Nucleation and Crystallization of the Metastable Hard Sphere Fluid

by Wilkin Wöhler

Physics - Master Thesis

Albert-Ludwigs University of Freiburg

May 2021

Elaborated within the
Research group for complex systems and soft matter

under the supervision of
Prof. Dr. Tanja Schilling

Ich versichere, dass ich die Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt, sowie Zitate kenntlich gemacht habe.

Freiburg, den ___ . ___ . _____ _____

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 developed 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 large systems enabling also the measurement of cluster growth rates and shape descriptors for clusters with sizes up to a hundred thousand particles without significant spatial influence to itself due to the periodic boundary conditions.

The program is used to simulate 2000 trajectories of systems containing about 1 million particles. In the analysis of the data a constant attachment rate to the cluster's surface is measured. Surprisingly the attachement rate seems unaffected by the packing fraction of the surrounding metastable liquid, even though varying between single clusters by about 50% leading to uncertainties that can not exclude a dependence on the diffusion time scale.

From shape descriptors based on the Tensor of Gyration a 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 completly spherical shape only slowly.

Also nucleation rates at volume fractions of $\eta \in [53.1\%,53.4\%]$ are measured at high precision compared to earlier measurements of these, confirming the discrepancy between real world experiments and numerical simulations. Beyond that the impact memory effects in nucleations for smaller systems is investigated with the finding of a Gaussian memory kernel. The width of which therby is comparable to the width of the phase transition time for a single trajectory, as previously shown by Meyer et al. 2021[1].

Contents

List of Figures

List of Tables

1 Introduction

1.1 The 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 to occupy space, 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 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 even more powerful tool to study phase transitions of simple systems.

The first of such simulations dates back to the beginning of electronic computer technology with first studies by Alder and Wainwright in 1959[2]. 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 Nucleation rate discrepancy and possible memory effects

Nucleation is a process in which a metastable state crosses a first order phase transition and ends in a stable and qualitatively different state. Because such processes are found in many circumstances, like atmosphere physics or metallurgy, people from various subjects have worked on understanding it.

Most descriptions of this phenomenon 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 behavior 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[3]. An extensive list of such approaches can also be found in the introduction by Kuhnbold et al.[4]

include Markovian embedding?

Apart from it, there are other theoretical works that not only tailor CNT to a specific problem but actually are based on more fundamental ideas. These usually 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 the non stationary Generalized Langevin Equation (nsGLE) was derived by Meyer et al. 2017[5]. 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 at the end of thesis 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 for a 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 we integrate over the memory kernel, we can state 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 influence is time dependent. Further we may note that a Markovian kernel only consists of a Dirac delta distribution, as no memory is

included. In this case eq. 1.2.1 is simplified 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.

An other question concerning the hard sphere system is to measure nucleation rates which summarize in a single number how fast the liquid to solid phase transition occurs. This is done 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 metastable fluid

The equation of state for the monodisperse hard sphere system has various parametrizations as for example listed by Mulero et al. 2001[6]. The most common of these, due to its simplicity, is the Carnahan-Starling approximation[7]

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

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

Similarly for the stable solid branch many approximations exist where a common one is given by the Almarza equation of state[8]

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

In it p is the pressure, v is the volume per particle, $v_0 = \sigma^3/\sqrt{2}$ is the volume per particle at close packing and $y = p\sigma^3/(k_B T)$, with k_B being the Boltzmann constant, T the temperature of the crystal and σ the diameter of the spheres.
We may note that the inverse of the volume per particle corresponds to the number of par-

ticles 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 hard sphere system, but it can always be exchanged by the density.

In the system a first order phase transition occurs when switching between the two stable branches, described by the two equations of state, between volume fractions of $\eta_{freeze} = 0.494$ and $\eta_{melt} = 0.55$. The characteristic volume fractions correspond to 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 interval the system tends towards a coexistence state, which in equilibrium varies by 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 where it becomes unstable. The particles then rearrange 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 phase transition. 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.
The equilibration solid fraction of the system $x_s = \frac{V_s}{V}$ with V_s the solid volume and V the total volume can be described by eq. 1.3.6.
For the derivation it is necessary to use that in the stationary coexistance 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

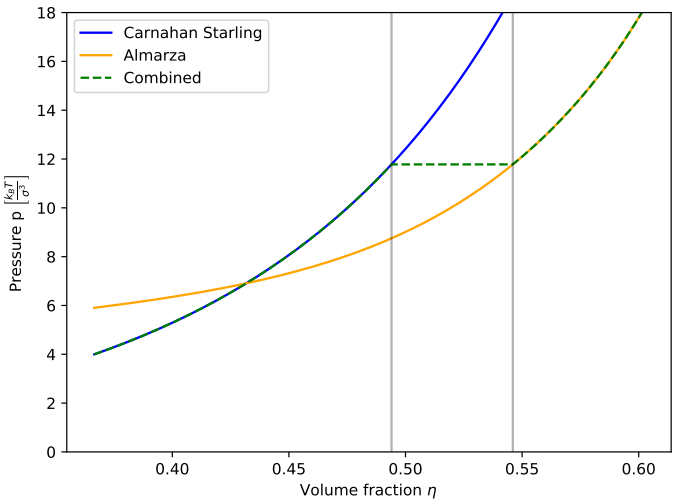


Figure 1.3.1: Phase diagram of the hard sphere system with freezing and melting volume fraction shown as shaded lines and the green dashed line indicating the equilibrium stable branch. Where liquid and solid branch do not coincide with the stable branch, systems are unstable and tend towards a state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

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

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

This leads 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 volume fractions where nucleations are accessible in simulations, 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 behavior 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

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

Evaluating the expression for relevant volume fractions of $\eta \in [53\%, 55\%]$ leads to the conclusion that crystalline fractions of $x_s < 5\%$ only reduce the packing fraction in the fluid by 0.1%. Especially for system sizes of about 1 million particles this 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[9] and since then used and modified multiple times to suit various types of systems. It still provides some reference or expectation, even if its framework does not seem to encompass the full nucleation process, to compare with the simulation data.

The simplest version of CNT assumes that a spherical crystallite of radius R may form in the liquid with properties of the bulk crystal while the fluid remains with the properties of the

bulk fluid. The difference in the free energy landscape is given by a surface and a volume term, each depending on the radius. The first arises from the surface tension γ between the fluid and the solid bulk phase, while the latter is caused by the difference in chemical potential $\Delta\mu$. The whole expression for the free energy is given by

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

with ρ being the particle density of the solid phase.

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

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

Setting the number of particles and the temperature constant and further 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 approximated by the Carnahan-Starling equation of state where we use $\eta = \frac{6\rho}{\pi}$ and $Z = \frac{pV}{NkT} = \frac{p(\rho)}{\rho kT}$. Integrating eq. 1.4.3 between two densities $\rho_{1/2}$ hence is given by

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

with the analytical solution

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

Dropping the lengthy notation for η 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 results which have been calculated before the analytical solution was found. In the following the free energy difference is identified with the difference in chemical potential $\Delta\mu$ as it is the driving force of the nucleation.

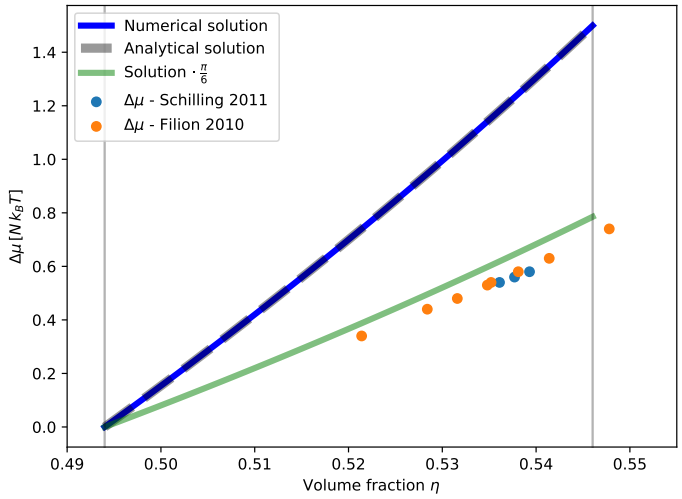


Figure 1.4.1: Free energy difference per particle between the metastable liquid phase and the coexistence phase. Values by Schilling et al. 2011[10] and Filion et al. 2010[11] deviate from the shown result, but we assume that a factor of $\frac{\pi}{6}$ in the calculations is missing in either this or 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 keep growing until the system settles at the equilibrium solid fraction. Here a cluster is defined as a structure having a locally crystalline like ordering. The critical radius, simply calculated by setting the derivative of eq. 1.4.1 to zero, is given by

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

and the height of the barrier at the critical radius is given by

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

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 for this often is given by $\gamma \approx 0.6\,\mathrm{k_B T}\sigma^{-2}$, but its precise value is under debate. Thus we may stick to one of the recently calculated values of $\gamma = 0.589\,\mathrm{k_B T}\sigma^{-2}$ by Bültmann and Schilling 2020[12].

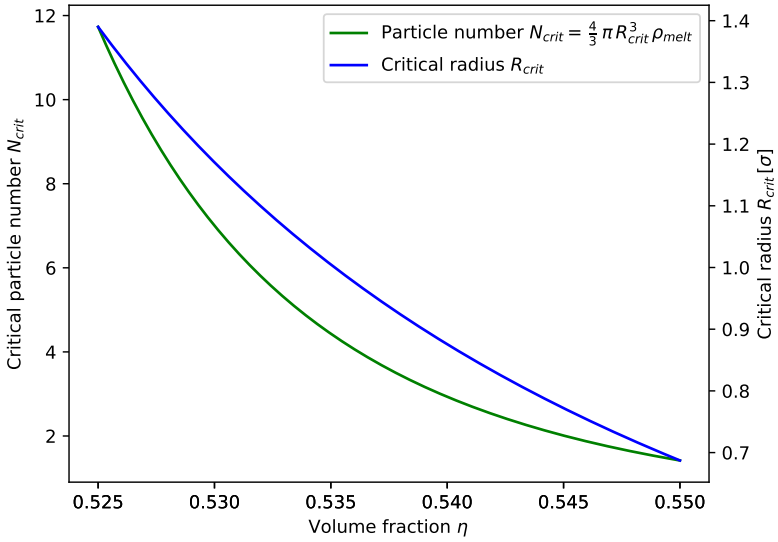


Figure 1.4.2: Critical radius R_{crit} calculated from CNT depending on volume fraction η. The critical radius obtained by our calculation is rather small, but when using the chemical potential calculated by Schilling and Filion the critical cluster size is of the order $N \approx 50$ at intermediate metastable volume fractions, corresponding better to typical fluctuations of the largest cluster found in simulations.

1.5 Computer precision and chaotic behavior

The finite floating point precision impacts the outcome of the simulation as it constitutes a many body problem with chaotic behavior. 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. Also it reminds us that the numerical simulation only is an approximation that never follows the phase space trajectories of the true system.

The exponential growth of induced variations in a chaotic system can be visualized by comparing a reference simulation with a perturbed one. In fig. 1.5.1 the mean of the squared displacements of all particles is recorded 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.

While only observing the left side leads to the assumption that the simulations remain the same to a certain point and then suddenly diverge, in the logarithmic representation we see

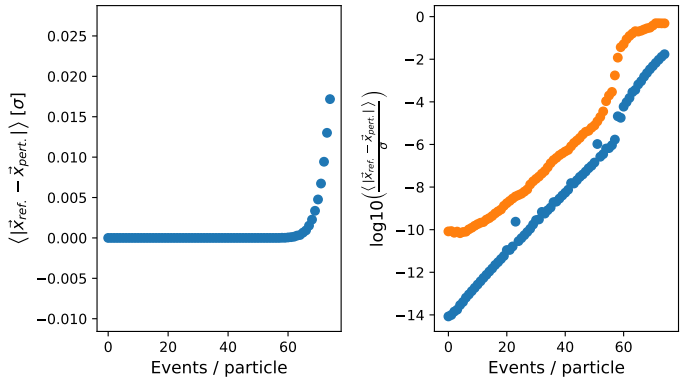


Figure 1.5.1: Mean difference of particle positions in the reference and perturbed simulation. The blue lines show the same data while the orange curve shows the maximum deviation present at each step. For comparison with datasets using the system time δt as units, a rough conversion is given by $T \approx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

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 behavior poses is that any perturbation pushes the system to a completely different trajectory. 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 measurement. Following it, 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 digits. For this reason it becomes necessary to save all precalculated events of the simulation to be able to restart it from a file.
But facing this challenge makes it possible for example to resimulate an 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[13], hard sphere like systems have been synthesized in the labaratory. Today a large variety of such is known, but still further systems are developed to better control stability, sphere size or also to reduce the possible impact of charges on top of the spheres as the Coulomb interaction alters the behavior of the system. Still all of these systems have in common that the hard spheres are suspended in some fluid. Even though nucleation experiments can be done without gravity in space[14], 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 the probe becomes opaque .

The absence of the bath in simple hard sphere simulations constitutes a large difference to those experiments in the laboratory. It has been argued that this only introduces a difference of the time scale which can be compensated by using the characteristic diffusion time as the unit of time, but a discussion on the possibility of hydrodynamic effects changing the behavior of the laboratory system compared to simulations is ongoing at the moment, see Radu and Schilling 2014[15].citation of the contra hydrodynamics? Also the mode spectrum of the suspending fluid within the cavities between the dispersed spheres might have a more important role than expected, but requires further investigation.
Still it usually is not possible to include the suspending fluid into simulations as the proliferation of particles raises calculation times by orders of magnitude.

A further difference is given by the spatial extent and geometry of the simulation. The geometry in simulations is often defined by periodic boundary conditions (PBC) to circumvent surface effects, which 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 major 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, making simple parrallelization of the calculations possible. While the expected behavior of disjunct volumes under the assumption of a constant nucleation rate is discussed in section 3.8.1, the overall solid fraction of a volume in the thermodynamic limit is inspected in the following.

In section 3.5 the cluster growth rate c is found to be mostly independent of the volume fraction in simulations. When extrapolating from the small region in which it was tested, we may approximate the number of particles N at time t for a cluster that emerged at time t_0

by

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

Furthermore approximating the stochastic nucleation events with rate density κ by instead just adding a new cluster after every $\Delta t = (\kappa V)^{-1}$, 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

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

Here 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 interferences are present and heterogeneous nucleation is assumed to be part of the cluster growth process.

With this 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 $x_s = \frac{1}{8}$, which 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, as can be seen in fig. 1.6.1, where the inverse of t^* is calculated from different simulation studies and depicted together with experimentally found nucleation rates.

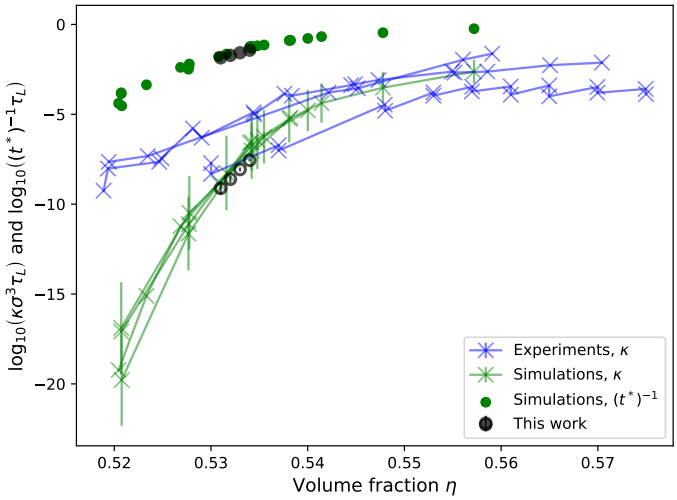


Figure 1.6.1: Diagram with modified nucleation rates calculated by eq. 1.6.3. The experiment and simulation data sets are the same as in fig. 3.9.1 where for each set the literature reference is noted.

What can be seen on the one hand is that the simulated nucleation rates steeply decrease for lower volume fractions in contrast to the experimentally found ones. On the other hand the gradient of $(t^*)^{-1}$, calculated from the simulation results, follows mostly the one of the experimental results, as the fourth root changes the slope by a factor of 4 in the logarithmic diagram. The rates still deviate by a factor of about 10000 which still requires explanation, but one possibility could be that experimentalists always use polydisperse systems which are known to nucleate slower.

While this is not a proof it might be a hint that laboratory based experiments measure more cluster growth than actual nucleation events. It has to be discussed with experimentalists if the assumptions leading to this result hold under close inspection or if measures against this behavior have been taken.

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 the actual dynamics of the system ais required to search for possible memory effects. This means that simulations only probing the phase space of the system, 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 equations of motion for the particles are numerically integrated. As the EDMD approach even requires these discontinuitie it is very well suited for the purpose at hand.
The key points of the program, possible extensions and a thorough testing are presented in the following, starting with the units of the simulation, which are also used throughout this thesis. They are σ the sphere diameter as the unit of length, m the mass of a particle as the unit of mass, k_BT as the unit of energy and resulting from these $\delta t = \sqrt{m/(k_Bt)}\,\sigma$ as the unit of time.

2.1 Algorithm and simulation details

In this section we will highlight the main differences to regular MD simulation schemes, which are an other main tool to probe the dynamics of molecular systems. Furthermore we will stick to the hard sphere example when discussing the EDMD simulation, but it can be kept in mind that the approach allows to simulate all systems governed purely 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 accordingly to the next time step, EDMD simulations do not use a predefined time step but instead the system is evolved from one event to the next one, where the next event is defined by the next collision of two particles within the simulation box.

The employed event prediction algorithm follows closely the approach proposed by Bannerman et al.[16] 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.

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

$$\begin{aligned} &\text{with} \quad \vec{r}_i(t) = \vec{r}_i(t_0) + (t - t_0)\ \vec{v}_i(t_0)\,, \\ &\qquad \Delta t \coloneqq t - t_0\,, \\ &\qquad \vec{v}_{ij}(t) \coloneqq \vec{v}_j(t) - \vec{v}_i(t)\,, \\ &\qquad \vec{r}_{ij}(t) \coloneqq \vec{r}_j(t) - \vec{r}_i(t)\,, \\ &\Leftrightarrow \quad \vec{r}_{ij}(t) = \vec{r}_{ij}(t_0) + \Delta t\ \vec{v}_{ij}(t_0) \end{aligned} \tag{2.1.2}$$

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

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

The overlap function has the property that it is negative for two particles being closer than their diameter, zero 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 given in eq. 2.1.5.

$$\begin{aligned} & 0 = rr + vv\ \Delta t^2 - 2rv\ \Delta t - \sigma^2 \\ \Leftrightarrow \quad & 0 = \Delta t^2 - \frac{2rv}{vv}\ \Delta t + \frac{rr - \sigma^2}{vv} \\ \Leftrightarrow \quad & \Delta t = -\frac{rv}{vv} \pm \sqrt{\left(\frac{rv}{vv}\right)^2 - \frac{rr - \sigma^2}{vv}} \end{aligned} \tag{2.1.5}$$

But a caveat when executing on a floating point machine is present as can be seen when considering which solution is the relevant one. 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

$$\Delta t = \frac{-rv - \sqrt{(rv)^2 - vv(rr - \sigma^2)}}{vv} \,. \tag{2.1.6}$$

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 for this as they tend to large inaccuracy in this case. But rewriting eq. 2.1.6 by making use of the third binomial formula leads to the mathematically identical expression

$$\Delta t = \frac{(rr - \sigma^2)}{-rv + \sqrt{(rv)^2 - vv(rr - \sigma^2)}} \,. \tag{2.1.7}$$

Comparably eq. 2.1.7 does not contain a cancellation of the type seen before and hence is better suited for the use in the computer simulation as stated by Goldberg 1991[17].

The event prediction algorithm proposed by Bannerman et al.[16] 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 true the particles collide and Δt is calculated by eq. 2.1.7.

All possible collision times for a particle are then stored in a queue that is sorted by the event times 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.

While this is the simplest description in section 2.1.2 the implementation of some widely used measures to reduce redundant calculations and the use of a cell system to reach $\mathcal{O}(N)$ computation time are further discussed.

An important 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 concerning 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 program. Furthermore differences to the MC program are mentioned for the readers that are familiar with it.

The *Event* struct

We start with the basic *Event* struct which includes 6 entries that are shown in tab. 2.1.1. The

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

Table 2.1.1: Content of the *Event* struct.

time variable holds the time for when the event is scheduled in the future. 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 defined as 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' neighbors 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.

The *Particle* class

The *Particle* class is comparable to the one from the MC code basis. Its MC related attributes 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 linearly with more backup events while the speedup does not increase much for more than two stored events Bannerman et al. 2011[18]. It also has been argued that the added complexity can not account for the increase in efficiency Donev et al. 2005[19].
However in our own simulations a calculation time reduction 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 neighboring 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 only constituting less than 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 as transfers are more frequent.

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.
To take the whole configuration to one point of time, the *transferToTime()* function of the particle is provided. This is obviously necessary for measurements 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 the 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 disturb-

ing 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 remained in principle the same as in the previous MC code. A new element is the *neighbors_lookup* table, which contains the indices to the elements in the cells' array *neighbors*, where pointers to the cells that share a surface are stored. 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.

We may also note the importance to preallocate all arrays that are used in the event calculations for the efficiency, because the collision prediction routine is executed about 30 times per step and particle, 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. Denoting for each dimension i, the position of a particle within its cell by r_i, its velocity by v_i, and the cell's length by l_i, we can write for each dimension the equations

$$t_{i1} = -\frac{r_i}{v_i} \quad \text{and} \quad t_{i2} = \frac{l_i - r_i}{v_i} \,, \tag{2.1.8}$$

which describe the times $t_{i\,1/2}$ when the particle pierces the cell's left and right boundaries in dimension i. 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' *neighbors_lookup* table. The index lies between zero and five for to the six possible neighbor cells sharing a surface with the current cell of the particle. Each valid case

represents a distinct neighbor cell and its index within the cell's *neighbors* array is clearly defined by the cell setup routines. The indices within the neighbors 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' *neighbors* 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.

While the here presented algorithm is only valid for single sized particles it can be extended to polydisperse systems as is shown in section 2.5.1.

As this routine is executed throughout 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 instead. 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 particle's PEL is then used. 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 particle's *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 the address within the lookup table of the box where the index of the pointer to the destination cell, in the departure cell's neighbors array, is deposited.

Scheduler::executeCollision()

The velocity change after a collision between particles i and j of same mass, with corresponding velocities $\vec{v}_{i/j}$ and relative position $\vec{r}_{ij} = \vec{r}_j - \vec{r}_i$ is given by

$$\vec{v}_i' = \vec{v}_i + \left(\frac{rv}{rr}\right)\vec{r}_{ij} \ , \tag{2.1.9}$$

where the definitions from eq. 2.1.5 of the scalar products, rr and rv, for the relative positions and velocities are used.
Demtröder and Bergmann-Schäfer both only give it in the center of mass frame. With the same work of taking it from a book and transforming it, i can just as well write down the derivation of it in the laboratory frame in the appendix...

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 its 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 events in the backup are still valid and the first entry is pushed into the FEL. In case the *backupEvents* array is empty, 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 by the case before.

Furthermore the routine counts the occurance 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 The simulation periphery

For the simulation to work also some more surrounding is required, which is briefly discussed in the following.

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.

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. So far only the *FCC_init()* routine is written, which initially places all particles on a fcc lattice and assigns the amplitude of their starting velocities by the equipartition theorem to $|\vec{v}| = \sqrt{3}\frac{\sigma}{\delta t}$. The directions are chosen at random, under the constraint to keep the center of mass at rest.

Tools

This is the toolbox of the simulation holding routines to measure quantities like mean squared displacement, radial distribution functions and the cluster finding routine. Also functions used within the simulation like an overlap or minimal distance routine for the compression are included at this point.

The cluster finding algorithm will be highligthed at this point, as the details of it are necessary to compare the direct data of cluster sizes with the data of other groups.
It is based on the q6q6-bond-order parameters first described by Steinhardt et al.[20]. The local structure around a particle i with N_b neigbours is characterized by the quantity

$$\bar{q}_{lm}(i) = \frac{1}{N_b} \sum_{j=1}^{N_b(i)} Y_{lm}(\hat{r}_{ij}) \ , \tag{2.1.10}$$

where $Y_{lm}(\hat{r}_{ij})$ are the spherical harmonics evaluated in the direction of the relative position of particles i and j in a given coordinate system.

$\bar{q}_{6m}(i)$ suffices to indicate the local fcc structure of hard-sphere crystals. Based on $\bar{q}_{6m}(i)$ a normalized vector $\vec{q}_6(i)$ is defined with elements for $m = -6$ to $m = 6$ given by

$$q_{6m}(i) = \frac{\bar{q}_{6m}(i)}{\sqrt{\sum_{m'=-6}^{6} |\bar{q}_{6m'}(i))|}} \ . \tag{2.1.11}$$

As a minimum threshold for the scalar product $\vec{q}_6(i) \cdot \vec{q}_6(j)$ we choose 0.6 to identify a pair of particles i and j as "orientationally bonded". To define a solid particle we set the minimum number of bonded neighbours to be $n_B > 8$, similar to ten Wolde et al.[21] or Schilling et al.[10].

2.2 Testing of the simulation code

To verify the dynamics of the simulation, we measure the long time diffusion constant and the radial distribution function of the stable hard sphere liquid, as there are measurements and theoretical predictions in the literature to compare with.

2.2.1 Diffusive behavior

The diffusive behavior of particles in a liquid usually can be seperated in three distinct regimes. 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 subdiffusive phase in which the particles are repelled for the first time by their nearest neighbors. And third, the long time diffusion to describe the random propagation of the particles through the fluid over time.

As the ballistic hard sphere system enters into the long time diffusion almost at once, due to the missing of the suspending fluid, only this is measured. By the definition of a diffusive process the average mean squared displacement (MSD) of a particle is described by

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

where $\langle x^2 \rangle$ is the expectation value of the MSD, d the number of spatial dimensions, D_L the long time diffusion constant and t the system time. With this relation we can use the measurement of $\langle x^2 \rangle(t)$ and take its linear regression to find the diffusion constant D_L.
The used testing systems are characterized in tab. 2.2.1. The equilibration phases have been carried out at the final volume fractions up to $\eta = 50\%$ 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 measurements 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.

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.

The resulting diffusion constants depending on the volume fractions are shown in fig. 2.2.1

alongside with values from the literature for the same hard sphere fluid.

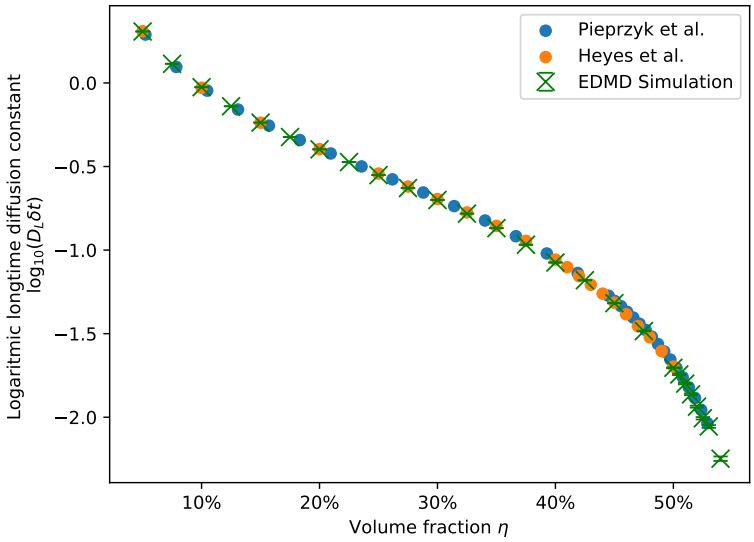


Figure 2.2.1: Logarithmic plot of long time diffusion constant for the hard sphere liquid as measured in our own simulations as well as measurements from Pieprzyk et al. 2019[22] and Heyes et al. 2007[23].

As it can be seen the EDMD simulation is very well capable of reproducing the diffusion constant for the hard sphere liquid and therefore we expect the dynamics of it to accurately approximate the purely ballistic hard sphere system.

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. As expected no particles come closer than the diameter of a sphere, verifying that no collisions are missed. Further for higher volume fractions the liquid shells become very well visible. At very high volume fractions we also find that a new peak becomes visible at $r < 2\sigma$, indicating the local structuring on the path to nucleation.
To compare with the Percus-Yevick approximation, the radial distribution functions for two volume fractions are shown with the corresponding theoretical solution in fig. 2.2.3.
As highlighted for example by Hansen and McDonald 2006[24], the theoretical approximation has some flaws as can be seen with $g(r)|_{r=1\sigma}$ being too low for the Percus-Yevick approxima-

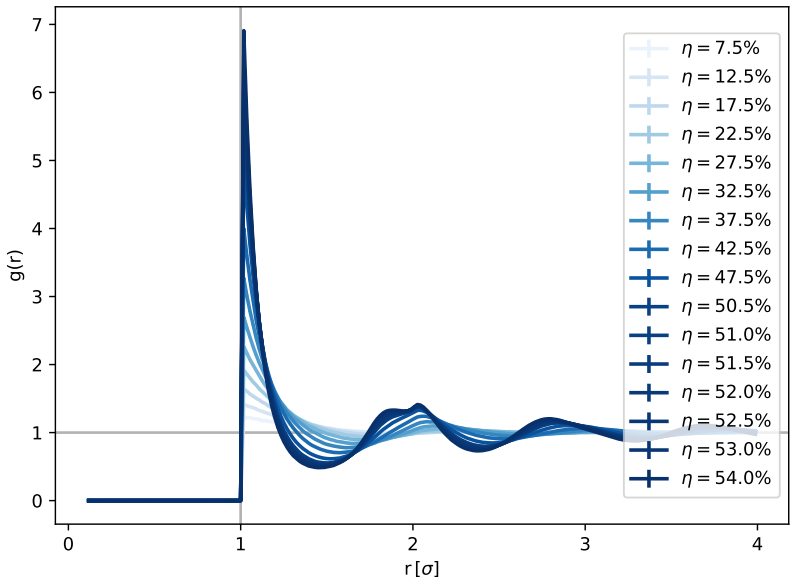


Figure 2.2.2: Radial distribution functions for a range of volume fractions, with color lightness corresponding to the volume fraction of the liquid.

tion. 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.

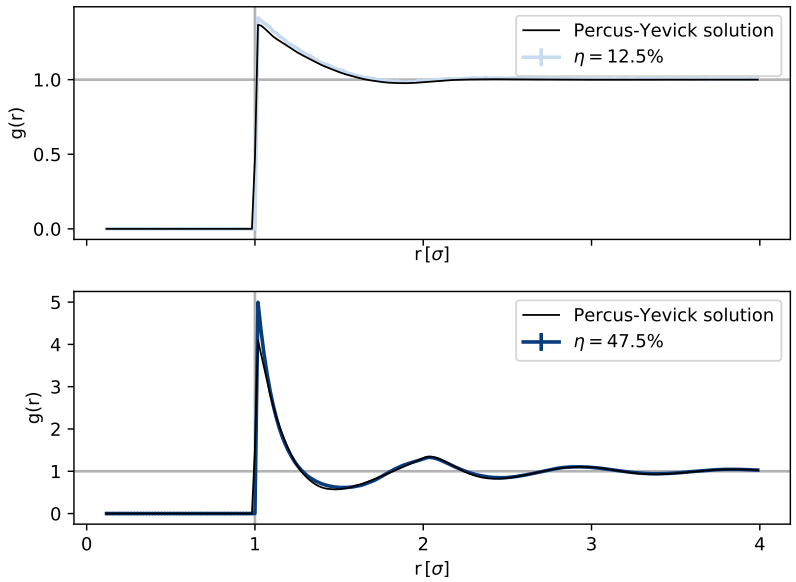


Figure 2.2.3: Radial distribution function of the particles in the hard sphere liquid at a low and at a high volume fraction, together with the theoretical Percus-Yevick approximation.

2.3 Estimate of required resources

To choose system parameters reasonable, calculation times and file sizes of the simulation have been characterized. This was needed 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 estimate

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 in the fluid state. 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.

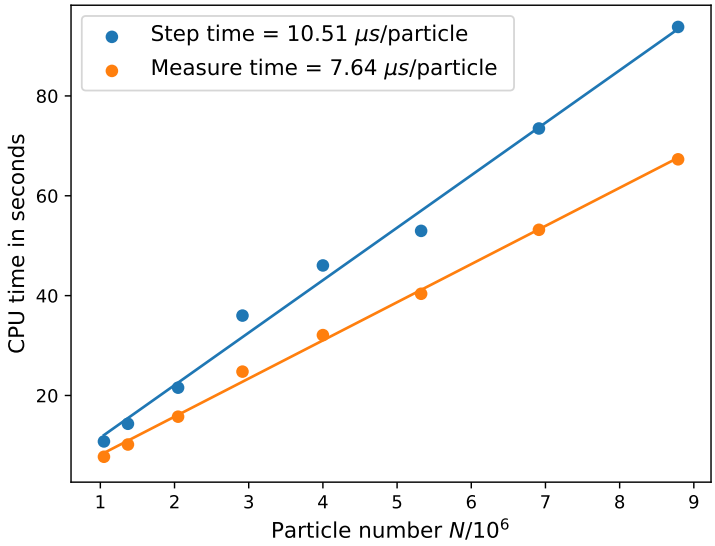


Figure 2.3.1: Overview of CPU time required for calculating a simulation step, consisting of an event for each particle, as well as a measurement of relevant quantities of the system. As assumed for a simulation algorithm with a theoretical $\mathcal{O}(N)$ calculation effort, the data points can be well described by a line. 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's policy and EDMD simulation program did not result in a save shutdown of the program after breaching the wall time limit of four days.

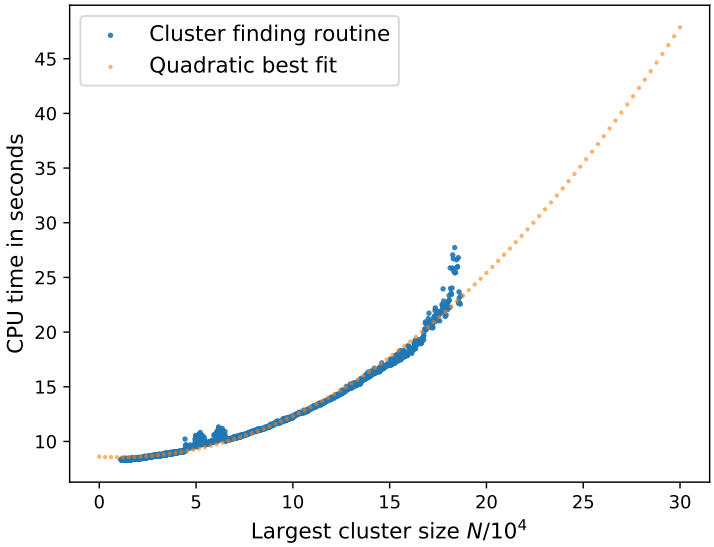


Figure 2.3.2: Calculation time of the q6q6 order parameter cluster finding routine with an increasing largest cluster size during one nucleation. The quadratic best fit indicates that the calculation effort can be approximated by $\mathcal{O}({N_{\mathrm{lc}}}^2)$, with N_{lc} being the size of the largest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

2.3.2 File sizes estimate

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 configuration 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 checkpoint which contains all positions, velocities, the FEL, all PEL's, all delayed times, all cell's first particles and properties of the box, are shown in fig. 2.3.3. It can be seen that the file size is proportional to the system size as each particle adds a pair 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 can only amount to substantial file sizes if measurements at about each step for long simulations are done.

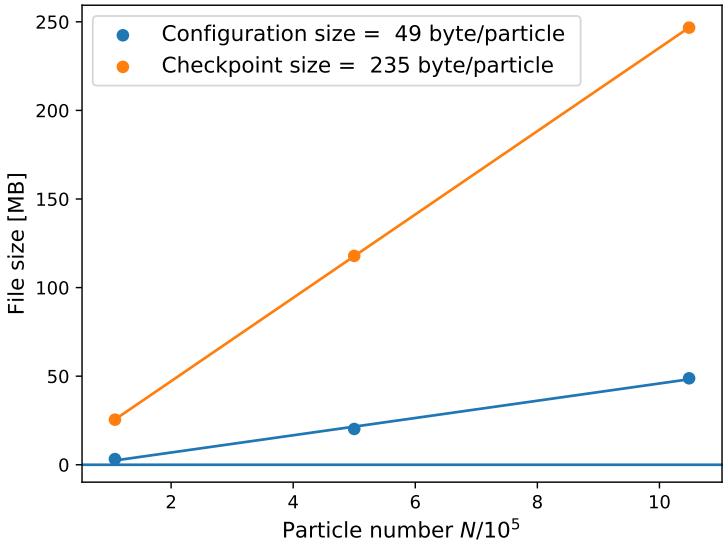


Figure 2.3.3: File sizes of a configuration snapshot and a full simulation state at different system sizes with their linear regressions. While for 3 points these are not statistically meaningful, they still remain a useful tool to extract the slope that corresponds to the required memory per particle for a configuration snapshot or a simulation checkpoint.

2.4 Preliminary data for equilibration test

The motivation to develope 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 the stable fluid phase becomes metastable . If this state is evolved in time, nucleations can be observed as stochastic distributed events. To measure those without artifacts originating from the simulation procedures, some parameters were tested within reasonable ranges prior to the data production.
Mainly the number of equilibration steps as well as the initial density before the volume quench seem likely to influence the measurement as both may impact the local ordering directly after the quench. 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 parameters given in tab. 2.4.1.

The general behavior of the fluid is analyzed by inspecting the cluster size distribution over time. Its average over all trajectories is shown in fig. 2.4.1 together with the same data smoothed by a Gaussian filter matrix. The smoothing is employed afterwards because for low

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.

count rates only flucutations are visible in the differences between the cluster size distributions that are compared.

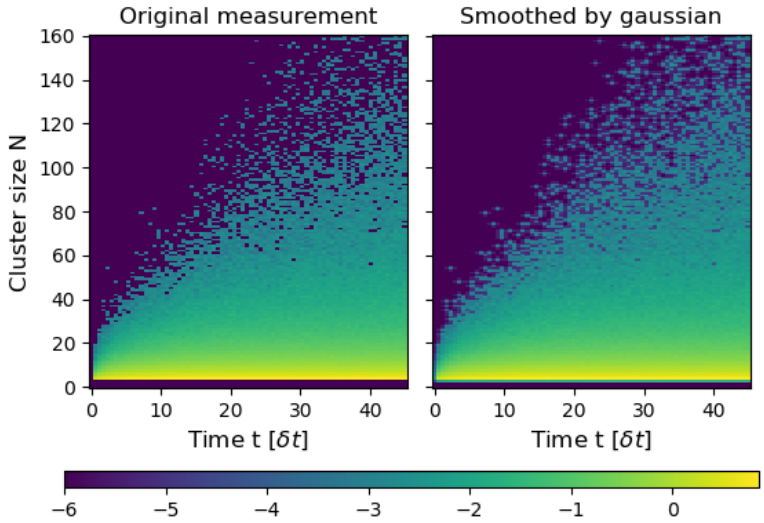


Figure 2.4.1: Heat map of the mean cluster distribution over time. The diagram encompasses 800 trajectories of 16384 particles each. The coloring indicates the decadic logarithm of the average cluster occurrence in a box, corresponding to a free energy in the stationary case.

From fig. 2.4.1 we can see the reaction of the fluid after the quench into the metastable state. As there are rarely any clusters present in the stable liquid and the spatial configuration of the particles requires some time to rearrange into the local ordering, no clusters are found directly after the quench. In the later evolution we then see how clusters form, and soon after begin to nucleate leaving the y-axis range of the diagram.

To identify differences between the ensembles with varying start parameters, the quantity defined in eq. 2.4.1 is used. To circumvent complications due to zero values, they are fixed to values below the regular signal. Three samples of this comparison are shown in fig. 2.4.2

and fig. 2.4.3 for different lengths of equilibration phase and different initial densities. The coloring indicates the value of $\Delta_{p(N,t)}$ defined in eq. 2.4.1. As mentioned before 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 mostly fluctuations would be visible.

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

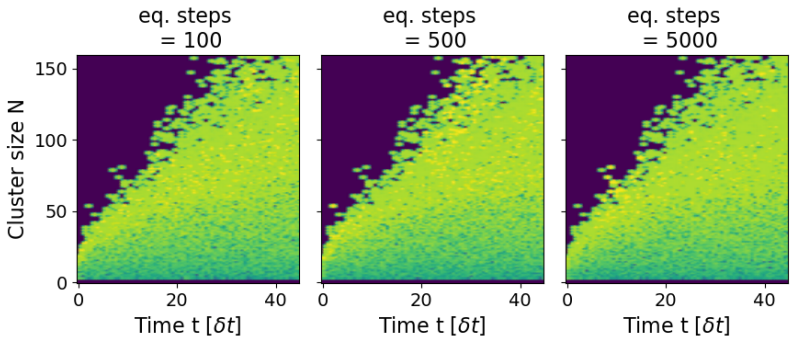


Figure 2.4.2: Heat map of differences between the cluster distributions within simulations carried out with varying the length of the equilibration phase. The quantity used for coloring is defined in eq. 2.4.1, where yellow indicates a large difference while blue indicates a small difference. Providing a legend of the coloring 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 behavior. 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 shining through. Actual differences not due to fluctuations would only be visible within the green and yellow non-zero region, but no such difference is observed.

While it seems like the set for an initial volume fraction of $\eta = 0.4$ and eq. steps = 5000 includes a little less irregular fluctuations, strong differences remain absent. Especially interesting is the ensemble with eq. steps = 100 because here the length of the equilibration phase is similar to the time the initial perfect crystal configuration takes to melt. For this reason one could expect that a significant part of these configurations might directly crystallize again, but instead we do not find any significant impact.

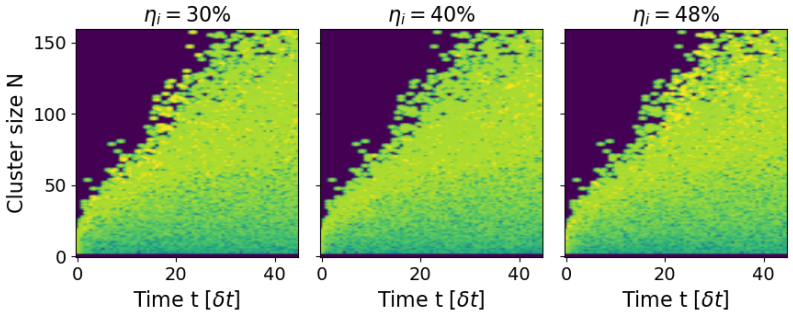


Figure 2.4.3: Heat map of differences between the cluster distributions within simulations carried out with varying the volume fraction of the liquid during the equilibration phase. The quantity used for coloring is defined in eq. 2.4.1, where yellow indicates a large difference while blue indicates a small difference. Providing a legend of the coloring 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 quantitative analysis is given by calculating the mean nucleation rates for each set of trajectories. The maximum likelihood estimator that is used for this purpose as well as its uncertainty is discussed in section 3.8.2. The results for the different data sets are depicted in fig. 2.4.4.

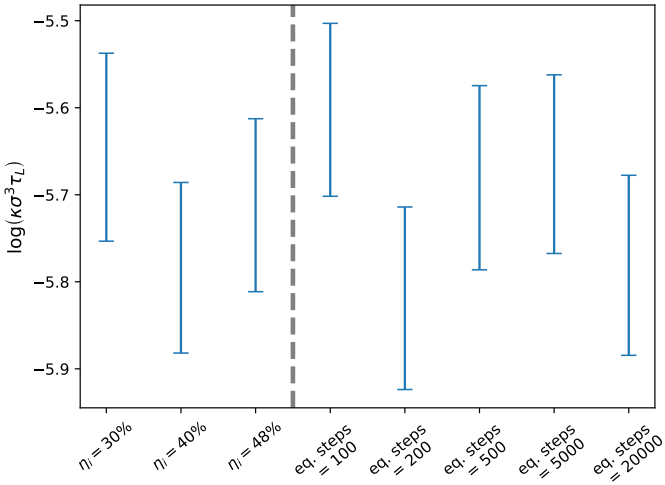


Figure 2.4.4: Comparison of nucleation rates under CNT assumptions for different initial densities during equilibration with eq. steps fixed at 5000, as well as varying eq. steps with η_i fixed at 45%.

As we see, no significant difference can be observed between the nucleation rates, 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 can conclude in this chapter that as long as parameters are set within broadly but reasonable boundaries, we expect not to have systematic influences of simulation parameters.

2.5 Extensions for future studies

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, additional features have been developed to suit the code for further studies. Polydispersity in the sense of radius and mass distributions have been implemented and roughly tested, as well as individual cluster tracking, to enable a detailed study of spatial information regarding the clusters. The state of these two features and their use are described in the following two sections.

2.5.1 Polydispersity for varying radius and mass

Polydispersity has been included in the simulation to make comparison with the real world simpler, as in actual experiments monodisperse spheres are practically not achieved. Also the phase diagram becomes richer as the complexity of the system increases as is shown for example by Pusey et al. 2009[25].
Within the implementation mostly the prediction of collisions has to be adjusted. When looking at the derivation of eq. 2.1.7 it is found that σ being the former diameter of a sphere in the monodisperse case only has to be changed to $\sigma = R_i + R_j$. In eq. 2.5.1 and eq. 2.5.2 the same definitions of scalar products are used as before in section 2.1.1 where the monodispere case is discussed.

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

Then 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.

$$\vec{v_i}' = \vec{v_i} + \frac{2m_j\,(rv)}{(m_i+m_j)(R_i+R_j)^2}\cdot\vec{r}_ij$$
$$\vec{v_j}' = \vec{v_j} + \frac{2m_i\,(rv)}{(m_i+m_j)(R_i+R_j)^2}\cdot\vec{r}_ij \tag{2.5.2}$$

A small caveat is given by the fact that the system with different masses requries a new routine to fix its center of mass frame as otherwise unnecessary transition events have to be calculated.

2.5.2 Single cluster tracking algorithm

Following trajectories of single metastable clusters within the fluid can be used to measure their mean lifetimes. Also the nucleation time can be observed 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. The latter one is used as it is easier comparable and accessible in our case because a routine to calculate the center of mass of a cluster already is implemented and less data has to be written and compared. An algorithm based on maximum movement from one time step to the other is tested and 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 mostly vary by the scaling. 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 very large fluctuations with short lifetimes might be caused by merging and splitting of clusteres as the simple algorithm does not test for such. The overall impression is that the fluctuation distribution is compact with a small but very far reaching tail towards the large lifetimes as well as towards the large cluster sizes.

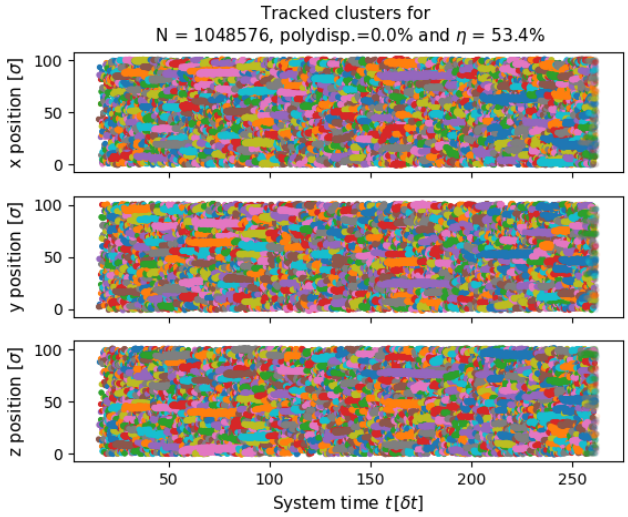


Figure 2.5.1: Example results of the cluster tracking algorithm in a monodisperse simulation. The three plots are the projections of the box onto the three spatial dimensions over time. Each cluster is given a color to identify it. With it we can see for example that two clusters mingling in one projection are actually some distance apart from each other in an other dimension.
In this example only small metastable clusters that dissolve after some time are present but also nucleation events can be visualized in this kind of plot. These are easily identified as the linewidht is proportional to the diameter of a sphere with a volume corresponding to the clusters volume under the assumption of it being spherical symmetric.

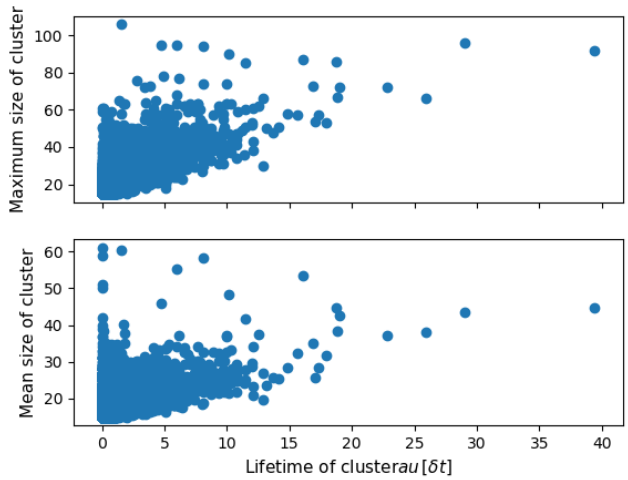


Figure 2.5.2: Example of lifetime depending on maximum (top) or mean (bottom) size of the metastable cluster. The size is defined by the number of particles in the cluster.

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 (HPC) 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 only rarely without nucleation event in the case of early breakdown.

The simulations comprise four series' at volume fractions of $\eta = 0.531$, 0.532, 0.533 and 0.534 where each series 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 to probe nucleation rates 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 their 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, but using CNT

as a guideline it can be shown that the computational effort for simulating nucleation events does not increase significantly with increasing system size. As 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, as the initial induction time would be 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 memory 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.

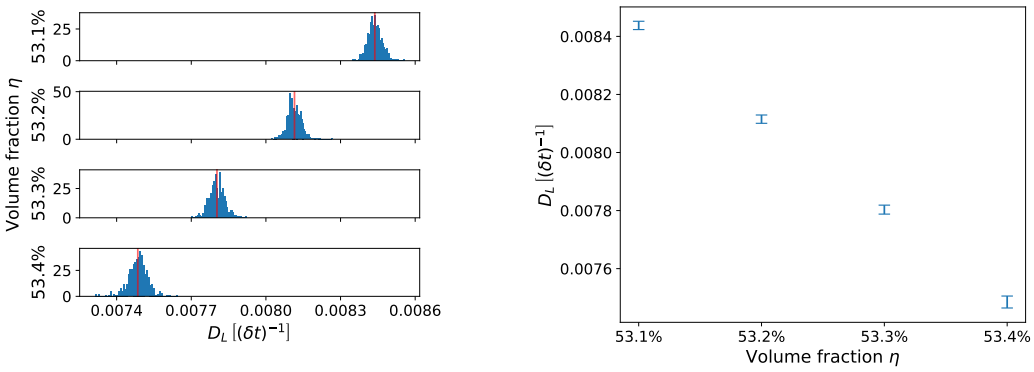
3.2 Long time diffusion time scale

Diffusion or more precisely self-diffusion, characterizes the movement of the single particles within the system. While the diffusive behavior often can be subdivided into different regimes with different physical meanings as discussed in section 2.2.1, only the long time diffusion constant is measured for the hard sphere system. To circumvent finite size effects we use unwrapped coordinates in which case the long time movement of the particles is governed by the relation eq. 3.2.1 which was first described by Einstein 1905[26].

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

With D_L^S the long time self-diffusion constant which will in the following be denoted only by D_L, $\vec{r}(t)$ 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 by saving a reference position of all particles at one point, and furthermore carrying a set of unwrapped positions through the simulation. The squared distance between reference position and unwrapped position is averaged over all particles and used as the 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, where the slopes of the linear regressions to the MSD trajectories are depicted.



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

(b) Mean of the histograms with the uncertainty on the mean given by $\sigma_{\langle D_L \rangle} = \sigma_{D_L}/\sqrt{n}$ 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 means with uncertainty. As we see the

The diffusion coefficients are used to make the time scales of different experiments comparable. It 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 behavior is better accessible with the ballistic simulations where only the long

time diffusion constant is measurable.

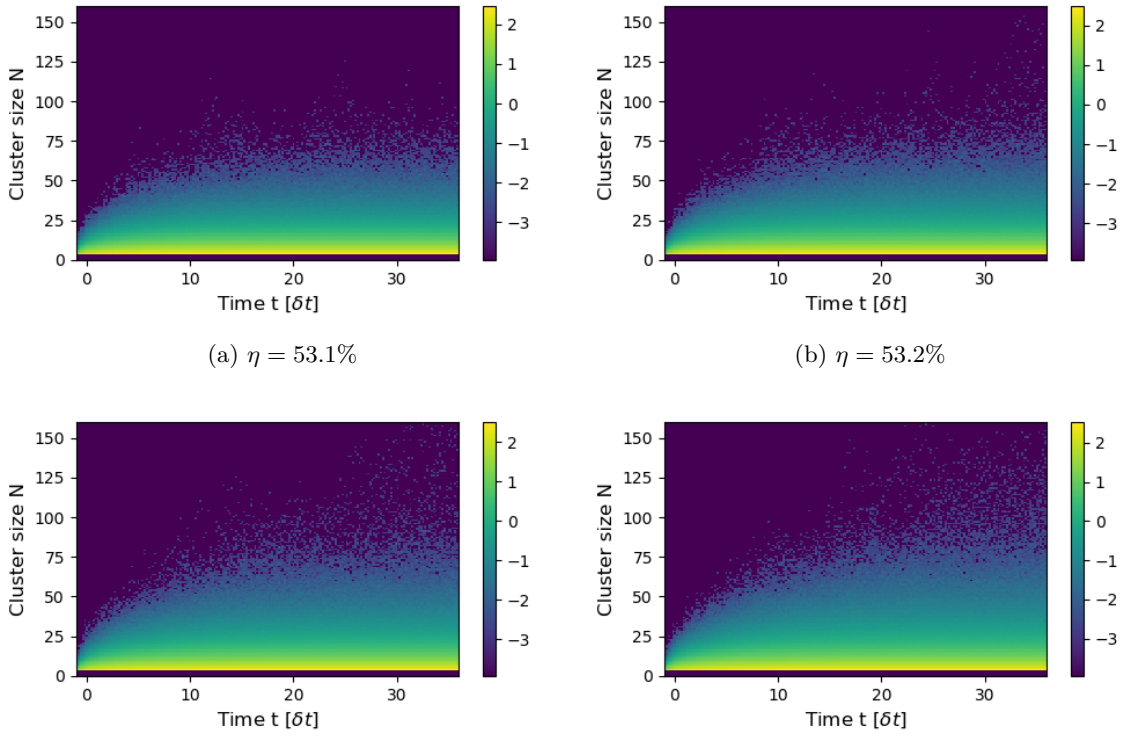
As we see in fig. 3.2.1 the diffusion constants can be measured at high precision with a relative standard deviation of $\sigma_{D_L}/D_L \approx 1\%$. Hence it does not introduce large uncertainties when normalizing time related quantities by the diffusion time $\tau_{D_L} = D_L^{-1}$.

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. 2019[4]. Testing the trajectories shown in fig. 3.3.1 and fig. 3.3.2 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 systems behavior.

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 logarithm is used because the mesurements span orders of magnitude and further it then can be interpreted as a quantity proportional to a free energy. This is justified by assuming that the cluster size distributions represents the correpsonding probability distribution and that stationary states may fluctuate in a free energy landscape where the probability for a particular state with some energy ΔE is given by a Boltzmann distribution $p \propto \exp\left(-\frac{\Delta E}{k_B T}\right)$ from which follows that $\log(p) \propto \Delta E$.

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 changes within the first $15 - 25\delta t$ after which the distribution becomes stable, where the exact length depends on the volume fraction. This might be explained by assuming that the initial phase is how long it takes for the system to build up the local ordering in the metastable liquid and as the clusters tend to be larger for higher volume fractions more particles are required to find their ordering.
To further compare the system time with the more intuitive number of collisions per particle we can use that at the given volume fraction we find $1\delta t \approx \frac{60 events}{particle}$. When further using a



(a) $\eta = 53.1\%$

(b) $\eta = 53.2\%$

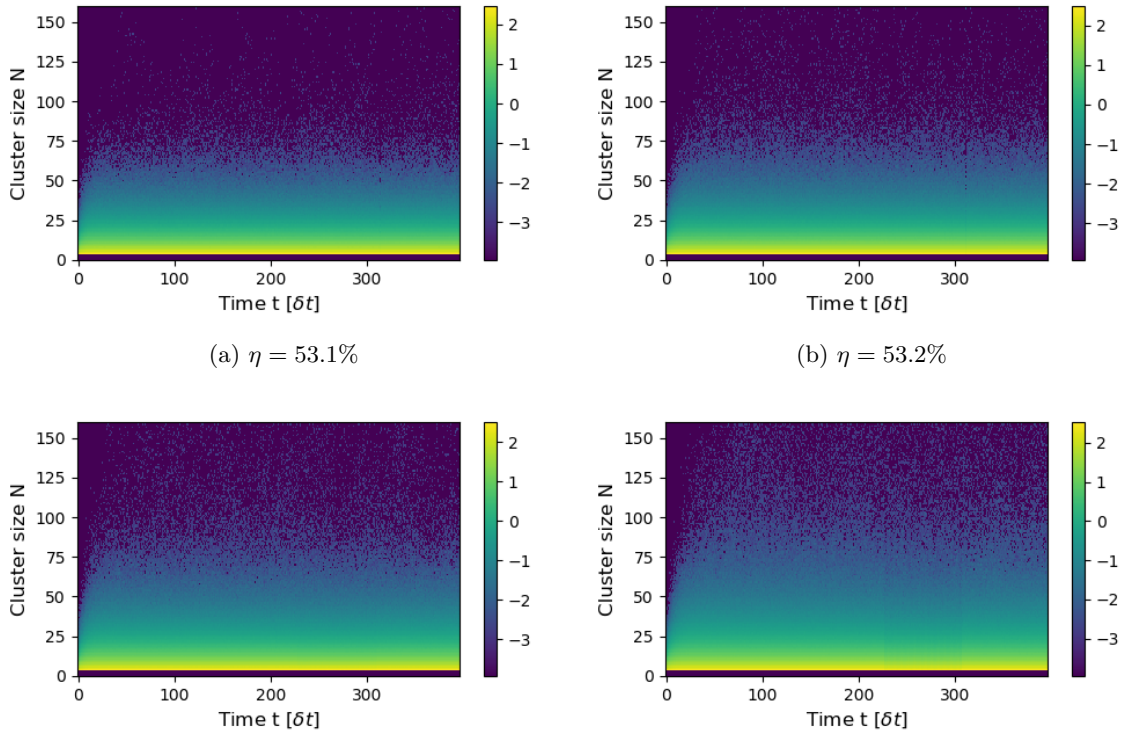
(c) $\eta = 53.3\%$

(d) $\eta = 53.4\%$

Figure 3.3.1: Decadic logarithm of cluster size distributions for different volume fractions in the initial phase after the quench.

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 cluster sizes indicate that this is not a stable process but only a metastable one. Nevertheless this does not mean that it simply can be viewed as a stationary process, because when taking the ensemble as a whole, at any point of time phase transitions take place at various parts of the system.



(a) $\eta = 53.1\%$

(b) $\eta = 53.2\%$

(c) $\eta = 53.3\%$

(d) $\eta = 53.4\%$

Figure 3.3.2: Decadic logarithm of cluster size distributions for different volume fractions during the waiting time.

3.4 Autocovariance functions of largest cluster in the metastable fluid

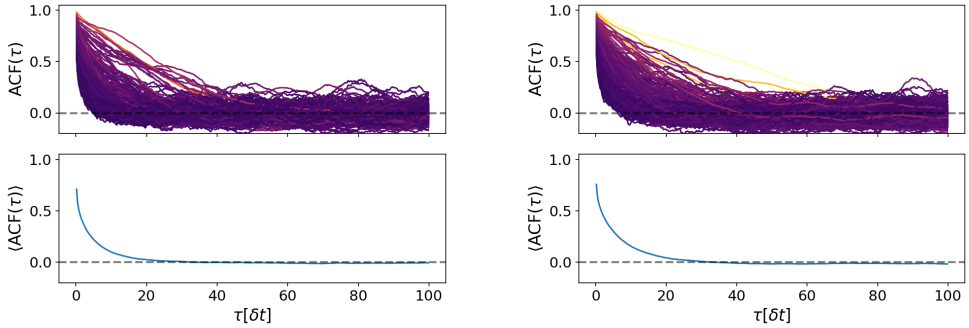
The autocovariance function (ACF) of the largest cluster contains information about how long a single cluster persists as the largest cluster within the volume. This is because fluctuations of clusters at different points of the volume are expected to be independent of each other and only the size of 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 and 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 $ACF(0) = 1$.

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

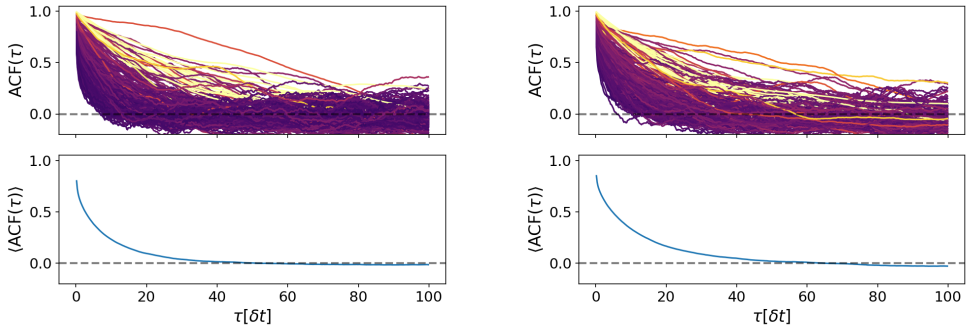
$$\text{with } X(t) = N_{\text{lc}}(t) - \langle N_{\text{lc}} \rangle_t \tag{3.4.2}$$

The ACF is calculated from the largest cluster measurement for each trajectory. Because after a nucleation event the largest cluster size surely is correlated in time, only those parts of the measurements that did not involve strong cluster growth are used. Therefore the ACF's in fig. 3.4.1 show the temporal correlations of the largest cluster 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 each diagram depicts all trajectories with coloring for the largest cluster size within the used time interval. The lightest color thereby indicates a largest cluster of more than 500 hundred particles which is a nucleation event. As these are rare in the given selection, the data represents the metastable fluid well. The bottom of each diagram shows the average of the above one with decay times of $15\delta t - 35\delta t$ depending on the volume fraction.

The decay of the autocovariance functions indicates that structural fluctuations persist for longer times at higher volume fractions. From the coloring, that corresponds to the maximum cluster size within the trajectory, 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. The larger mestastable clusters were also seen before in the cluster size distributions in section 3.3.
The time scale on which the ACF decays corresponds closely to the initial ordering time observed for the cluster distribution directly after the quench. Furthermore 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 within the metastable fluid.

3.5 Cluster growth and constant attachement rate

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, with k being the constant attachment rate, N the number of particles in a specific cluster, A the surface of the cluster, R the radius of the cluster and ρ_{solid} the bulk density which for large clusters is a good approximation of the cluster density.

$$
\begin{aligned}
&\dot{N} = Ak \\
&\quad\Big|\ \text{with} \quad N = \frac{4}{3}\pi R^3 \rho_{solid} \\
&\qquad\Leftrightarrow R = \left(\frac{3N}{4\pi\rho_{solid}}\right)^{\frac{1}{3}}, \\
&\quad\ \text{and} \quad A = 4\pi R^2 \\
&\qquad\Leftrightarrow A = \left(\frac{4\pi 3^2}{\rho_{solid}^2}\right)^{\frac{1}{3}} N^{\frac{2}{3}}, \\
&\frac{dN}{dt} = \left(\frac{4\pi 3^2}{\rho_{solid}^2}\right)^{\frac{1}{3}} N^{\frac{2}{3}} k
\end{aligned}
\qquad
\begin{aligned}
&\text{From the bottom left side} \\
&\Rightarrow \quad dN\ N^{-\frac{2}{3}} = \left(\frac{4\pi 3^2}{\rho_{solid}^2}\right)^{\frac{1}{3}} k\ dt \\
&\quad\Big|\ \text{setting} \quad N(t=0) = 0 \\
&\Leftrightarrow \quad 3N^{\frac{1}{3}} = \left(\frac{4\pi 3^2}{\rho_{solid}^2}\right)^{\frac{1}{3}} kt \\
&\Leftrightarrow \quad N^{\frac{1}{3}} = \left(\frac{4\pi}{3\rho_{solid}^2}\right)^{\frac{1}{3}} kt
\end{aligned}
\tag{3.5.1}
$$

As the systems are able to accommodate clusters up to a few hundred thousand particles and mostly just one cluster forms 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. As an example this is visualized for the trajectories at $\eta = 0.532$ in fig. 3.10.1.
Subsequently the slopes of the linear regressions have been collected in histograms shown in fig. 3.5.2. By eq. 3.5.1 these slopes correspond to constant attachment rates with a prefractor

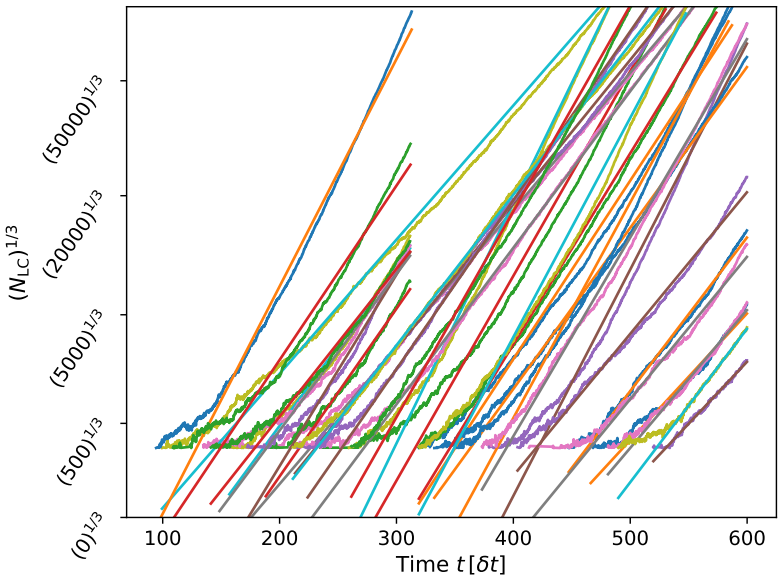
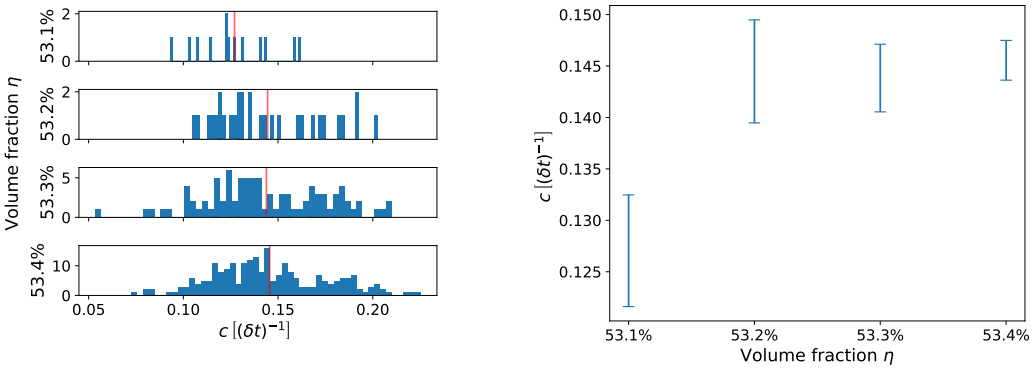


Figure 3.5.1: Trajectories of the third root of the number of particles within the largest cluster $(N_{\mathrm{LC}})^{1/3}$ over time. Clearly visible is the linear proportionality for which a linear regression is shown together with the data. The cut of some data sets at $t \approx 300\delta t$ is due to the trespassing of the maximum wall time of the NEMO computational cluster. Systems that hosted a nucleation event in the first simulation interval before $T \approx 300\delta t$, contain a too large cluster in the next simulation interval leading to the 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 before $t \approx 300\delta t$ might not have been recognized due to this flaw. But the number of trajectories concerned by this is small and the impact is not easy to recognize when looking at the induction time distributions in fig. 3.7.1.

depending on the density within the cluster, but 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 in the qualitative comparison and the constant attachement rate with its prefactor is defined as $c := k\left(\frac{4\pi}{3\rho_{solid}^2}\right)^{\frac{1}{3}}$. With this approximation the equation for the number of particles in a cluster over time, given in eq. 3.5.1, simplifies to the one given before in eq. 1.6.1.

What we see from the histograms is that the distribution is rather spread out, but not significantly depending on the volume fraction. Only for $\eta = 0.531$ we find a smaller growth rate. A possible explanation for this behavior could be that the growth by heterogeneous crystallization on the cluster surface leads to a higher growth rate for higher volume fractions as it is less likely for the lower volume fractions. But due to the low statistics at the lowest



(a) Histograms of the slopes from the linear regressions to third root of the largest cluster during the 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.

volume fraction it is also possible that only a statistical fluctuation is seen.
To investigate if the attachement is diffusion or reaction controlled we may note that 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 as the relative statistical uncertainty of the growth rates is of the order of 5% it requires a larger number of samples to answer this question.

3.6 Tensor of gyration evaluation

The tensor of gyration is a very useful tool as it describes the second moments of the position distributions. Thus it comprises information about the spatial extent in all three dimensions with commonly defined quantities being the radius of gyration, asphericity and anisotropy, see Theodorou and Suter 1985[27].

The tensor of gyration itself is defined by

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

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

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 that are chosen with $\lambda_1^2 \leq \lambda_2^2 \leq \lambda_3^2$. These three Eigenvalues correspond to the spatial extents of the cluster within the Cartesian system in which the tensor of gyration becomes diagonal. The aforementioned shape descriptors are defined in eq. 3.6.3 - 3.6.6.

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

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

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

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

For a better understanding of the above defined descriptors their meaning is discussed in the following.

Radius of gyration R_G

An averaged radius of the structure. For a sphere with radius R it is given by $R_G = \sqrt{\frac{3}{5}}R$.

Asphericity b

The difference of the largest extent and the average of the two smaller extents. For a sphere these are the same and the asphericity becomes zero, even though this is also the case for a cube.

Acylindricity c

The difference of the two smaller extents, as for a long cylinder they are the same and the acylindricity becomes zero.

Relative shape anisotropy κ^2

A weighted squared sum of the asphericity and the acylindricity normalized by the fourth power of the radius of gyration to obtain a dimensionless quantity between 0 and 1. For a sphere it is zero while it becomes one in the case of all particles being aligned in a straight line.

To spot possible correlations between a cluster's shape and its growth, the radius of gyration, the asphericity and the relative shape anisotropy have been plotted against the cluster size and then colored 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 they 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 behavior for similar cluster sizes. Because the cluster size depending on time is almost monotonic for cluster sizes above a few hundred particles, it roughly corresponds to 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 coloring schemes in the horizontal direction.

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 to the cluster growth. Also a similar approach for the three scalar quantities that describe the growth process has been done, but neither showing significant correlations.

Nevertheless from the calculated means, 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, as this also would lead to unspherical growth. But as the clusters are rather close to a sphere, the attachemnt rate would also not vary much between the different crystal faces. It also 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 domains form as it seems that heterogenous nucleation takes place close to the surface of the cluster, leading to new crystal orientations that are included into the crystal.

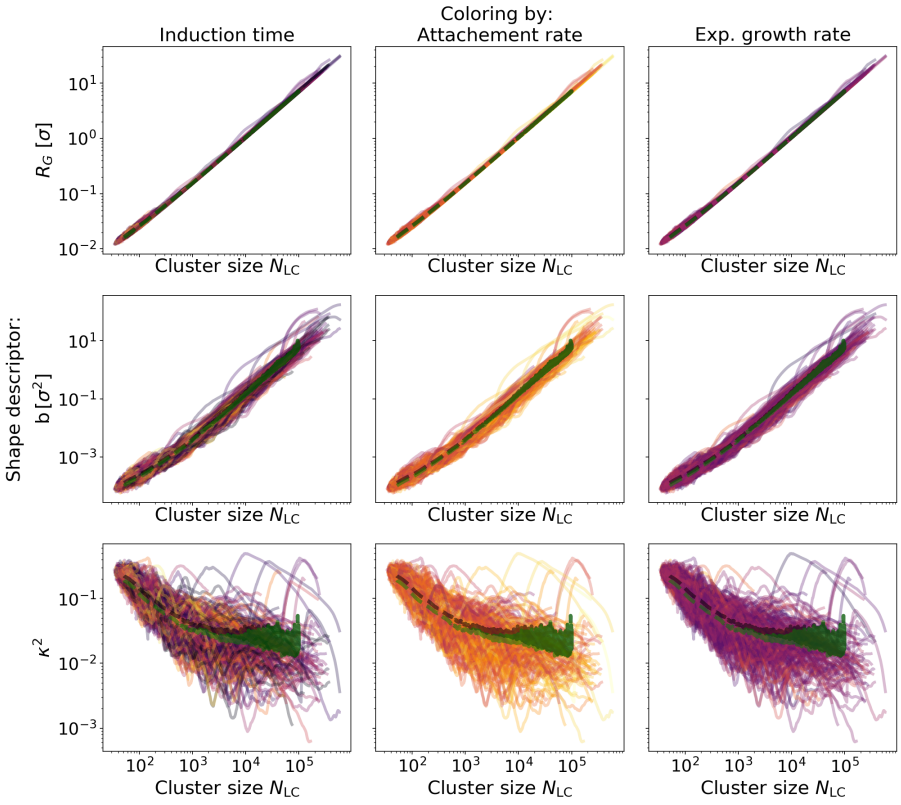


Figure 3.6.1: Overview of the shape descriptors radius of gyration (R_G), asphericity (b) and anisotropy (κ^2), depending on the size of the cluster are shown. The coloring indicates the scalar quantities induction time, constant attachment rate and initial growth rate. Further a smoothed arithmetic mean and median are included in black and green respectively.

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 the point at which a cluster is not merely an unstable fluctuation in the liquid anymore, but instead becomes a stable crystalline solid.

Many definitions can and have been used used for htis purpose. For example a cluster can be defined as crystalline soon as it surpasse the CNT's critical size or a multiple of it. One can also use a committer analysis to find the size where a crytallite keeps growing with a 50:50 chance. Also often apllied is the method to rewind a trajectory with a stable crystal back to the point where the cluster's size 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 only by a delay Δ_τ which is a distribution 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, is not distinguishable from any other structural fluctuation in the liquid i.e. when the size of the cluster is 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 others 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. After all this means that the induction time depends directly on the definition of crystallization and they are only roughly comparable. In fig. 3.7.1 three distributions with varying definitions for 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 nucleated system back to the point where it last crossed the average largest cluster of the metastable fluid. The name horizon crossing refers to the idea that fluctuations of the largest cluster are mostly independent, as the largest cluster is not fixed in the box, but fluctuations at different locations contribute. Only extraordinary large fluctuations will be seen for a prolonged periods of time and therefore will lead to correlated fluctuations of the largest cluster size. The crossing of the trajectory below this horizon, where it does not describe a distinct cluster anymore, 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 function crossed 10 particles, which is then taken as the induction time. The method tends to find negative induction times that are not physical, but only an artifact of the method.

Constant extrapolation

The name refers to the constant attachment rate found at later times for the cluster

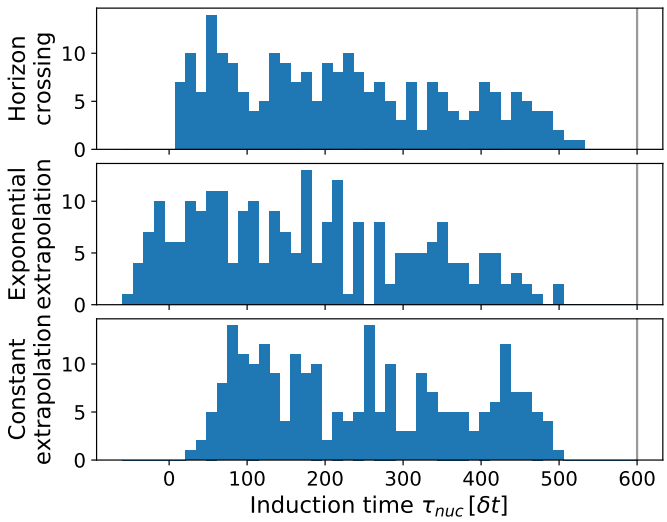


Figure 3.7.1: Induction time distribution obtained by different definitions. While the two methods using extrapolation seem to have the two effects of smearing the signal as well as shifting them, the method of defining the nucleation as the time when the largest cluster is last below the horizon of 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 clearly be recognized as crystals, no nucleation events are seen towards the end of the simulation interval. To counteract this we truncate the distribution in the analysis such it does not introduce a bias on the final result.

growth. It can be extrapolated to earlier times until the cluster completely vanishes. 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 assumption

Some of this intorduction stuff may better fit into comparison to real world?

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 system and information.

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 MD simulations their velocities as well.
On the experimental side light scattering and optical methods are mostly employed to measure the structural properties of the probe, on the theoretical side different cluster finding algorithms are used.
While experimentalists may define an induction time by how long 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 for example by Filion et al. 2010[11] cite it here or not? It not such a positive statement. This method requires 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 which can be compared to other literature values.

As 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 it is commonly defined as a nucleation rate density κ. By using the diffusion time $\tau_L = D_L^{-1}$ as a unit of time furthermore makes the comparsion to other systems with faster or slower dynamics possible.

Considering a set of m simulations at a given volume fraction, we can describe the total system as a sum of m subvolumes, each of size 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 subvolumes we can deduce the expected induction rate.

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

$$\Leftrightarrow \quad \frac{\dot{n}}{m} = \left(1 - \frac{n(t)}{m}\right) V_{box}k$$

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

$$\Leftrightarrow \frac{n(t)}{m} = 1 - \exp\left(-V_{box}kt\right) \tag{3.8.2}$$

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

$$\Leftrightarrow \frac{\dot{n}(t)}{m} = \frac{1}{\tau}\exp\left(\frac{-t}{\tau}\right) \tag{3.8.3}$$

The final result in eq. 3.8.3 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 the 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 (ML) estimators. The derivation follows the one by Deemer and Votaw 1955[28].

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, it is given by

$$\mathcal{L}(\kappa) = \prod_{i=1}^{N} p(x_i) = \prod_{i=1}^{N} \kappa^N \exp\left(-\kappa x_i\right) \ . \tag{3.8.4}$$

We continue by find the maximum of this product. To simplify it 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. 3.8.5 - eq. 3.8.8.

$$0 \stackrel{!}{=} \left.\frac{\partial \log(\mathcal{L})}{\partial \kappa}\right|_{\kappa=\hat{\kappa}} \tag{3.8.5}$$

$$\Leftrightarrow \quad 0 = \left.\frac{\partial}{\partial \kappa}\left(N\log(\kappa) - \kappa\sum_{i=1}^{N} t_i\right)\right|_{\kappa=\hat{\kappa}} \tag{3.8.6}$$

$$\Leftrightarrow \quad 0 = \frac{N}{\hat{\kappa}} - \sum_{i=1}^{N} t_i \tag{3.8.7}$$

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

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 distributions refer to sets of samples that are incomplete in the sense that they only include samples up to some threshold $t_i < T$, but while in the case of truncated distributions the number of samples larger than this threshold is unknown, for the censored distribution it 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 boxes is known but the simulation is just stopped at some point without all boxes having had a nucleation event. The probability of an event later than the censoring time T is given by

$$p(t_i > T) = \int_T^\infty \kappa\exp(-\kappa t)dt = \exp(-\kappa t)\ . \tag{3.8.9}$$

Therefore we can write the complete probability distribution as

$$f(t) = \begin{cases} \kappa\exp(-\kappa t) & t < T \\ \exp(-\kappa T) & t \geq T \end{cases}\ . \tag{3.8.10}$$

In the simulation we can then 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 which are calculated in the binomial prefactor $\binom{N}{m}$. The whole expression for the likelihood function $\mathcal{L}(\kappa)$ then is given by eq. 3.8.11 and the extremum of it is evaluated in the subsequent reformulations.

$$\mathcal{L}(\kappa) = \binom{N}{m} \kappa^n \ \exp(-\kappa \sum_{i=1}^{n} t_i) \ \exp(-\kappa T)^m \quad \left| \ \frac{\partial \log(...)}{\partial \kappa}\right|_{\kappa=\hat{\kappa}} \tag{3.8.11}$$

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

$$\Leftrightarrow \frac{\partial \log(\mathcal{L}(\kappa))}{\partial \kappa} = \frac{n}{\kappa} - \sum_{i=1}^{n} t_i - mT \quad \Big|_{\kappa=\hat{\kappa}} \tag{3.8.13}$$

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

$$\Leftrightarrow \qquad 0 = \frac{n}{\hat{\kappa}} - \sum_{i=1}^{n} t_i - mT \tag{3.8.14}$$

$$\Leftrightarrow \qquad \hat{\kappa}^{-1} = \frac{1}{n} \left(\sum_{i=1}^{n} t_i + mT \right) \tag{3.8.15}$$

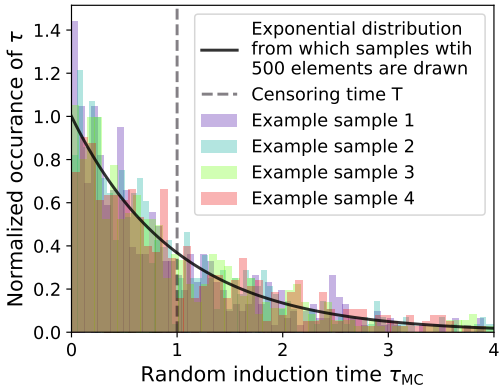
The final line eq. 3.8.15 is the estimator of the decay rate of the censored exponential distribution. It is used for the estimation of induction times to compare with other published results in the next sections.

3.8.3 Monte Carlo uncertainty estimation

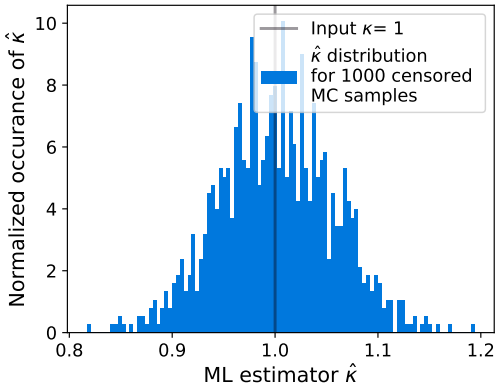
Having determined the estimator for the nucleation rate, the next question concerns its uncertainty i.e what is the distribution of $\hat{\kappa}$? While corresponding literature on analytic expressions for the distribution has been published for example by Chen and Bhattacharyya 1988[29], the complexity becomes inappropriate for the task at hand. Thus we will follow instead a Monte Carlo approach described for example in the book Numerical Recipes[30] 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 estimators uncertainty $\sigma_{\hat{\kappa}}$.

Concerning the uncertainty in detail we can ask how long a simulation should last to yield precise results. For this we can first look at the case where $1 \gg \kappa T$ corresponding to a



(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.

simulation where all boxes showed a 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's are exponentially distributed we know that $\sigma_t = \kappa^{-1}$. Gaussian error propagation then results in

$$\frac{\sigma_{\hat{\kappa}}}{\hat{\kappa}} = \frac{1}{\sqrt{N}} \,. \tag{3.8.16}$$

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 nucleation events due to the smaller not nucleated 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 total number of boxes. As n is Poisson distributed we know that $\sigma_n = \sqrt{n}$. Fixing N and T and using the expectation value of nucleations $n = N\hat{\kappa}T$, the Gaussian error propagation for the relative uncertainty is given in eq. 3.8.17.

$$\frac{\sigma_{\hat{\kappa}}}{\hat{\kappa}} = \frac{1}{\hat{\kappa}}\frac{\sqrt{n}}{NT} = \frac{\sqrt{N\hat{\kappa}T}}{NT\hat{\kappa}} = \frac{1}{\sqrt{N\hat{\kappa}T}} \tag{3.8.17}$$

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 set consistently with the actual number of used simulations to 500 and to achieve good precision on the uncertainty, the standard deviation of 1000 samples is used. To compare the analytically derived limits of the uncertainty with the Monte Carlo results both are drawn into fig. 3.8.2.

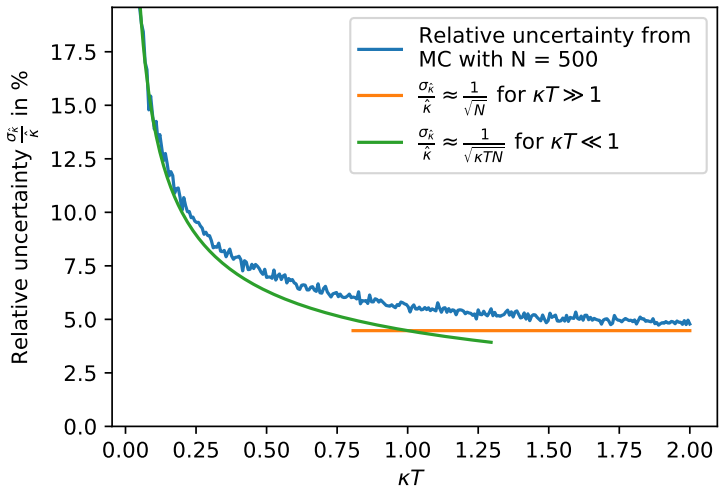


Figure 3.8.2: Relative uncertainty of the ML estimator for varying κT. The x scale is chosen dimensionless such that it indicates the simulation time in comparison to the characteristic nucleation time.

We find that for the limits of $\kappa T \ll 1$ as well as $\kappa T \gg 1$, Monte Carlo 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 induction time, after which it only obtains little more precision. This is not surprising as the nucleation times contain the rate and more nucleation events occur at the beginning while long simulation times only add little further information. Thus simulating until all boxes hosted a nucleation event is only necessary if one wants to use the simpler arithmetic mean of the induction times as the setimator, or if any other constraints make it necessary to reach crystallization 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.

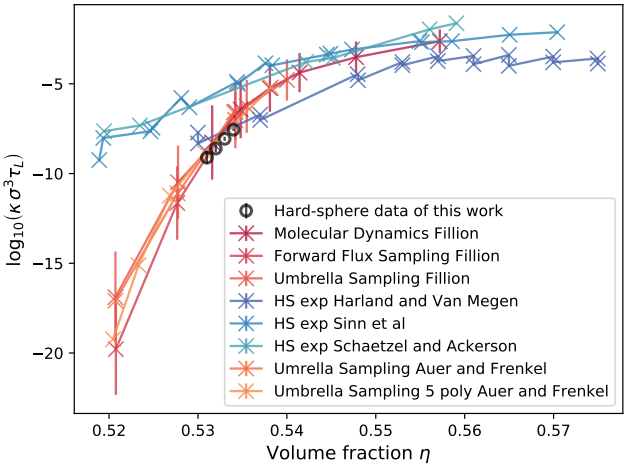


Figure 3.9.1: Experimental and theoretical examples of nucleation rates in the hard sphere system at different volume fractions from the literature citations and/or other data, to compare with the own measured data points.

From the diagram we can state that our measurements confirm the previous simulation results, that still stand against the experimentally found ones. Further the results are calculated together with their statistical uncertainty which is mostly visible for the data point at $\eta = 53.1\%$. It is also indicated for the others but due to the logarithmic scaling the uncertainty is almost not visible.
While uncertainties in the literature are often just very roughly given, the here presented method makes it possible to quantify the statistical uncertainty of the rates and the large number of simulations give us high precision in comparison to rates elsewhere found.

3.10 Memory kernels of nucleating ensemble

The approach by Hugues Meyer et al. 2019[31], to calculate memory kernels from an ensemble of trajectories, is used on the data discussed in the previous sections as well as on trajectories of a system characterized in tab. 3.10.1. The second system is used because the first ensemble was neither simulated up to the point where most boxes contained a stable cluster nor until the boxes were fully crystallized as the transition width is large and takes very long simulations to fill up the large box. The second system's parameters are chosen to fulfill both

objections.

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 crystallization 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 closely to the width of the memory kernel, as lately shown by Meyer et al. 2021[1].

Still the memory kernel of the large system has been calculated but except of the Markovian contribution only little 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 nucleations 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, such trajectories above 0.5 as fully crystallized and all trajectories in between as in the crystallization process.

While for the large system only little of the crystallites reached the box boundaries in this latter we see that almost all clusters fill the whole box at the end of the simulation.
Further as there is no clear analysis yet on how the direct quantities and the memory kernel are related, the subdivision is an approach to show direct observables that possibly are related to properties of the memory kernel.

We see in on the left of fig. 3.10.2, that 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

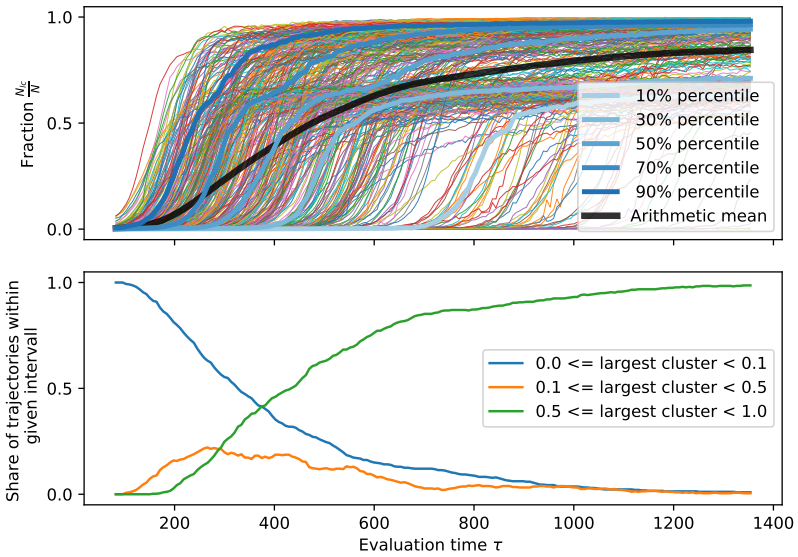
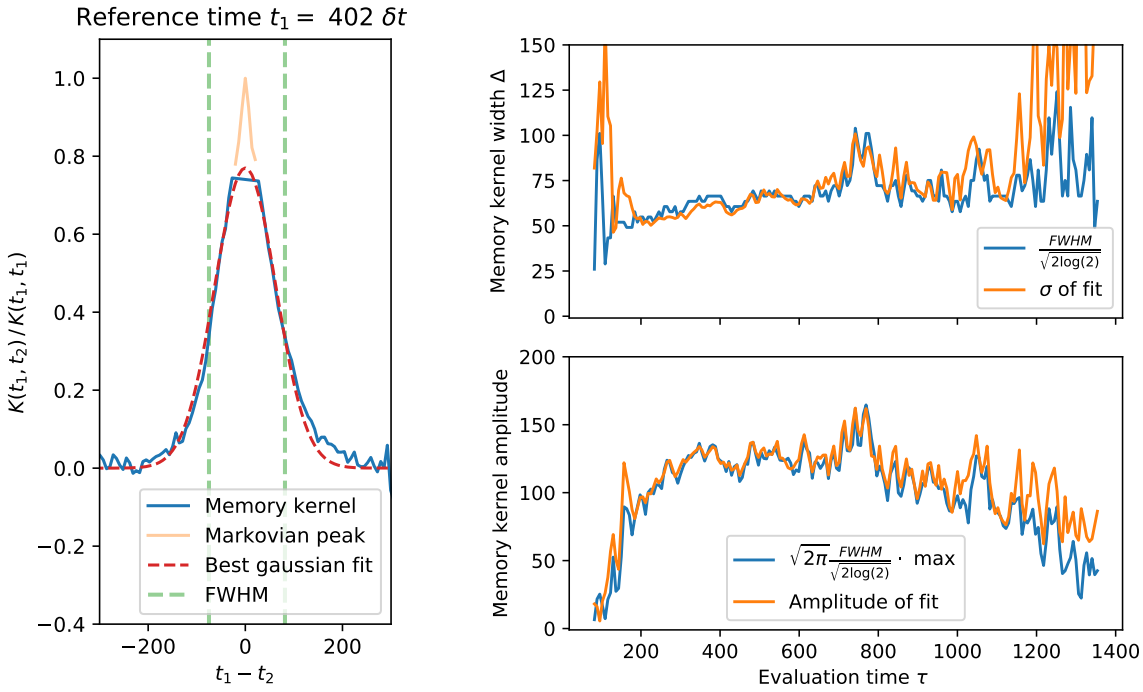


Figure 3.10.1: Top: Normalized trajectories of largest cluster with percentiles and arithmetic mean indicated. It can be observed that a fraction of the trajectories nucleates in more than one step where at first only about 60% of the box is filled by the crystal and at later times they sometimes crystallize further until almost the complete box is filled by the solid phase. From eq. 1.3.6 we would expect an equilibrium solid fraction of 80% by volume, closely corresponding 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.

is determined by the mean value of the peak's crest.
As the properly normalized results for both methods are in good agreement, we can conclude that the shape of the memory kernel in this case is mostly defined by a width and an amplitude over time which are depicted on the right of fig. 3.10.2.

The width of the memory kernel sections are mostly constant over the whole measurement with the exception that it becomes very noisy 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 published by Meyer et al. 2021[1], the width of the memory kernel seems to depend on the phase transition time. Because for the hard sphere system the transition width is mostly given by the arbitrarily chosen box size, the dependence is possibly only an artifact with other memory effects buried beneath.



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

To seperate these memory effects one could generate trajectories with a purely Markovian approach, like Brownian dynamics, with corresponding characteristic properties. Then comparing the memory kernels of the purely Markovian ensemble with the a priori non Markovian hard sphere ensemble may help to distinguish memory effects due to the system size from those related to the dynamics of the fluid.

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 and by itself is already bounded between zero and one, which is a requirement for the memory kernel analysis.

4 Conclusion

5 Appendix

.1 A

Bibliography

[1]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).

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

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

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

[6]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).

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

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

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

[10]T. Schilling, S. Dorosz, H. J. Schöpe, and G. Opletal, "Crystallization in suspensions of hard spheres: A Monte Carlo and molecular dynamics simulation study", Journal of Physics Condensed Matter **23** (2011).

[11]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).

[12]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).

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

[14]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).

[15]M. Radu and T. Schilling, "Solvent hydrodynamics speed up crystal nucleation in suspensions of hard spheres", Epl **105**, 1–7 (2014).

[16]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).

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

[18]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).

[19]A. Donev, S. Torquato, and F. H. 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).

[20]P. J. Steinhardt, D. R. Nelson, and M. Ronchetti, "Bond-orientational order in liquids and glasses", Physical Review B **28**, 784–805 (1983).

[21]P. R. Ten Wolde, M. J. Ruiz-Montero, and D. Frenkel, "Numerical evidence for bcc ordering at the surface of a critical fcc nucleus", Physical Review Letters **75**, 2714–2717 (1995).

[22]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).

[23]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).

[24]I. R. Hansen, Jean-Pierre McDonald, "Chapter 4 - Distribution-function Theories", in *Theory of simple liquids*, Third Edit (Academic Press, 2006), p. 94.

[25]P. N. Pusey et al., "Hard spheres: crystallization and glass formation", Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences **367**, 4993–5011 (2009).

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

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

[28]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).

[29]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).

[30]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).

[31]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).