \quad dN\ N^{-\frac{2}{3}} = dt\ k\left(\frac{4\pi 3^2}{\rho_{solid}^2}\right)^{\frac{1}{3}} \\
&\quad \Big| \quad \text{setting} \quad N(t=0) = 0 \\
&\Leftrightarrow \quad 3N^{\frac{1}{3}} = k\left(\frac{4\pi 3^2}{\rho_{solid}^2}\right)^{\frac{1}{3}} t \\
&\Leftrightarrow \quad N^{\frac{1}{3}} = k\left(\frac{4\pi}{3\rho_{solid}^2}\right)^{\frac{1}{3}} t
\end{aligned}
\tag{3.5.1}
$$

As the systems are able to accommodate clusters up to a few hundred thousand particles and usually only one very large cluster is formed during a simulation, the attachment rate can be measured by a linear regression to the third root of the number of particles in the largest cluster over time. This is visualized for the trajectories at $\eta = 0.532$ in fig. 3.10.1. The volume fraction $\eta = 0.532$ is chosen arbitrarily.
Subsequently the slopes of the linear regressions have been collected in histograms shown in fig. 3.5.2. As shown in eq. 3.5.1 these slopes correspond to constant attachment rates with a dependence on the density within the cluster. As the densities of concern are very close to each other, they only introduce a relative difference of 0.5% between the rates of lowest and highest volume fractions. For this reason the dependence is neglected for the qualitative comparison.

What we see from the histograms is that the distribution is rather spread out, but interestingly not significantly depending on the volume fraction. Except for $\eta = 0.531$ we find a smaller growth rate. A possible explanation for this behaviour could be that growth by heterogeneous crystallization on the growing cluster surface, leading to a mean higher growth rate for higher volume fractions, is less likely for the lower volume fractions. Either way due to the low statistics at the lowest volume fraction it is also possible that only a statistical fluctuation is seen. From the similarity of the growth rates we can

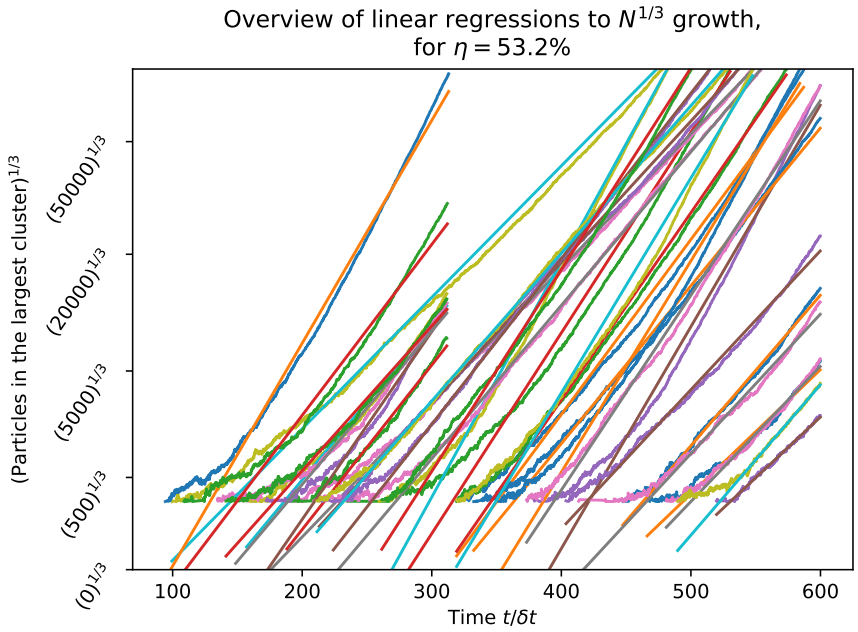
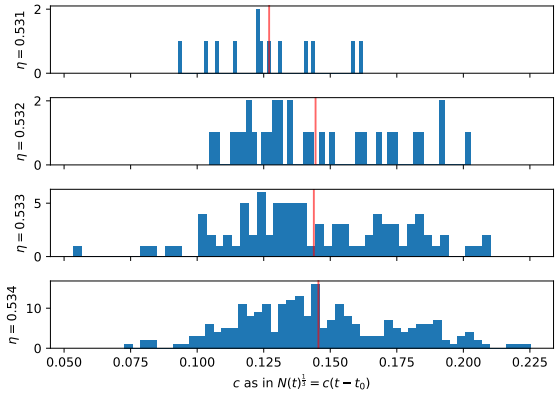


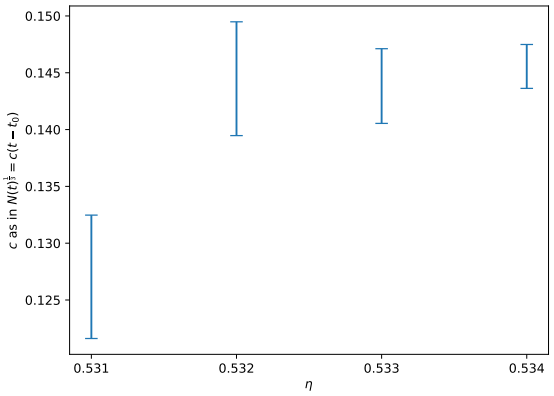
Figure 3.5.1: Trajectories of the third root of the number of particles within the largest cluster of a system over time. Clearly visible is the linear proportionality for which a linear regression is shown together with the data. The cut of some data sets at $t/\delta t \approx 300$ is due to the trespassing of the maximum wall time of the NEMO cluster. This means that the simulation of 20000 production steps yields a system time of $T/\delta t \approx 300$. Clusters present already in the first step would become too large in the next simulation interval, leading to a breach of the wall time limit due to the quadratic effort required for the q6q6 cluster finding routine. It can be assumed that clusters forming just around $t/\delta t \approx 300$ might not have been recognized due to this flaw. But as the number of trajectories concerned by this is rather small the impact is not easy to recognize when looking at the induction time distributions in fig. 3.7.1.

deduce that the attachment of the particles to the cluster is a reaction controlled process. is there a reasoning for it being a reaction or diffusion controlled process?

As the diffusion constants vary from $D = 0.0081|_{\eta=0.532}$ to $D = 0.0075|_{\eta=0.534}$ they span a difference of about 7.5%, but does that mean they are either reaction or diffusion controlled, or is their only nothing to see, as the uncertainty on the growth rate is also of the size 5 % ?



(a) Histograms of the slopes for the linear regressions to the largest clusters during the later stable growth process. The histograms are for $\eta = 0.531, 0.532, 0.533, 0.534$.



(b) Mean of the histograms with the uncertainty on the mean given by $\sigma_{\langle c \rangle} = \sigma_c/\sqrt{n}$ with n being the number of measurements included in the average.

Figure 3.5.2: Comparison of growth rates in the constant attachment regime.

3.6 Tensor of Gyration properties

The tensor of gyration is a very useful tool as it describes the second moments of the position distributions. It comprises information about the spatial extent in all three dimensions with commonly defined quantities being the radius of gyration, asphericity and anisotropy[20], which are further discussed in the following.

The tensor of gyration itself is defined by eq. 3.6.1.

$$S_{mn} = \frac{1}{N} \sum_{i=1}^{N} r_m^{(i)} r_n^{(i)} \tag{3.6.1}$$

$$\text{with} \quad \sum_{i=1}^{N} \vec{r}^{(i)} = 0 \tag{3.6.2}$$

As described by eq. 3.6.2 the matrix S_{mn} is calculated in the center of mass frame for particles with the same mass. The tensor of gyration can be diagonalized, with the three Eigenvalues λ_1^2, λ_2^2 and λ_3^2. The Cartesian system thereby is chosen such that $\lambda_1^2 \leq \lambda_2^2 \leq \lambda_3^2$. These Eigenvalues correspond to the spatial extents of the cluster within the Cartesian system in which the tensor of gyration becomes diagonal. From these three Eigenvalues the quantities defined in eq. 3.6.3 - 3.6.6 are common to characterize clusters of particles.

$$\text{(squared) Radius of gyration:} \quad R_G^2 = \sum_{i=1}^{3} \lambda_i^2 \tag{3.6.3}$$

$$\text{Asphericity:} \quad b = \lambda_3^2 - \frac{1}{2}(\lambda_1^2 + \lambda_2^2) \tag{3.6.4}$$

$$\text{Acylindricity:} \quad c = \lambda_2^2 - \lambda_1^2 \tag{3.6.5}$$

$$\text{Relative shape anisotropy:} \quad \kappa^2 = \frac{b^2 + \frac{3}{4}c^2}{R_G^4} = \frac{3}{2}\frac{\sum_{i=1}^{3} \lambda_i^4}{\left(\sum_{i=j}^{3} \lambda_j^2\right)^2} - \frac{1}{2} \tag{3.6.6}$$

For better understanding of the shape descriptors mentioned before, we can have a more detailed look at their interpretation:

Radius of gyration R_G:

An averaged radius of the structure.

Asphericity b:

The difference of the largest extent with an average of the two smaller extents.

Acylindricity c:

The difference of the smaller extents

Relative shape anisotropy κ^2:

A sum of the asphericity and the acylindricity normalized by the radius of gyration to obtain a dimensionless quantity between 0 and 1.

To spot possible correlations between a cluster's shape and it's growth, the three quantities eq. 3.6.3 - 3.6.6 derived from the tensor of gyration have been plotted against the cluster size and then coloured by three scalar quantities characterizing the growth process of each trajectory.
The first of them is the induction time, as early nucleations might arise from less ordered clusters resulting in a higher asphericity. The second is the constant attachement rate during cluster growth, where similarly one may expect that clusters including more defects may grow slower and also be less spherical. The third quantity is an exponential initial growth rate which is used to characterize how swift the precursor grows into the later crystal, again with the intuition that clusters with a higher asphericitiy may tend to a slower initial growth as theys might be less ordered. For quantifying the initial growth rate, an exponential function has been fitted to the data up to a cluster size of 500 particles.
The representation depending on the cluster size is used, to make the different trajectories comparable, as we expect similar behaviour for similar cluster sizes. Because the cluster size depending on time becomes almost monotonic for cluster size above a few hundred particles, it mostly is a transformation

of the time axis, while the order is only little influenced. Nevertheless it should be kept in mind that this does not constitute a function anymore.
Finally the number of particles, as well as the shape descriptors can span many orders of magnitude making logarithmic scales useful.

A large overview produced by this procedure is given in fig. 3.6.1 for the nucleated trajectories at $\eta = 0.534$ with the three shape descriptors in the vertical direction and the three scalar colouring schemes in the horizontal direction.

A similar approach trying to correlate the three scalar quantities derived for each trajectory have been done, but also did not show any significant pattern.
From the overview we get no obvious sign that there are any correlations between cluster shape and growth rates or between cluster shape and the induction time. Because of that no deeper analysis is done, but instead we conclude that by this superficial analysis we cannot relate the shape descriptors of the cluster to growth or structural properties.

Nevertheless the calculated means give a hint that especially for the anisotropy κ^2 we can see that up to a size of about 1000 particles the clusters become more spherical while at higher particle numbers this tendency towards a sphere comes to an halt. This could be explained for example by the fact that the clusters always exhibit crystal faces leading to some unavoidable asphericity. An other explanation could be that the attachement rate for one crystal face might be higher than for an other. This also would lead for a single crystal to unspherical growth, but as the clusters arre already rather close to a sphere the attachemnt rate does not seem to vary much between the different crystal faces. Also it has been observed that very large single crystals of a few hundred thousand particles may only form at volume fractions of $\eta = 53.2\%$. At higher volume fractions on the other side, domains form as it seems that heterogenous nucleation takes place close to the surface of the cluster, leading to a new crystal orientation in the further growth in this spot.

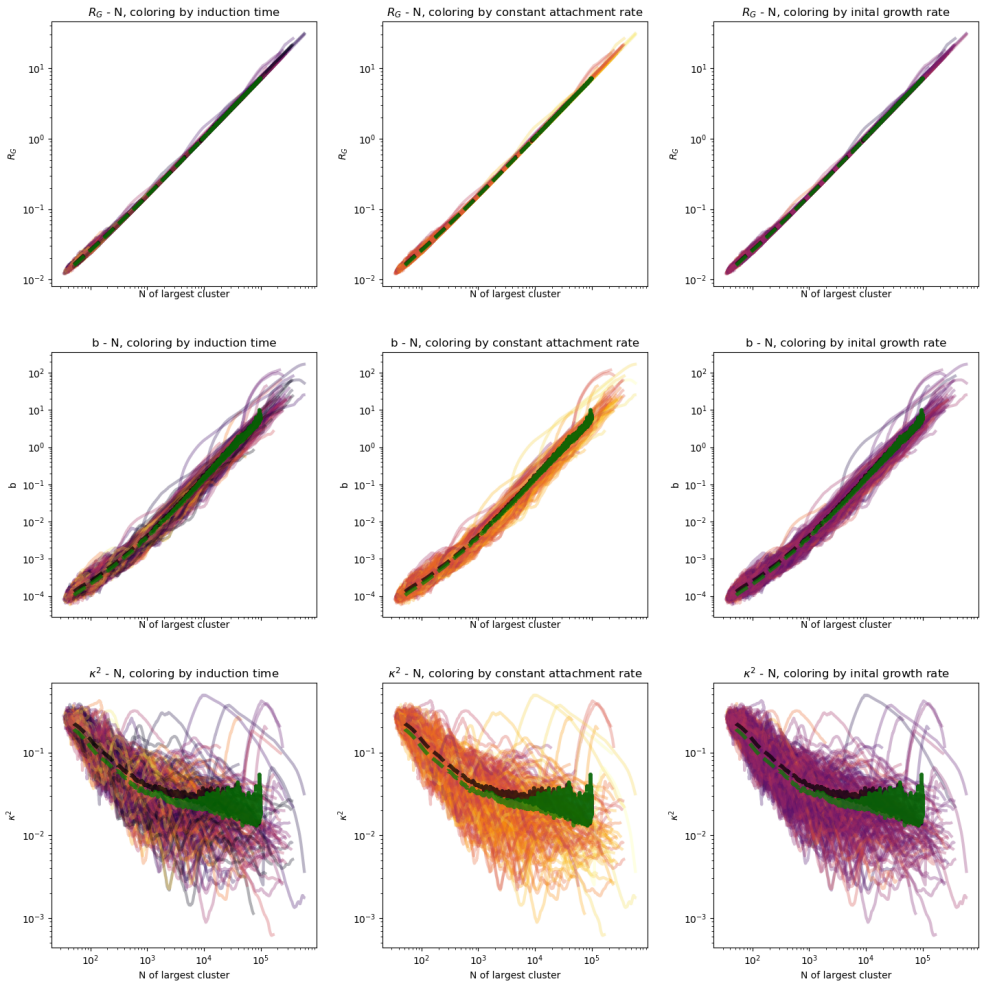


Figure 3.6.1: Overview of the cluster shape describing quantities: Radius of gyration (R_G), asphericity (b) and anisotropy (κ^2), depending on the size of the cluster. The colouring depicts the scalar quantities induction time, constant attachment rate and initial growth rate. Further a smoothed arithmetic mean and median are calculated and depicted.

3.7 Nucleation time dilemma

To calculate induction times or average nucleation times, we will require a definition of when a crystal is called nucleated. This means we have to define from which point a cluster is not merely an unstable fluctuation within the liquid anymore, but instead becomes a stable solid crystalline phase.
In the literature many concepts are used. For example a cluster can be defined as crystalline soon as its of the critical size, calculated by CNT or by doing a committer analysis. An other possibility often used is to rewind the trajectory in which a clearly stable crystal is found, back to the point where the crystal cluster's size more or less vanishes. A further approach is to fit the growth during later times

and extrapolate it to the time when the cluster vanishes.

All these definitions differ more or less only by a delay Δ_τ which is a distribution of times holding the information of how long it takes for varying clusters to pass from the first criterion to the next.
For example we can take as a first point the time when a cluster, known to crystallize at later times, cannot be differentiated anymore from any other structural fluctuation in the liquid, i.e. when the size of the cluster falls below some threshold given by the size of clusters regularly present in a given volume.
The second point we can set by either the critical size of CNT or by some other criterion when we are sure that the cluster has stabilized and will only continue to grow.
At the first of these two points, the fluctuation leading to the crystallization occurs but it would not be possible to tell yet if this precursor melts or continues to grow, while at the second point the crystal is stable. For this reason the first might be called a precursor nucleation and the second crystal nucleation. Between these two points we find the time difference to be the time it takes for the precursors to form a stable crystal. This includes also that some precursors might loiter for awhile before forming the stable phase while other pass this gap rather directly.

When calculating a mean induction time, the delay Δ_τ propagates also to the final result and as it is a stochastic distribution also its higher moments are propagated leading to a smaller precision. This after all only means that the induction time depends on the definition of crystallization and they are only roughly comparable.
In fig. 3.7.1 three distributions with varying definitions of the induction time are visualized.
The three methods explicitly used here are given by the following:

Horizon crossing The time of nucleation is obtained by following the trajectory of the largest cluster within a system after it clearly nucleated back to the point where it was the last time at the average largest cluster of the metastable fluid without stable clusters. The name horizon crossing refers thereby to the idea that fluctuations of the largest cluster in the metastable fluid are more or less independent fluctuations. This is caused by the fact that the large fluctuations of the system alternate at being the largest one, and therefore the largest fluctuations is not bound locally and the fluctuations becomes independent events. On the other side an extraordinary fluctuation will be seen for a longer period of time as the largest cluster, and thus the corresponding fluctuations are not independent in time. The crossing of the trajectory below this horizon where it cannot be followed any longer is meant by the name.

Exponential extrapolation For this method an exponential growth is fitted to the largest cluster data up to N<500. Extrapolating to smaller times makes it possible to evaluate when the exponential crossed 10 particles, which is then taken as the induction time. The method tends to find negative induction times that are not physical.

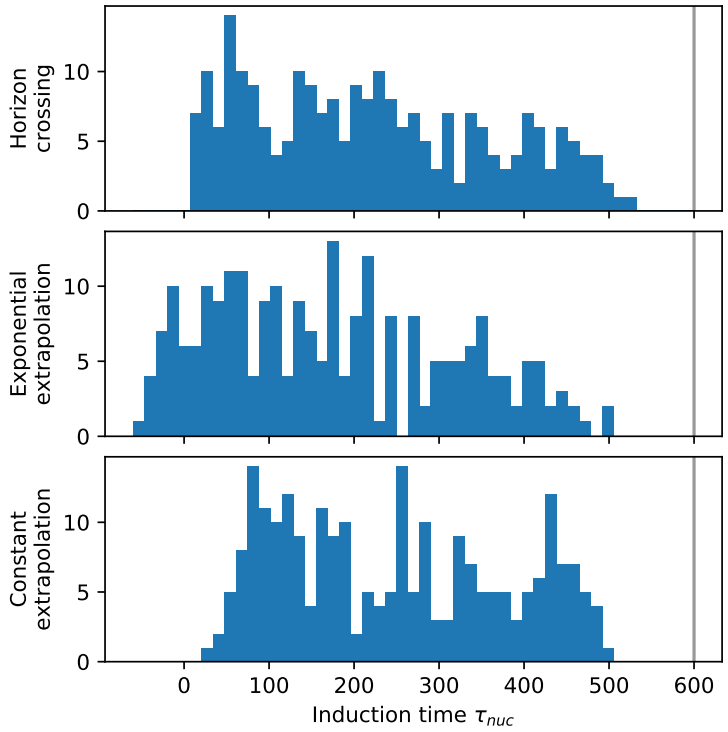


Figure 3.7.1: Induction time distribution obtained by different definitions. While the two methods using extrapolation seem to have the two effects of smearing the signal as well as shifting them, the method of defining the nucleation as the time when the largest cluster is last below the horizon of fluctuations seems to return the most accurate and precise distribution. The final simulation time is marked by the grey line. As clusters require some time to be clearly recognised as crystals no nucleation events are seen towards the end of the simulation interval. To counteract this we will truncate the distribution in the following analysis such that this does not introduce a bias on the final result.

Constant extrapolation The name refers to the constant attachment rate found at later times for the cluster growth. It can be extrapolated to earlier times until the cluster has completely vanished i.e N=0. As the constant attachment rate is higher than the initial growth rate this method returns too large induction times.

As can be seen the horizon crossing method returns a rather smooth distribution that also roughly can be approximated by an exponential decay that is expected for a constant nucleation rate as is shown in section 3.8.1.

3.8 Induction time by exponential distribution

define nucleation times as well as induction times? It seems like people use them as equivalents. Nucleation rates for the meatstable hard sphere fluid have been measured on the experimental as well as on the theoretical side but with a large discrepancy as discussed in section 1.6. The employed procedures and definitions also vary but not to a point to explain the discrepancy so far. The differences mostly originate from the kind of accessible information and system. While the experimentalists often have access to very large systems but without knowing all positions at all times, theorists mostly have smaller systems in numerical simulations but with the advantage of being able to access all particle positions, and in case of simulations probing the dynamics also all velocities.
On the experimental side light scattering and optical methods are mostly employed to measure the structural properties of the probe, comparably on the theoretical side different cluster finding algorithms are used.

While experimentalists may define an induction time by the time interval it takes for a quenched system to reach some level of overall crystallinity, theorists have often used simple approaches like the average time to nucleation for a couple of trajectories to measure their induction times[21] cite it here or not?. This constraints the theorist to wait for all trajectories of an ensemble to show nucleation, what renders it very unsuitable for systems at low volume fractions where the induction time increases steeply.

To circumvent this problem we will define the nucleation rate in the following differently without requiring all simulations to nucleate. In fact we can also show that the uncertainty of the induction time obtained from the data is not significantly reduced anymore for measurements longer than the mean induction time.

3.8.1 CNT expectation of the induction time distribution

In section 1.4 we introduced classical nucleation theory and its constant nucleation rate depending on the barrier height in the free energy landscape. Even if there are signs that CNT is not appropriate for describing nucleation process completly, we will use its prediction of a constant nucleation rate as an assumption to define a constant scalar nucleation rate as well, to compare with other literature values.

As also mentioned before, in the discussion of the system sizes (section 3.1), the induction time of a system depends on the volume under consideration and for this reason the nucleation rate is commonly defined as a nucleation rate density k.

Considering now the simulations we can describe them as a total of m volumes with a size of V_{box}. Further we can define the number of boxes in which a nucleation occurred as $n(t)$ and exclude these from the further simulation.

In this case the total nucleation rate $\dot{n}$ can be written by eq. 3.8.1 from which in the continuous limit of an infinite number of different simulations we can deduce the expected induction rate.

$$\dot{n} = (m - n(t))V_{box}k \tag{3.8.1}$$

$$\Leftrightarrow \frac{\dot{n}}{m} = (1 - \frac{n(t)}{m})V_{box}k \tag{3.8.2}$$

$$\text{in the limit } m \to \infty$$

$$\Leftrightarrow \frac{n(t)}{m} = 1 - \exp\left(-V_{box}kt\right) \tag{3.8.3}$$

$$\text{defining } \tau = (V_{box}k)^{-1}$$

$$\Leftrightarrow \frac{\dot{n}(t)}{m} = \frac{1}{\tau}\exp\left(\frac{-t}{\tau}\right) \tag{3.8.4}$$

The final result in eq. 3.8.4 is the well known stochastic exponential distribution. As the expectation value of the exponential distribution is given by its parameter τ the common approach of using the mean induction time when all simulations have nucleated yields an accurate result and precision can be obtained by taking a large number of simulations.

3.8.2 Maximum likelihood estimator of induction time

In case the simulation time is not accessible we instead will have to deal with truncated exponential distributions. For this we can use Maximum likelihood estimators. The derivation follows Deemer and Votaw 1955 [22].

Maximum likelihood estimators are based on the idea that we can write down the expression of the total probability called likelihood $\mathcal{L}$ for a given set of measurements x_i depending on parameters of the assumed underlying distribution. For the exponential distribution parameterized by the characteristic decay rate κ this is given by eq. 3.8.5.

$$\mathcal{L}(\kappa) = \prod_{i=1}^{N} p(x_i) = \prod_{i=1}^{N} \kappa^N \exp\left(-\kappa x_i\right) \tag{3.8.5}$$

During the process we try to find the maximum of this product. To simplify this product and also

to evade overflow problems on floating point machines, the logarithm of the likelihood is used and maximized yielding the same parameters because the logarithm is a monotonic function and thus does not shift the extrema.
The maximum probability can then be found by usual means of analysis executed in eq. 3.8.6.

$$0 \stackrel{!}{=} \left.\frac{\partial \log(\mathcal{L})}{\partial \kappa}\right|_{\kappa=\hat{\kappa}} \tag{3.8.6}$$

$$\Leftrightarrow \quad 0 = \left.\frac{\partial}{\partial \kappa}\left(N\log(\kappa) - \kappa\sum_{i=1}^{N} t_i\right)\right|_{\kappa=\hat{\kappa}} \tag{3.8.7}$$

$$\Leftrightarrow \quad 0 = \frac{N}{\hat{\kappa}} - \sum_{i=1}^{N} t_i \tag{3.8.8}$$

$$\Leftrightarrow \quad \hat{\kappa}^{-1} = \frac{1}{N}\sum_{i=1}^{N} t_i \tag{3.8.9}$$

By this we have found that the maximum likelihood estimator of κ, for a set of samples drawn from an exponential distribution, is given by the inverse arithmetic mean of the samples. This result is neither new nor surprising but is shown to illustrate how the method of maximum likelihood works. In the following we then show how to handle censored and truncated distributions by the maximum likelihood method.
Both terms in this context refer to sets of samples that are incomplete in the sense that they only include samples up to some threshold $t_i < T$. In the case of truncated distributions the number of samples larger than this threshold is unknown while for the censored distribution the number of samples is known. Taking the example of time consuming nucleation events in computer simulations we are in the case of censored distributions, as the total number of simulation boxes is known but the simulation is stopped at some point when enough nucleations have been collected and hence the number of samples that would have nucleated at later times is known. The probability of an event after the end of the simulation is given by eq. 3.8.10.

$$p(t_i > T) = \int_T^\infty \kappa\exp(-\kappa t)dt = \exp(-\kappa t) \tag{3.8.10}$$

The probability distribution not only below the threshold but also above can then be written as in eq. 3.8.11.

$$f(t) = \begin{cases} \kappa\exp(-\kappa t) & t < T \\ \exp(-\kappa T) & t \geq T \end{cases} \tag{3.8.11}$$

In the simulation we can split up the number of boxes N, into n boxes where a nucleation event was found, and $m = N - n$ others where no nucleation event was spotted during the simulation time T.

Further we have to account for the fact that the samples without distinct times are indistinguishable. This is done by weighting them with the number of possible permutations given by the binomial prefactor $\binom{N}{m}$. The whole expression is then given in eq. 3.8.12 and the extremum of the likelihood function is evaluated in the subsequent reformulation.

$$\mathcal{L}(\kappa) = \binom{N}{m} \kappa^n \ \exp(-\kappa \sum_{i=1}^{n} t_i) \ \exp(-\kappa T)^m \quad \left| \ \frac{\partial \log(...)}{\partial \kappa}\right|_{\kappa=\hat{\kappa}} \tag{3.8.12}$$

$$\Leftrightarrow \quad \log(\mathcal{L}(\kappa)) = \log \binom{N}{m} + n \log(\kappa) - \kappa \sum_{i=1}^{n} t_i - m\kappa T \quad \left| \ \frac{\partial(...)}{\partial \kappa}\right|_{\kappa=\hat{\kappa}} \tag{3.8.13}$$

$$\Leftrightarrow \frac{\partial \log(\mathcal{L}(\kappa))}{\partial \kappa} = \frac{n}{\kappa} - \sum_{i=1}^{n} t_i - mT \quad \Big|_{\kappa=\hat{\kappa}} \tag{3.8.14}$$

$$\Big| \qquad \text{with } \frac{\partial \log(\mathcal{L}(\hat{\kappa}))}{\partial \kappa} \stackrel{!}{=} 0$$

$$\Leftrightarrow \qquad 0 = \frac{n}{\hat{\kappa}} - \sum_{i=1}^{n} t_i - mT \tag{3.8.15}$$

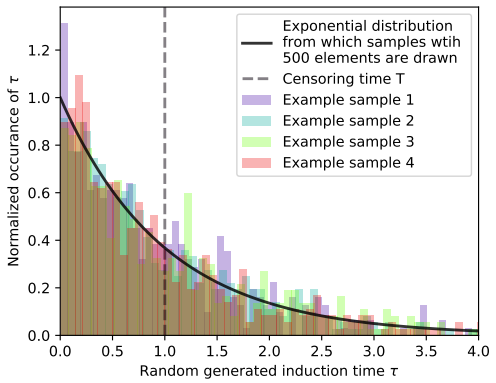
$$\Leftrightarrow \qquad \hat{\kappa}^{-1} = \frac{1}{n} \left(\sum_{i=1}^{n} t_i + mT \right) \tag{3.8.16}$$

The final line eq. 3.8.16 is the estimator of the decay rate of the censored exponential distribution. It is used for the estimation of induction times to compare with other published results in the next sections.

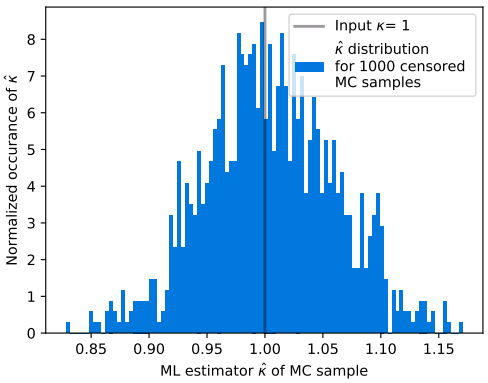
3.8.3 Monte Carlo uncertainty estimation

Having found the estimator the next question is what is its uncertainty, i.e what is the distribution of κ. While corresponding literature on analytic expressions for the distribution exist[23], the complexity becomes inappropriate for the task at hand. Thus we will follow instead a Monte Carlo approach described for example in the book Numerical Recipes[24] to find the uncertainty of the estimator.

For this purpose we draw samples from an exponential distribution characterized by the estimator calculated from the actual simulation data. Afterwards the samples are censored by cutting off all elements larger than T and calculate the corresponding estimator $\hat{\kappa}_{MC}$ for the Monte Carlo sample. From multiple such random sets we can create a histogram of estimates for $\hat{\kappa}$ that can be seen together with some exemplary random samples in fig. 3.8.1. As the distribution seems to incorporate only little higher moments the standard deviation of the distribution is used as the uncertainty $\sigma_{\hat{\kappa}}$.
Concerning the uncertainty in detail we can ask how long a simulation should be to yield precise results. For this we can first look at the case where $1 \gg \kappa T$ corresponding to a simulation where all



(a) Exponentially distributed random samples of size 500 with an exemplary censoring time of $T = \kappa^{-1}$



(b) Distribution of $\hat{\kappa}$ for the previously generated MC samples. The distribution can be described mostly by mean and standard deviation as the number of estimates in the tails are small.

Figure 3.8.1: Exemplary samples for a given κ as well as the distribution of estimates calculated from the random samples. The uncertainty on $\hat{\kappa}$ is approximated by the standard deviation of the distribution from the corresponding Monte Carlo analysis at a given κ.

boxes showed an nucleation event. In this case we have seen before that $\hat{kappa}^{-1} = \frac{1}{N}\sum_{i=1}^{N} t_i$. As we assume that the t_i are exponentially distributed we further know that $\sigma_t = \kappa^{-1}$. The Gaussian error propagation then results in

$$\frac{\sigma_\kappa}{\kappa} = \frac{1}{\sqrt{N}} \,. \tag{3.8.17}$$

Similarly we can take the limit of $1 \ll \kappa T$ which is the case when the mean nucleation time is much larger than the simulation time and therefore only a small fraction of the boxes hosted a nucleation event. In this case we can expand the estimator in the fraction of nucleated trajectories $\frac{n}{N}$ to find $\hat{\kappa} \approx \frac{n}{N}\frac{1}{T}$. In this case the decrease of nucleations events due to a smaller amount of available total volume is not seen yet, and the only information about the nucleation rate is obtained from the number of boxes with nucleations compared to the number of total amount of boxes used. As n is Poisson distributed we know that $\sigma_n = \sqrt{n}$. Fixing N and T and using the expectation value of nucleations $n = N\kappa T$ the Gaussian error propagation for the relative uncertainty is given in eq. 3.8.18.

$$\begin{aligned}\frac{\sigma_\kappa}{\kappa} &= \frac{1}{\kappa}\frac{\sqrt{n}}{NT} \\ &= \frac{\sqrt{N\kappa T}}{NT\kappa} \\ &= \frac{1}{\sqrt{N\kappa T}}\end{aligned} \tag{3.8.18}$$

Finally we are also able to not only look at limits analytically, but also to approximate the relative uncertainty directly by means of the aforementioned Monte Carlo method. For this purpose the same procedure as before is used. The number of elements per sample is consistently with the performed simulations taken to be 500 and to archive good precision the standard deviation of 1000 samples is used for the uncertainty. As can be seen in fig. 3.8.2 the fluctuations between different evaluations becomes rather small, but increase if using a lower number of samples. To compare the analytically derived limits of the uncertainty with the Monte Carlo results both are drawn into fig. 3.8.2.

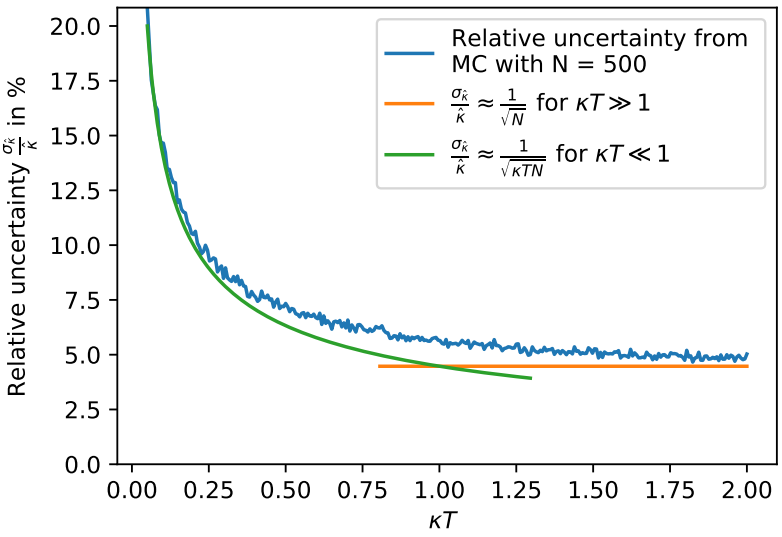


Figure 3.8.2: Relative uncertainty of the ML estimator for varying κT. The x scale is chosen dimensionless such that it indicates the simulation time in comparison to the characteristic nucleation time.

We find that for the limits of $\kappa T \ll 1$ as well as $\kappa T \gg 1$ Monte Carlo results and analytical results are in good accordance while in between the analytical limits only can be used as a rough estimate.

What can be seen from fig. 3.8.2 is that the uncertainty of the estimation drops sharply until about half of the characteristic time, after which it only gains little more precision. This is not surprising as the

information is contained in the nucleation times and rather fast many nucleations have occurred and the long simulation times add only little of further nucleations. Thus simulating until all boxes had an nucleation event is only necessary if one want s to use the simpler arithmetic mean of the induction times as a characteristic time, or if any other constraints make it necessary to reach nucleation of all boxes.

3.9 Nucleation rate comparison

Finally we are able to evaluate the induction time distribution to find the rates given in fig. 3.9.1.

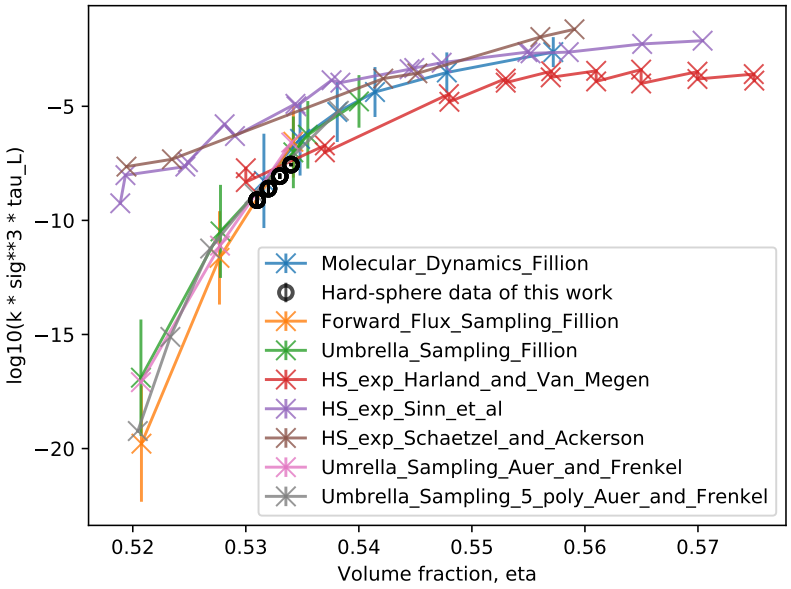


Figure 3.9.1: Some examples of nucleation times in the hard sphere system for varying volume fractions from the literature have been collected, to compare with the own measured data points. All data points have for this purpose been scaled by the self diffusion constant to make comparisons between different experiments possible, as the diffusion sets the timescale at which the system evolves.

From the diagram we can state that our Event driven molecular dynamics simulations confirm the previous simulation results that stood against the experimentally found ones. Further these results are calculated together with their statistical uncertainty, which is mostly visible for the data point at $\eta = 53.1\%$ but is indicated for the others as well but due to the logarithmic scaling almost not visible.
All Nucleation rates that can be found.-> may hap ask Hajo.
-nucleation rates without
-nucleation rates with small particles

3.10 Memory Kernels

Following the approach of Hugues Meyer to calculate memory kernels for an ensemble of trajectories[25], memory kernels for trajectories of about one million particles at volume fractions between $\eta = 53.1\% - 53.4\%$ as well as for a system containing 16384 particles and a volume fraction of $\eta = 54.0\%$ have been calculated. While the first ensemble was not simulated up to the point where almost all boxes where nucleated, and the transition width can be assumed to be large as it takes long simulations for clusters to fill up the large box, the second is chosen to fulfill both objections with parameter given in tab. 3.10.1.

Parameter	Value
N	16384
eq_steps/particle	5000
pr_steps/particle	200000
η_i	45.0 %
η_f	53.4 %

Table 3.10.1: Input parameters of simulations on the NEMO HPC cluster. The large number of production steps is chosen together with the final volume fraction η_f in a way to simulate nucleation and full crystalization of the boxs in almost all cases as can be seen in the top diagram of fig. 3.10.1. Furthermore the small box size leads to a small transition width Δ of about $150\delta t$ corresponding closley to the width of the memory kernel.[26]

Still the memory kernel of the large system has been calculated but except of the Markovian contribution only a slight idea of the memory kernel was visibible, indicating that the sample is not sufficently long or that the largest cluster is not an appropriate observable for nucleation in large systems.

To compare the memory kernel with direct measurements of the observable the evolution of the ensemble is depicted in the top of fig. 3.10.1. The trajectories have been normalized by the number of particles in the box and some statistical properties likes percentiles and arithmetic mean are also shown as the large number of trajectories otherwise makes it hard to distinguish the actual density of lines at some points. At the bottom of the figure the share of trajectories at different stages of the nucleation process is identfied. For this it is assumed that trajectories below an normalized largest cluster of 0.1 can be identified as not nucleated, normlaized trajectories above 0.5 as fully crystallized and all trajectories in between as in the process of filling the box.
As there is no clear analysis yet on how the direct quantities and the memory kernel are related, the differentiation tries to relate intuitively corresponding quantities of the memory kernel and the direct

observable.

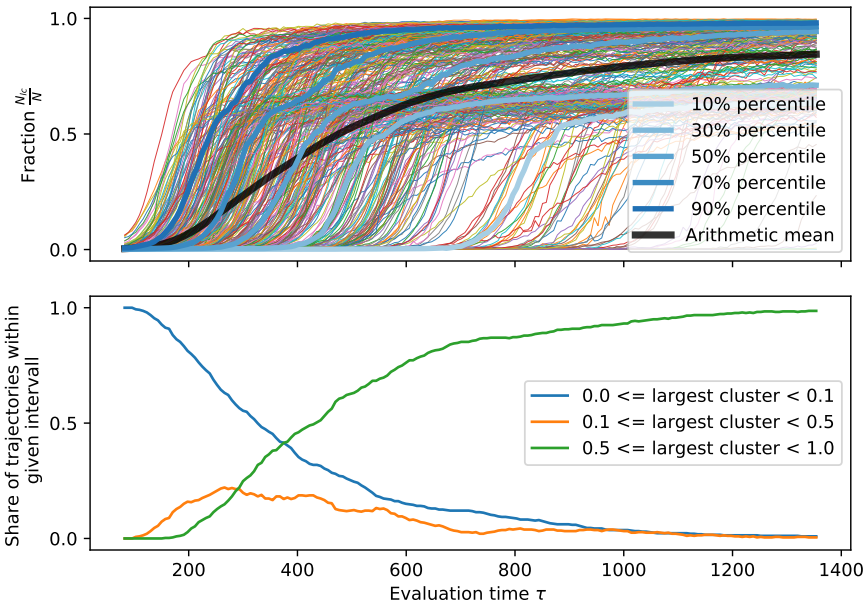
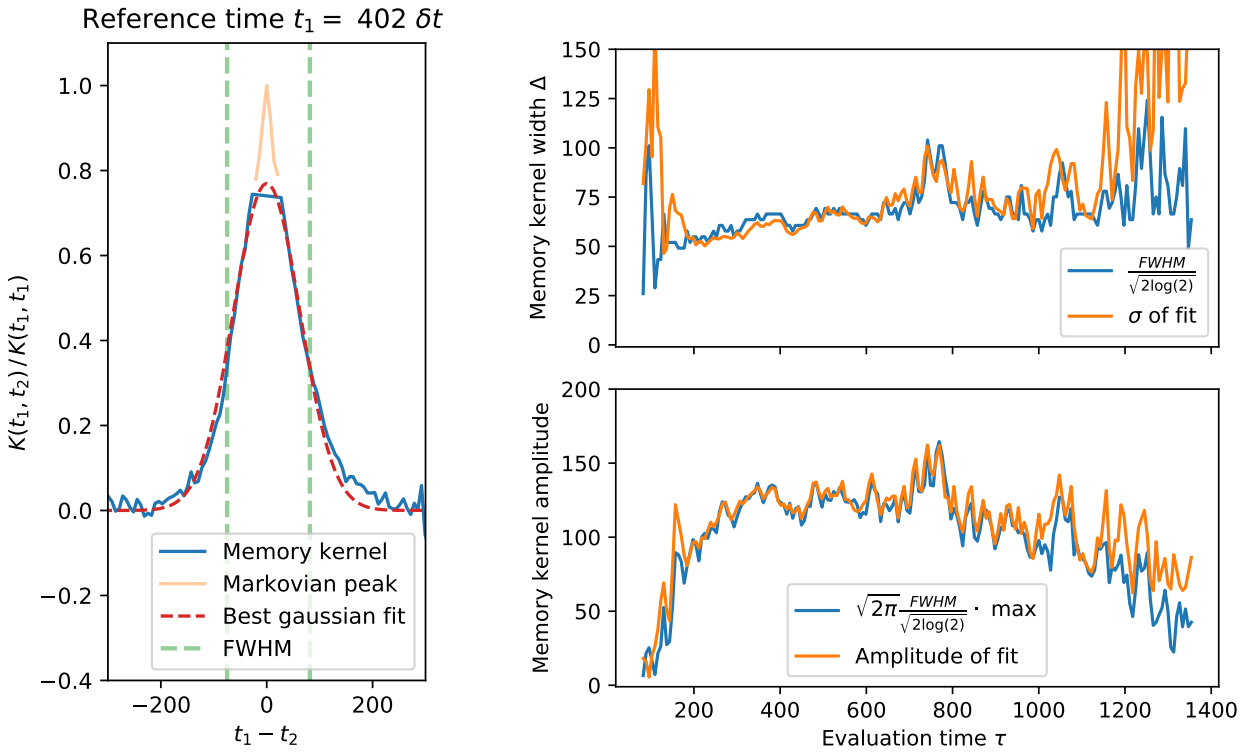


Figure 3.10.1: Top: Normalized trajectories of largest cluster with percentiles and arithmetic mean indicated. It can be observed that some fraction of the trajectories nucleates in more than one step where at first only about 60% of the box is filled by the crystal and at later times they sometimes crystalize further until almost the complete box is filled by the solid phase. From eq. 1.3.6 we would expect a solid fraction of 80% by volume corresponding closely to the expected solid fraction by particles.
Bottom: Fraction of trajectories within intervals chosen to identify nucleated trajectories, momentary growing trajectories and fully nucleated trajectories. As the growth process is much faster than the distribution of nucleations, the orange curve roughly resembles the derivative of the other two curves.

While for the large system only little of the crystallites reached the box boundaries in this latter almost all clusters fill the whole box at the ende of the simulation. As pointed out earlier this finite phase transition time is of the same size as the width of the memory kernel at the mean induction time of the trajectories. As we can see in fig. 3.10.2 the shape of a memory kernel slice at some reference time is rather simple. For this reason we use a Gaussian fit to approximate the width and amplitude of the kernel. For this purpose we neglect the Markovian part of the kernel at around $t_1 - t_2 \approx 0$. To validate the fit results we further use the FWHM, where the maximum is determined by the mean value of the peaks crest.
As the properly normalized result for both methods are in good agreement, we can conclude that the shape of the memory kernel in this case is rather easily defined by a width and an amplitude over time which is depicted in fig. 3.10.2.

As we see the width of the memory kernel sections is more or less constant over the whole measurement with the exception that it becomes very noise at the end.
The amplitude in comparison increases at the beginning, remains over a prolonged period of time constant and then declines towards the end of the measurement.
As pointed out in our article[26] the width of the memory kernel seems to depend on the transition width. As the width in this case is mostly given by the arbitrarily chosen box size it might be possible that we only can see this effect, and otherwise present memory effects are buried beneath. To find these possibly covered memory effects one could generate trajectories with a purely Markovian approach like Brownian dynamics, that corresponds closely to the characteristic properties found for the hard sphere system. Calculating memory kernels from these purely Markovian ensembles would make it possible to compare with the a priori non Markovian ensembles, possibly giving more insight.
An other approach would be to use the committer probability of the largest cluster as an observable, as it would not include a direct system size dependence, that possibly buries other memory effects.



(a) Slice through memory kernel at the given reference time. With the data the excluded Markovian part of the kernel is depicted. Further the full width at half maximum (FWHM) is shown as a first measure of the kernel width as well as a Gaussian fit. The FWHM is normalized to the value of a corresponding Gaussian curve.

(b) Top: Width of the memory kernel slices by FWHM and Gaussian fit. The FWHM is normalized to the value of a corresponding Gaussian curve. Bottom: Amplitude of the memory kernel slice by on the one hand using the mean value of the data around the maximum and on the other by using the amplitude derived from the best Gaussian fit. The amplitude derived from the maximum value is normalized to the value of a corresponding Gaussian curve.

Figure 3.10.2: Example memory kernel together with width and amplitude depending on time.

4 Conclusion - Summary

4.1 Conclusion

5 Appendix

.1 A

Bibliography

[1]B. J. Alder and T. E. Wainwright, "Studies in Molecular Dynamics. I. General Method", The Journal of Chemical Physics **31**, 459–466 (1959).

[2]H. Meyer, "Generalized Langevin Equations and memory effects in non-equilibrium statistical physics", PhD thesis (Université du Luxembourg, Albert-Ludwigs-Universität Freiburg, 2020).

[3]H. Meyer, T. Voigtmann, and T. Schilling, "On the non-stationary generalized Langevin equation", The Journal of Chemical Physics **147**, 214110 (2017).

[4]A. Kuhnhold et al., "Derivation of an exact, nonequilibrium framework for nucleation: Nucleation is a priori neither diffusive nor Markovian", Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics **052140**, 1–7 (2019).

[5]A. Mulero, C. Galán, and F. Cuadros, "Equations of state for hard spheres. A review of accuracy and applications", Physical Chemistry Chemical Physics **3**, 4991–4999 (2001).

[6]N. F. Carnahan and K. E. Starling, "Equation of state for nonattracting rigid spheres", The Journal of Chemical Physics **51**, 635–636 (1969).

[7]N. G. Almarza, "A cluster algorithm for Monte Carlo simulation at constant pressure", Journal of Chemical Physics **130** (2009).

[8]B. R. and D. W., "Kinetische Behandlung der Keimbildung in übersättigten Dämpfen", Annalen der Physik **416**, 719–752 (1935).

[9]M. Bültmann and T. Schilling, "Computation of the solid-liquid interfacial free energy in hard spheres by means of thermodynamic integration", Physical Review E **102**, 1–7 (2020).

[10]P. N. Pusey and W. van Megen, "Phase behaviour of concentrated suspensions of nearly colloidal spheres", Nature **320**, 340–342 (1986).

[11]M. P. Doherty, C. T. Lant, and J. S. Ling, "The physics of hard spheres experiment on MSL-1: Required measurements and instrument performance", 36th AIAA Aerospace Sciences Meeting and Exhibit (1998).

[12]M. N. Bannerman, S. Strobl, A. Formella, and T. Pöschel, "Stable algorithm for event detection in event-driven particle dynamics", Computational Particle Mechanics **1**, 191–198 (2014).

[13]D. Goldberg, *What Every Computer Scientist Should Know About Floating-Point Arithmetic* (1991).

[14]M. N. Bannerman, R. Sargant, and L. Lue, "DynamO: A free O(N) general event-driven molecular dynamics simulator", Journal of Computational Chemistry **32**, 3329–3338 (2011).

[15]A. DONEV, S. TORQUATO, and F. STILLINGER, "Neighbor list collision-driven molecular dynamics simulation for nonspherical hard particles.II. Applications to ellipses and ellipsoids", Journal of Computational Physics **202**, 765–793 (2005).

[16]S. Pieprzyk et al., "Thermodynamic and dynamical properties of the hard sphere system revisited by molecular dynamics simulation", Physical Chemistry Chemical Physics **21**, 6886–6899 (2019).

[17]D. M. Heyes, M. J. Cass, J. G. Powles, and W. A. Evans, "Self-diffusion coefficient of the hard-sphere fluid: System size dependence and empirical correlations", Journal of Physical Chemistry B **111**, 1455–1464 (2007).

[18]J.-P. Hansen and I. R. McDonald, "Theory of Simple Liquids (Third Edition)", (2006).

[19]E. Albert, *Über die von der molekularkinetischen Theorie der Wärme geforderte Bewegung von in ruhenden Flüssigkeiten suspendierten Teilchen*, 1905.

[20]D. N. Theodorou and U. W. Suter, "Shape of Unperturbed Linear Polymers: Polypropylene", Macromolecules **18**, 1206–1214 (1985).

[21]L. Filion, M. Hermes, R. Ni, and M. Dijkstra, "Crystal nucleation of hard spheres using molecular dynamics, umbrella sampling, and forward flux sampling: A comparison of simulation techniques", Journal of Chemical Physics **133** (2010).

[22]W. L. Deemer and D. F. Votaw, "Estimation of Parameters of Truncated or Censored Exponential Distributions", The Annals of Mathematical Statistics **26**, 498–504 (1955).

[23]S. M. Chen and G. K. Bhattacharyya, "Exact confidence bounds for an exponential parameter under hybrid censoring", Communications in Statistics - Theory and Methods **17**, 1857–1870 (1988).

[24]W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in Fortran 77: the Art of Scientific Computing. Second Edition*, 2nd ed. (Cambridge University Press, 1992).

[25]H. Meyer, P. Pelagejcev, and T. Schilling, "Non-Markovian out-of-equilibrium dynamics: A general numerical procedure to construct time-dependent memory kernels for coarse-grained observables", Epl **128** (2019).

[26]H. Meyer, F. Glatzel, W. Wöhler, and T. Schilling, "Evaluation of memory effects at phase transitions and during relaxation processes", Physical Review E **103**, 22102 (2021).