Crystallization Pathways and Mechanisms of Charged Macromolecules at low Supersaturations

Von der Fakultät Mathematik und Physik der Universität Stuttgart
zur Erlangung der Würde eines Doktors der
Naturwissenschaften (Dr. rer. nat.) genehmigte Abhandlung

Vorgelegt von

Kai Kratzer

aus Stuttgart

Hauptberichter:	Jun.-Prof. Dr. Axel Arnold
Mitberichter:	Prof. Dr. Hans-Rainer Trebin
Mitberichter:	Prof. Dr. Frank Noé
Tag der mündlichen Prüfung:	18.12.2014

Institut für Computerphysik der Universität Stuttgart

2014

Erklärung

Hiermit versichere ich, die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt, sowie die Zitate deutlich kenntlich gemacht zu haben.

Stuttgart, den 11. November 2014

Kai Kratzer

Contents

Acronyms

bcc	body-centered cubic
CNT	Classical Nucleation Theory
CPU	Central Processing Unit
DNA	Deoxyribonucleic Acid
DFFS	Direct FFS
DLVO	Derjaguin, Landau, Verwey and Overbeek theory
ESPResSo	Extensible Simulation Package for Research on Soft matter
fcc	face-centered cubic
FENE	Finitely Extensible Nonlinear Elastic
GROMACS	Groningen Machine for Chemical Simulations
hcp	hexagonal close-packed
HLRS	Höchstleistungsrechenzentrum Stuttgart
LAMMPS	Large-scale Atomic/Molecular Massively Parallel Simulator
LJ	Lennard-Jones
MC	Monte Carlo
MD	Molecular Dynamics
NPT	Isothermal-isobaric ensemble
NS-FFS	Non-Stationary Forward Flux Sampling
FFS	Forward Flux Sampling
FRESHS	Flexible Rare Event Sampling Harness System
RDF	Radial Distribution Function

SEGL Science Experimental Grid Laboratory

SPRES Stochastic Process Rare Event Sampling

TCP/IP Transmission Control Protocol/Internet Protocol

WCA Weeks-Chandler-Andersen

Nomenclature

β	inverse temperature $(k_B T)^{-1}$
$\Delta\mu$	supersaturation
$\delta\Phi$	escape flux deviation
ΔG	free energy difference
ΔG^*	maximum of the free energy difference
ΔP	pressure tail correction
Δt	time step
ϵ	characteristic unit of energy
ϵ_0	vacuum permittivity
ϵ_c	energy cutoff
ϵ_r	dielectric constant
η	volume fraction
Γ	kinetic pre-factor
γ	surface tension
κ	inverse screening length
λ	order parameter
λ_A	border of state A
λ_B	border of state B
λ_i	interface position
$\langle A \rangle$	averaged observable
$\mathbf{a}$	acceleration vector

$\mathbf{f}$	force vector
$\mathbf{p}$	momentum vector
$\mathbf{r}$	position vector
$\mathcal{C}$	computational cost
$\mathcal{E}$	efficiency
$\mathcal{R}$	Gaussian random variable
$\mathcal{V}$	statistical error
μ	chemical potential
∇	Nabla symbol
Ω	number of eigenstates
Ω_B	degeneracy of heat bath B
$\overline{q}_l$	averaged local bond order parameter
Φ	escape flux
Φ_A	escape flux of the forward simulation
Φ_B	escape flux of the backward simulation
$\pi(q;\lambda_i)$	order parameter distribution of interface i
Ψ_A	contribution to the stationary distribution of the forward simulation
Ψ_B	contribution to the stationary distribution of the backward simulation
ρ	number density
σ	characteristic unit of length
τ	simulation time step
$\tau_+(q;\lambda_0)$	weighted factor of the forward simulation
$\tau_-(q;\lambda_n)$	weighted factor of the backward simulation
$\widetilde{N}_b$	neighbors of the second shell
ξ	friction coefficient

a	acceleration
B_n	cluster with n particles
c	concentration
$C(n)$	cluster size distribution
E	energy
F	Helmholtz free energy
f	force in one direction
f_+^*	attachment rate to the critical cluster
f_i	constant net flux measurement function
f_x	force in x-direction
G	Gibbs free energy
$g(r)$	radial distribution function
H	Hamiltonian
h	barrier height
k_B	Boltzmann's constant
k_+	attachment rate
k_-	detachment rate
k_{AB}	Transition rate from A to B
L	box length
l	order of the spherical harmonics
l_B	Bjerrum length
l_D	Debye screening length
M	number of trial runs
m	mass
N	number of particles/neighbors

n	number of particles
n^*	number of particles in the critical cluster
N_0	number of configurations on λ_A
N_b	neighbors of the first shell
N_f	degree of freedom
n_i^0	mean concentration of charges
$N_n(t)$	time dependent cluster size distribution
P	pressure
p_A	probability to be in A
P_B	probability to reach B
p_B	probability to be in B
P_i	probability to find system in state i
p_i	transition probability to interface $i+1$
Q	partition function
q	general order parameter
q_l	local bond order parameter
$Q_{1,2}$	charges
R	cluster radius
r	(radial) distance
R^*	critical cluster radius
r_{cut}	cutoff distance
S	entropy
T	temperature
t	time
U	potential

V	Volume
v	velocity
v_0	molecular volume
x	X-coordinate
y	Y-coordinate
Y_{lm}	spherical harmonics
Z	Zeldovich factor
z	Z-coordinate

Zusammenfassung

Die Kristallisation von geladenen Makromolekülen spielt eine wichtige Rolle in vielen Fachgebieten wie der Biologie, der Medizin, der Physik und im Materialdesign. Zum Beispiel benutzt man die Kristallisation bei Proteinen, um diese von anderen Bestandteilen in einer Lösung zu trennen, und kolloidale Kristalle sind vielversprechende Bausteine für photonische Kristalle mit optischen Anwendungen. Ein weiteres Beispiel ist die Kristallisation für die Strukturaufklärung mit elektromagnetischer Strahlung wie z. B. Röntgenstrahlung.

Um eine geschlossene Theorie der Kristallisation von geladenen Makromolekülen zu formulieren, ist es notwendig, die mikroskopischen Details der Kristallisation zu erforschen, insbesondere die Entstehung des Keims aus der vorhandenen Phase. Dieser Prozess wird Nukleation genannt und kann im grundlegenden Fall durch die klassische Nukleationstheorie beschrieben werden. In dieser wird der Wachstumsprozess als ein Gleichgewicht aus Oberflächenspannung zwischen dem Kristall und der Flüssigkeit sowie des chemischen Potentials beschrieben, wobei der Oberflächenanteil quadratisch mit der Größe des Kristalls eingeht und der Volumenanteil kubisch. Dies führt zu einer Energiebarriere, die überwunden werden muss, wenn man einen Kristall aus einer flüssigen Phase aufbauen möchte. Mit der klassischen Nukleationstheorie können quantitative Vorhersagen für den Nukleationsprozess gemacht werden. Jedoch werden in dieser Theorie einige vereinfachende Annahmen getroffen, zum Beispiel wird eine ideale Kugelform des wachsenden Kristalls angenommen. Mit einer vollständigen Theorie wäre es möglich, Kristallisationsbedingungen vorherzusagen. Damit lassen sich Kristalle aus geladenen Makromolekülen zielorientiert ziehen. Z. B. könnte man defektfreie Kristalle für optische Anwendungen herstellen, oder auch das Wachstum unerwünschter Kristalle unterdrücken.

Um weiter mit dem Vorhaben einer geschlossenen physikalischen Theorie der Kristallisation voranzukommen und die Abweichung der Theorie von den realen Systemen zu minimieren, untersuchen wir in dieser Arbeit die Kristallisation von geladenen Makromolekülen mit Hilfe von Molekulardynamik- (MD-) Simulationen. Hierfür benutzen wir eine 3D Simulationsbox mit periodischen Randbedingungen im NPT-Ensemble. Dabei werden die geladenen Makromoleküle durch ein effektives Potential modelliert, welches die Abschirmung der dissoziierten Ladungen und der neutralisierenden Salzionen der Lösung beinhaltet, ein sog. Yukawapotential.

Dieses effektive Potential beschreibt ein reales kolloidales System, wie es in Experimenten verwendet wird. Dies haben wir während dieser Arbeit in Kooperation mit dem 2. Physikalischen Institut der Universität Stuttgart für kolloidale Systeme überprüft.

Dabei benutzten wir unsere Simulationen, um das Wechselwirkungspotential zu finden, welches die radiale Verteilungsfunktion des experimentellen Systems am besten reproduzieren kann. Bei niedrigeren Systemdichten konnte sehr gut ein Yukawapotential an die erhaltenen Potentiale angepasst werden und somit die Abschirmlänge und der Wechselwirkungsfaktor bestimmt werden. Bei höheren Dichten konnte nur der repulsive Teil des Potentials damit beschrieben werden, da Dreikörperwechselwirkungen an Gewicht gewinnen. Die Abschirmlänge der Wechselwirkung hängt von der Dichte des Systems ab, da die Ladungen im System von den dissoziierten Oberflächenladungen der Kolloide herrühren, deren Anzahl von der Menge der Kolloide pro Volumen abhängt.

Um das Kristallwachstum im Detail zu untersuchen, ist es nötig, nahe der Koexistenzlinie im Phasendiagramm, an der die Wachstumsrate gering ist, zu simulieren. An solchen Phasenpunkten ist die Energiebarriere für die Nukleation sehr hoch. Dies erschwert Untersuchungen sowohl im Experiment als auch in Simulationen, da man für eine statistische Auswertung sehr lange beobachten bzw. simulieren müsste.

In letzter Zeit wurden einige *Rare Event Sampling* Methoden entwickelt, um mit einer Simulation trotzdem Aussagen in solchen Fällen zu ermöglichen. Dabei werden die unnötigen Wartezeiten, in denen nur Trajektorien simuliert werden, die nicht zum Ziel führen, reduziert. Statistische Methoden korrigieren dann diesen Prozess und es lassen sich Aussagen über die Übergangsraten zwischen den jeweiligen Zuständen treffen. Weiterhin können erfolgreiche Pfade extrahiert und physikalische Mechanismen analysiert werden.

In dieser Arbeit verwenden wir das sog. *Forward Flux Sampling* (FFS). Dabei wird der Reaktionsweg durch den Phasenraum durch einen Ordnungsparameter charakterisiert und durch einen Satz von sog. Interfaces in mehrere Stufen unterteilt. Der Vorteil dabei ist, dass die Übergänge der einzelnen Stufen mit konventionellen Simulationsmethoden simuliert werden können und sich daraus dann der Gesamtübergang ergibt.

Für unsere Untersuchungen war es nicht nur nötig, diese Methode zu parallelisieren, sondern auch im Hinblick auf die Effizienz, welche sich aus Gleichgewicht von Rechenaufwand und statistischem Fehler ergibt, weiterzuentwickeln. Für die parallele Implementierung haben wir die FFS-Methode mit mehreren Simulationen analysiert, z. B. mit einem eindimensionalen Teilchen, welches sich anfangs in einem Potentialminimum befindet und eine Energiebarriere überqueren muss, als auch mit komplexeren Problemen wie der Gasbläschenbildung in einer Flüssigkeit und der Translokation eines Polymers durch eine Nanopore.

Die Effizienz der FFS-Methode hängt hauptsächlich von der Position und dadurch der Anzahl der Interfaces für die Übergänge ab. Während meines Auslandsaufenthaltes an der University of Edinburgh entwickelten wir zwei Methoden, um diese Positionen automatisch zu schätzen und damit die Interfaces an den optimalen Positionen im Phasenraum zu platzieren. Dafür entwickelten wir ein analytisches Modell, das beschreibt, wie die Effizienz von den Übergangswahrscheinlichkeiten zwischen den

Interfaces abhängt, welches wir anschließend durch Simulationen verifizierten. Diese Optimierung kann nun direkt während der Simulation automatisch geschehen, sodass für die Simulation nur noch der Anfangszustand und der Endzustand spezifiziert werden müssen, und die Simulation dann von selbst und optimiert den Weg durch den Phasenraum findet.

Um unsere Weiterentwicklungen und parallele Implementierungen auch anderen Forschungsgruppen zugänglich zu machen, erstellten wir ein Framework, in dem alle diese Funktionen enthalten sind, unser sog. *Flexible Rare Event Sampling Harness System* (FRESHS), welches nun in Kooperation mit einer Forschungsgruppe in Luxemburg weiterentwickelt wird.

Mit Hilfe dieser Vorkenntnisse und Werkzeuge war es möglich, die Kristallisation von geladenen Makromolekülen zu untersuchen. Um den Fortschritt der Kristallisation im System zu quantifizieren, wurde der sog. lokale Bindungsordnungsparameter $\overline{q}_l$ benutzt, mit Hilfe dessen man erkennen kann, ob das entsprechende Molekül in einem Kristallgitter eingebaut ist. Die Anzahl der festen Partikel im größten Cluster wird dann als Ordnungsparameter für die FFS-Methode verwendet. Mit diesem System wurden FFS-Simulationen an verschiedenen Punkten im Phasendiagramm durchgeführt und die Kristallisation der Makromoleküle simuliert. Dabei wurden u.a. die Übergangsraten und finalen Strukturen für diese verschiedenen Punkte bestimmt. Es zeigte sich, dass die Übergangsraten bei Annäherung an die Phasenkoexistenzlinie drastisch abnahmen und dadurch wie erwartet Nukleationsereignisse sehr selten wurden.

Durch Extrahieren der erfolgreiche Pfade konnte der Nukleationsprozess direkt verfolgt werden. Dabei fanden wir heraus, dass an den Stellen, an denen sich der Kristall bilden wird, schon sehr früh eine lokale sechszählige Symmetrie in der flüssigen Phase beobachten lässt. Wir fanden keine Korrelation zu einer vierzähligen Symmetrie oder zur lokalen Dichte. Somit wird die Nukleation über lokale Fluktuationen der räumlichen Ordnung getrieben, und nicht über Dichtefluktuationen, wie klassisch oft argumentiert wird.

Je nach Druck und Kontaktenergie des Yukawapotentials zeigt die Kristallisation eine andere feste Struktur an der Grenze des Endzustands B. Für niedrige Kontaktenergien und hohe Drücke wuchs eine hcp/fcc-artige Struktur, was auch mit dem Phasendiagramm übereinstimmte, während für höhere Kontaktenergien und niedrige Drücke eine bcc-artige Struktur vorherrschte, was bei den untersuchten Punkten aber nicht die thermodynamische Gleichgewichtsstruktur darstellt.

Die Erklärung ist, dass es sich hierbei um einen zweistufigen Kristallisationprozess mit zwei Energiebarrieren handelt. Die erste Stufe ist der Übergang von der flüssigen Phase zur metastabilen bcc-artigen Phase, und die zweite Stufe dann von der vorhandenen kritischen bcc-artigen Phase zur hcp/fcc-artigen Phase. Dabei ist die Energiebarriere niedriger für kleinere Kontaktenergien des Yukawapotentials und höher für größere Kontaktenergien. Deshalb konnte nur in der Simulation mit der niedrigeren Kontaktenergie der spontane Übergang zur stabilen Phase beobachtet werden, während in der Simulation mit der höheren Kontaktenergie die zweite Energiebarriere nicht

spontan überwunden werden konnte. Dazu benötigt man eine weitere FFS-Simulation, die das System auf eine größere Anzahl von fcc-artigen Partikel optimiert. Dies war aber nur möglich, wenn von den Zuständen gestartet wurde, die die kritischen Cluster beinhalteten, und nicht von dem vollständig bcc-kristallisierten System. Dies bedeutet, dass für den zweiten Übergang die Existenz einer Oberfläche flüssig-bcc Voraussetzung ist. Der Übergang wird durch hcp/fcc-Fluktuationen an der Oberfläche des Kristalls begünstigt, was eine heterogene Kristallisation in der anfangs homogenen Umgebung darstellt.

Mit Hilfe von Untersuchungen der stationären Verteilungen des Ordnungsparameters und Gewichtung mit den FFS-Interface-Übergangswahrscheinlichkeiten in einer Vorwärts- und Rückwärtssimulation, bei der der Kristall wieder aufgelöst wurde, konnte die freie Energielandschaft der Kristallisation analysiert werden. Diese lässt sich gut mit der freien Energielandschaft der klassischen Nukleationstheorie erklären, wenn man eine kleine Verschiebung in Richtung der Kristallgröße zulässt. Dies lässt sich dadurch begründen, dass die wirkliche Größe des Clusters nicht bekannt ist, da die Kristallgröße durch die Schwelle des Ordnungsparameters für die Erkennung der festen Partikel beeinflusst wird. Dabei war die Verschiebung sowohl für niedrige als auch für hohe Kontaktenergien kleiner als ein Partikeldurchmesser. Aus der Anpassung an die Nukleationstheorie konnten Oberflächenspannung und chemisches Potential des Kristallisationsprozesses bestimmt werden, welche gut mit existierenden Werten aus anderen Arbeiten übereinstimmen. Die klassische Nukleationstheorie kann also in beiden Fällen angewandt werden, wobei die zuerst geformte, metastabile Phase zählt. Diese Phase ist immer ein bcc-Kristall, auch wenn die thermodynamisch stabile Phase ein fcc-Kristall ist. In diesem Fall ist die Nukleation ein zweistufiger Prozess, der jedoch nicht die Nukleationsrate oder die Struktur des kritischen Clusters beeinflusst. Die Nuklei sind praktisch rund, sodass Kanten des Kristalls eine untergeordnete Rolle spielen. Weiterhin ist die Kristalloberfläche diffus, was jedoch bereits in der Oberflächenspannung berücksichtigt ist. Die Nukleation ist hauptsächlich die Ausbildung einer sechszähligen Symmetrie, die sich schon beim Einsetzen der Kristallisation in der übersättigten Flüssigkeit zeigt.

Auf dem Weg zu diesen Ergebnissen wurden moderne Methoden optimiert und parallelisiert, Tools zur Analyse geschaffen und damit die Kristallisation von geladenen Makromolekülen ausführlich untersucht. Es wurden also nicht nur Möglichkeiten für viele Forscher geschaffen, weitere schwer zugängliche Ereignisse zu untersuchen, sondern auch die Theorie für das Kristallwachstum, insbesondere für den Nukleationsprozess, einen weiteren Schritt vorangebracht.

Summary

The crystallization of charged macromolecules plays an important role in many fields like biology, medicine, physics and material design. For example, crystallization is used to purify proteins from other ingredients in a solution. Colloidal crystals are promising candidates for photonic crystals in optical applications, and macromolecules have to be crystallized for structure determination with electromagnetic radiation, e.g. X-ray radiation.

For developing a closed theory of crystallization it is indispensable to investigate the underlying microscopical details, in particular to investigate the onset of crystal growth. This process is called nucleation and can be described qualitatively by the Classical Nucleation Theory (CNT). In CNT, crystal growth is described as a balance of surface tension between the crystal cluster and the liquid, and the chemical potential difference between those phases. The contribution of the surface tension is proportional to the surface area of the cluster, and the contribution of the chemical potential is proportional to its volume, which leads to an energy barrier towards nucleation. In principle, CNT can be used for quantitative predictions of the nucleation process. However, this theory is based on simplifying assumptions which lead to discrepancies when applied to real systems, e.g. concerning nucleation rates.

Having a closed theory, targeted crystallization would be possible, and one could create defect-free crystals for optical applications, create crystals with defined features, or prevent crystal growth if desired.

To advance towards a closed physical theory of crystallization and to minimize the deviations of theory and real systems, we investigate the crystallization of charged macromolecules with the help of Molecular Dynamics (MD) computer simulations. In these simulations, we use a 3D simulation box with periodic boundary conditions and an NPT ensemble. The charged macromolecules are modeled by an effective pair potential, the Yukawa potential, which accounts for the electrostatic screening by neutralizing salt ions.

The Yukawa potential can be used to model real colloidal systems like they are used in experiments. This has been shown during this work in collaboration with the 2nd Physical Institute of the University of Stuttgart. We used simulations to compute the pair interaction potential of the macromolecules from the radial distribution function (RDF) of the experimental system. We found that at lower system densities the Yukawa potential fits very well to the interaction potentials obtained from the RDFs. Using this procedure, the screening length of the colloids and the contact value of the interaction potential could be determined. At higher densities, the repulsive part of

the interaction potential could be described by the Yukawa potential, while for large distances, three body interactions play a role. The screening length of the colloids depends on the density of the system, which can be explained by the different number of charges in solution for the different number of molecules per volume, because the charges dissociate from the surface of the colloidal spheres.

To study the crystal growth in detail, it is necessary to simulate near the coexistence line in the phase diagram, where the attachment rate of growth units is low, which leads to a high free energy barrier towards nucleation. Under these conditions, observing nucleation is difficult not only in simulations, but also in experiments, because the waiting time before seeing such a process is very long compared to the process itself.

Recently, several *Rare Event Sampling* methods have been developed to enhance the sampling in the regions of interest. These methods ratchet the system towards the interesting event, and statistical sampling is used to correct for this ratchet-like manner. With that, transition rates and physical pathways can be investigated.

In this work, we use the *Forward Flux Sampling* (FFS) method, where the reaction path through phase space, characterized by an order parameter, is partitioned by a set of interfaces. The advantage of this scheme is that successive transitions between these interfaces can be calculated by conventional computer simulations, which can then be composed to the overall transition event.

For our investigations, it was not only indispensable to parallelize the method, but also to improve the FFS method concerning the efficiency, e.g. the balance of computational effort and statistical error. For the parallel implementation we analyzed the FFS method with diverse simulation problems, e.g. a one-dimensional particle which is placed initially in a potential minimum and has to cross an energy barrier, as well as with more complex problems like vapor bubble nucleation and translocation of polymers through nanopores. Moreover, we developed an algorithm to parallelize the initial simulation run of the FFS method.

The efficiency of the FFS method depends mainly on the positions and hence on the amount of interfaces for the transitions. During my stay abroad at the University of Edinburgh we developed two methods to place these interfaces automatically and at their optimized locations, on-the-fly during simulation. To this aim, we developed an analytical model which describes the dependency of the efficiency on the transition probability, which is the tuning parameter of the interface placement. This analytical model was verified by simulations. Beyond the increase of efficiency, this work leads to a tremendous simplification of the method: Only the initial state (e.g. liquid) and the final state (e.g. a certain number of solid particles in the largest cluster) have to be defined in terms of the order parameter. Then, the simulation finds its way through phase space automatically and optimized.

To make our extensions and parallel implementations publicly available for other research groups, we introduced a framework which contains all these features, our so-called *Flexible Rare Event Sampling Harness System* (FRESHS). FRESHS was developed during this work and which is now further developed in collaboration with

a research group in Luxembourg.

With the help of these precognitions and tools it was possible to investigate the crystallization of charged macromolecules in simulations. For quantifying the progress of the crystallization in the system, we implemented the so-called local bond $\overline{q}_l$ order parameter, which is a local, per-particle property to analyze if the particle can be considered as solid-like or liquid-like. Furthermore, we implemented a cluster algorithm to detect the size of the largest cluster of solid-like particles in the system, which is then used as order parameter for the FFS method. With this system, many FFS simulations have been conducted at different phase points, and the crystallization of charged macromolecules could be directly simulated and be investigated.

The transition rates decrease drastically for smaller pressures, which means approaching the phase coexistence line, and nucleation events become very rare in this case, as expected. The backtracking of different crystallization pathways in the FFS scheme yields the successful crystallization paths. These pathways were analyzed in the post-processing for determining nucleation mechanisms and to identify possible precursors for the onset of crystal growth. We found that at the local position, where the critical cluster will be formed, a local structure of sixfold order can already been discovered in the fluid at an early stage. We did not find correlations with the fourfold symmetry or the local density. Thus, the nucleation is driven by local sixfold order fluctuations, and not by density fluctuations as classically argued.

Depending on the pressure and contact energies of the potential, we arrive at different solid-like structures in the final state B. For lower contact values and higher pressures the structure is an hcp/fcc-like lattice, which is consistent with the thermodynamically stable fcc phase. In contrast, at higher contact values and lower pressures we obtained a bcc-like structure, which is not the stable structure according to the phase diagram.

This can be explained by the fact that the crystallization process is a two-stage mechanism with two energy barriers involved. The first stage is the transition from the liquid phase to the metastable bcc-like phase, and the second stage is the transition from the bcc-like phase to the hcp/fcc-like phase. The energy barrier for the second transition is lower for smaller contact values of the Yukawa potential, and higher for larger contact values. This is the reason, why we could only observe the spontaneous transition to the stable phase for the low contact value case.

In the simulation with the higher contact value, the second energy barrier could not be overcome spontaneously, but requires performing an additional FFS simulation, which optimizes for a higher number of fcc-like particles in the system. Moreover, the transition was only possible when starting from the critical clusters and not from the fully converted bcc-like system. Hence, a condition for the second transition is the existence of a liquid-bcc interface. The process of the transition is facilitated by hcp/fcc-like fluctuations at the surface of the crystal cluster. Thus, we observe a heterogeneous nucleation in the initially homogeneous system.

With the help of analyzing stationary distributions of the order parameter and

weighting with the FFS interface transition probabilities in a forward and a backward simulation, where the crystal cluster was dissolved again, it was possible to calculate the free energy landscape of crystallization. The free energy landscape fits well to CNT, if one allows a small shift for the cluster size. This is justified because the unknown real dimensions of the cluster depend on the (arbitrary) threshold for the solid particle detection in the local bond order parameter analysis. For both, high and low contact values, the shift was smaller than a particle's diameter. From the fitting we obtained the surface tension and chemical potential of the crystallization process, which we compared to previous work. CNT can be applied in both cases considering the first, metastable phase. This phase is always a bcc crystal, even if the thermodynamically stable phase is an fcc crystal. In this case, nucleation is a two-stage process, which however does not influence nucleation rates or the structure of the critical cluster. The nuclei are almost spherical, so that edges of the crystal play a minor role. Also, the crystal surface is fairly diffuse, which however is taken into account by the surface tension. Nucleation is mainly the formation of a sixfold symmetry, which can already be seen at the onset of crystallization in the supersaturated liquid.

On our way, we optimized and parallelized state-of-the-art rare event sampling techniques, created powerful analysis tools and investigated the crystallization of charged macromolecules in great detail. In this work, we therefore created not only the possibility for many researchers to investigate rare events which are difficult to access, but we also pushed the theory for colloidal crystal growth, notably for the nucleation process, one step further.

1 Introduction

Many phenomena in nature are related to crystallization, from the formation of mineral crystals, complex processes in organisms to the general formation of droplets in many systems [1]. Beyond the phenomena in nature, the crystallization process is important for many applications, e.g. photonic crystals where the optical properties are determined by the underlying crystal structure [2, 3], protein purification where proteins can be extracted from a mixture by crystallizing them [4], and structure determination via X-ray scattering which is very important for chemical physics and medicine [5, 6].

For all these applications and from a fundamental point of view a closed picture of the crystallization process is desirable, in particular for systems consisting of charged components [7]. Since a long time it is known that crystals grow by accretion, where growth units attach under appropriate conditions, which is called nucleation at the early stage, when "a crystal is born from its mother solution" [8].

Despite of important recent advances in this field [8, 9, 10, 11, 12, 13, 14], crystal growth is a complex process and is not yet fully understood: "However, even when pure soluble protein is available, producing high-quality crystals remains a major bottleneck in structure determination" [6]. Hence, in experiments crystals are often grown by an empirical trial and error approach, where hundreds of different conditions are tested until a certain crystal is obtained [15], which can be seen as more an art rather than a technique because of the non-complete understanding of the underlying physical mechanisms. The problem is, that the exact conditions must be met under which nucleation takes place, which is mainly at low supersaturations.

Colloidal nucleation can be described to some extent by the Classical Nucleation Theory (CNT) [16]. However, discrepancies exist when comparing with real systems, e.g. concerning the nucleation rates [11], because in real systems e.g. long-ranged interactions can play a role, as well as mixtures of different crystal structures. According to CNT, the nucleation process is a balance of a surface term and a bulk term which leads to an energy barrier towards crystal growth which is higher for lower supersaturations. This makes the nucleation process a rare event in such cases, which is difficult to study.

In general, physical phenomena like crystallization processes can be studied with the help of experiments, theoretical approaches and computer simulations (Fig. 1.1). In the ideal case, all these approaches are combined to obtain a maximal descriptive picture.

In this work we use Molecular Dynamics (MD) computer simulations together with

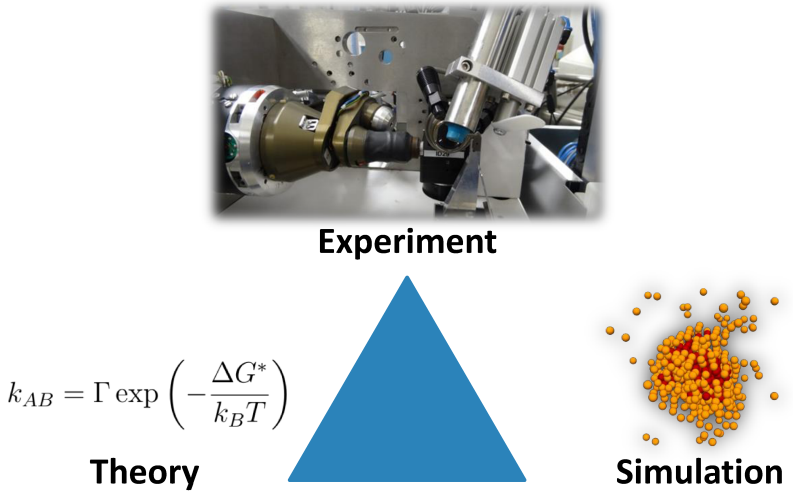

Figure 1.1: The pyramid of research for understanding nature: Experiments, theory and simulations are used to explain physical phenomena. Thereby, all these pillars interact to obtain a picture of maximal clarity. In this illustration, the theory is a formula to quantify crystal growth, the simulation shows solid particles with a crystal cluster, and the experiment is a scattering setup to determine the crystal structure of e.g. protein crystals via X-Ray scattering. The photo of the experiment was taken at a visit of the European Synchrotron Radiation Facility in Grenoble, France.

theoretical models to investigate the crystallization process of – generally speaking – charged macromolecules from the bulk phase. The charged macromolecules are a generic coarse-grained model and can represent colloidal particles[1] carrying surface charges as well as proteins or DNA, where the functional groups are dissociated in solution. We use this generic model of charged macromolecules to transfer our findings to different systems of e.g. different length scales. In addition, we also compare the simulation model of charged macromolecules to an experimental colloidal system to verify the practicability. All the simulations are performed using the open source software package ESPResSo [17].

Simulations have the great advantage that particles, which are e.g. growth units of a crystal cluster, can be tracked directly, physical quantities can be calculated from the simulations because a lot of details are known, and we can apply statistical sampling methods to observe unlikely events which are practically never observed in a real system because of the long waiting time towards the event.

The last point is very important for this work because we are interested in investigating the crystallization process from the parent phase, where a small cluster is nucleated from the parent phase. To this aim, we have to simulate at conditions

[1] In experiments, colloidal particles made of Polystyrene are often used for the investigations.

where the attachment rate is low, namely at low supersaturations of the fluid. This means, that we have to wait for spontaneous fluctuations to form a small crystal cluster, which is very unlikely at these conditions without any wall or artificial impurities present.

With conventional brute-force computer simulations such rare events are difficult to investigate, if even possible at all. Recently, several methods have been developed to succeed in this challenge, e.g. umbrella sampling [18, 19], Bennett-Chandler methods [19, 20, 21], transition path sampling [22, 23, 24, 25, 26], transition interface sampling [27, 28, 29], milestoning [30, 31, 32], nudged elastic band [33, 34], string methods [35, 36], weighted-ensemble methods [37], non-equilibrium umbrella sampling or 'splitting'-type methods [38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50].

In this work, we use the Forward Flux Sampling (FFS) method [39, 40, 42] from the splitting family, which is based on calculating many trajectory fragments which are then combined to the overall result.

By using FFS, transition rates and pathways can be investigated. In addition, with the help of a backward simulation run, stationary distributions and free energy landscapes can be obtained with FFS [43].

On our way towards investigating the crystallization process we encounter the following contents: In chapter 2 we summarize the current state of the art and introduce the theoretical background which is necessary for this work as well as the simulation model for charged macromolecules which is based on a screened Coulomb potential. For our investigations and for FFS an order parameter has to be implemented to the simulation tool ESPResSo for characterizing the progress of crystallization. We use the largest cluster size of solid particles as order parameter in which solid particles are detected via a so-called local bond order parameter.

Since the FFS method is described in a serial way in literature, we present the parallel implementation in chapter 3 as well as further improvements and optimizations of the Forward Flux Sampling method for a tremendous increase of the efficiency, which was mainly developed in collaboration with Rosalind Allen during a stay abroad at the University of Edinburgh, Scotland.

In chapter 4 we present our *Flexible Rare Event Sampling Harness System* (FRESHS), which we developed from scratch during this thesis. Many rare event sampling methods, mainly of the 'splitting'-type family, can be implemented in the context of this framework. For our work, we implement FFS in a highly efficient, parallel way to make the crystallization of charged macromolecules computationally possible at all.

In order to verify the applicability of our simulation model we compare experimental data with our simulations in chapter 5, where we identify the appropriate pair interaction with its screening length and contact values for reproducing a given radial distribution function of the experimental systems at different densities.

We present the results of the crystallization simulations at low supersaturations in chapter 6 which we perform with the help of our optimized FFS method together with FRESHS. Therefore, we simulate on high performance computing hardware. In the

post-processing we analyze crystallization rates and physical pathways to investigate crystallization mechanisms. In addition, we identify possible precursors for nucleation, and compare our direct simulation results to the Classical Nucleation Theory by calculating the free energy landscapes via a stationary distribution analysis.

Finally, in chapter 7, we summarize and discuss the findings of this work and give an outlook on possible future work.

Publications

The following publications are related to this thesis:

- Yevgen Dorozhko, Kai Kratzer, Yuriy Yudin, Axel Arnold, Colin W. Glass and Michael Resch. – "Rare Event Sampling using the Science Experimental Grid Laboratory". Civil-Comp. CC2013/2013/00402 (2013)
- Kai Kratzer, Axel Arnold, Rosalind J. Allen – "Automatic, optimized interface placement in forward flux sampling simulations". J. Chem. Phys. 138, 164112 (2013).
- Dominic Roehm, Kai Kratzer and Axel Arnold – "Heterogeneous and Homogeneous Crystallization of Soft Spheres in Suspension". High Performance Computing in Science and Engineering '13, pages 33-52. Editors: Nagel, Wolfgang E. and Kroener, Dietmar H. and Resch, Michael M., Springer International Publishing (2013).
- Kai Kratzer, Joshua T. Berryman, Aaron Taudt, Johannes Zeman and Axel Arnold – "The Flexible Rare Event Sampling Harness System (FRESHS)". J. Comp. Phys. Comm. 185(7), 1875-1885 (2014).
- Kai Kratzer, Dominic Roehm and Axel Arnold – "Homogeneous and Heterogeneous Crystallization of Charged Colloidal Particles". High Performance Computing in Science and Engineering '14, Springer International Publishing (2014).
- Kai Kratzer and Axel Arnold – "Two-stage crystallization of charged colloids at low supersaturations", arXiv:1410.8695 (2014).

2 State of the art

In this chapter we address the current state of the art by proceeding as follows: First, we introduce the framework of statistical mechanics. The next part is about the simulation technique for sampling the dynamics. Then, we present the model for charged macromolecules, the crystallization theory, and finally the theory about rare events.

2.1 Statistical mechanics

We are interested in the collective properties of the particles in our system according to *statistical mechanics* [19, 51]. In this section we follow Ref. [19] for the description. From a theoretical point of view[1], a system can be in a certain state $|i\rangle$,

$$H|i\rangle = E_i|i\rangle, \tag{2.1}$$

with the *Hamiltonian* H and the energy E_i of state $|i\rangle$. In our case, with many degrees of freedom, the degeneracy of energy states is very large. A system of N particles, volume V and energy E has

$$\Omega(E, V, N) \tag{2.2}$$

eigenstates. A basic assumption of statistical mechanics is that such a system is equally likely to be found in one of the eigenstates $\Omega(E)$. If we consider two subsystems with E_1 and E_2 and $E_1 + E_2 = E$ and we fix E_1, the degeneracy is $\Omega_1(E_1) \times \Omega_2(E_2)$ or in an additive notation, $\ln \Omega(E_1, E - E_1) = \ln \Omega_1(E_1) + \ln \Omega_2(E - E_1)$. The most likely value of E_1 is obtained by maximizing $\ln \Omega(E_1, E - E_1)$, hence with

$$\beta(E, V, N) \equiv \left(\frac{\partial \ln \Omega(E, V, N)}{\partial E} \right)_{N,V} \tag{2.3}$$

$$\Longrightarrow \beta_1(E_1, V_1, N_1) = \beta_2(E_2, V_2, N_2). \tag{2.4}$$

If Eq. (2.4) is fulfilled, there is no net energy transfer between the subsystems which describes thermal equilibrium. This implies that using the second law of thermodynamics, in thermal equilibrium the entropy

$$S(E, V, N) \equiv k_B \ln \Omega(E, V, N) \tag{2.5}$$

[1]Note, that here we use the elegant formulation of quantum mechanics, but we use it only for the description of the basic laws of statistical mechanics. Our simulations are all non-quantum.

of a composed system is maximal. k_B in this relation is the Boltzmann constant[2]. From thermodynamics we know that

$$\frac{1}{T} = \left(\frac{\partial S}{\partial E}\right)_{V,N}, \text{ hence} \tag{2.6}$$

$$\beta = \frac{1}{k_B T}. \tag{2.7}$$

We can now couple a system A to a heat bath B, fixing $E = E_A + E_B$. A is now prepared in state i with energy E_i. For the bath it follows that $E_B = E - E_i$ with the degeneracy Ω_B. The probability of finding the system in state i is then given by

$$P_i = \frac{\Omega_B(E - E_i)}{\sum_j \Omega_B(E - E_j)} = \frac{\exp\left(-E_i/k_B T\right)}{\sum_j \exp\left(-E_j/k_B T\right)}, \tag{2.8}$$

which is the *Boltzmann distribution*. The right hand side of equation (2.8) was obtained via an expansion of $\ln \Omega_B(E - E_i)$ around $E_i = 0$.

For example, the average energy $\langle E \rangle$ of a system with temperature T can then be calculated by

$$\langle E \rangle = \sum_i E_i P_i = -\frac{\partial \ln Q}{\partial 1/k_B T} \tag{2.9}$$

with the partition function $Q = \sum_i \exp\left(-E_i/k_B T\right)$. With the thermodynamic equation $E = (\partial F/T)/(\partial 1/T)$ we see in comparison with Eq. (2.9) that the Helmholtz free energy F is related to the partition function by

$$F = -k_B T \ln Q = -k_B T \ln \sum_i \exp\left(-E_i/k_B T\right), \tag{2.10}$$

which is the "workhorse of equilibrium statistical mechanics" [19].

Observables

In our work we are interested in measuring different observables. In general, using classical statistical mechanics a thermal averaged value of an observable A can be obtained with

$$\langle A \rangle = \frac{\int d\mathbf{p}^N d\mathbf{r}^N \exp\left\{-\beta \left[\sum_i p_i^2/(2m_i) + U(\mathbf{r}^N)\right]\right\} A(\mathbf{p}^N, \mathbf{q}^N)}{\int d\mathbf{p}^N d\mathbf{r}^N \exp\left\{-\beta \left[\sum_i p_i^2/(2m_i) + U(\mathbf{r}^N)\right]\right\}}. \tag{2.11}$$

In the next section we present a way to calculate such observables via sampling of the ensembles.

[2] $k_B = 1.38066 \times 10^{-23} m^2 kg s^{-2} K^{-1}$

2.2 Molecular Dynamics Simulations

We use the *Molecular Dynamics* (MD) simulation technique to answer the questions from the introductory chapter by simulating the dynamics of the particles and sampling the particular observables. In contrast to other techniques like Monte Carlo (MC), the MD simulation method is suited for tracking particles in each simulation step which is desirable e.g. for investigating crystallization mechanisms. Thus, MD simulations are related in many details to real experiments [19].

To measure observables in MD simulations, the observables must be expressed as a function of positions and momenta of the particles in the system. The temperature T in a simulation can be defined via the average of the kinetic energy per degree of freedom N_f via

$$\left\langle \frac{1}{2}mv^2 \right\rangle = \frac{1}{2}k_B T, \tag{2.12}$$

with the mass m and velocity v of a particle. Because of the fluctuations of the kinetic energy in simulations, the instantaneous temperature also fluctuates,

$$T(t) = \sum_{i=1}^{N} \frac{m_i v_i^2(t)}{k_B N_f}, \tag{2.13}$$

with the sum running over all N particles i in the simulation.

2.2.1 Ergodicity

Above we have used the ensemble average of multiple states. However, in experiments one usually doesn't prepare multiple setups and performs only a single measurement. In contrast, many measurements are conducted in a time series. The same is true for MD simulations: We would like to study the average behavior of a system by computing the time evolution and by averaging the investigated quantities. Hence, we assume that the average of the states is equal to the average of the time series,

$$\overline{\rho_i(r)} = \langle \rho_i(r) \rangle, \tag{2.14}$$

where $\rho_i(r)$ is the density at a distance r of an atom i in a simulation as an example. The bar denotes a time average and $\langle ... \rangle$ denotes an ensemble average. This is called *ergodicity hypothesis* and is plausible for a lot of systems. But, there are also systems which can't be considered to be ergodic, then Eq. (2.14) doesn't hold. The ensemble average is usually calculated using MC simulations and the time average using MD simulations.

2.2.2 Force calculation, equation of motion and ensembles

In MD simulations, the position and velocity of many-body systems are derived from the force acting on them in an iterative way according to classical mechanics (see

also [19, 51]). Forces can be of different nature, like interaction forces or external forces (e.g. an electric field). The interaction forces $\mathbf{f}$ per particle pair are derived from the interaction potential $U(r)$ with the absolute value of the particle-particle connection vector $|\mathbf{r}| = r = \sqrt{x^2 + y^2 + z^2}$ via

$$f_x(x) = -\frac{\partial U(x)}{\partial x}, \tag{2.15}$$

here exemplarily shown for the x-dimension. The force calculation is the most demanding part in an MD simulation, usually most of the computational effort is spent in this calculation. This arises from the fact that for N particles we must compute the forces of $N(N-1)/2$ pair distances which scales like $O(N^2)$. For details how to implement this efficiently e.g. by using *Verlet lists* with the aim that this calculation doesn't scale with $O(N^2)$ but rather with $O(N)$ refer to [19].

At this point, the forces on the particles are known, hence we are able to introduce Newton's equation of motion

$$\mathbf{f} = \dot{\mathbf{p}} = m\mathbf{a} \stackrel{!}{=} -\nabla U(r), \tag{2.16}$$

with the mass m, acceleration $\mathbf{a}$ and momentum $\mathbf{p}$ of the particle. Note, that we use the *Newtonian* scheme in this case, but an *Hamiltonian* description of the equations of motion would be also possible [51]. From equation (2.16) the position $x(t + \Delta t)$ and velocity $v(t + \Delta t)$ can be calculated by numerical integration schemes. This will be covered in the next section.

Numerical integration

To propagate the particles from timestep t to $(t + \Delta t)$ the equations of motions for the particles must be integrated. In our self-written simulations we use the *Velocity Verlet*[3] integration scheme. We use this scheme for our purposes because it is symplectic[4] and has a good numerical stability and accuracy with an error $\propto O(\Delta t^4)$. In this scheme, the new position x and the new velocity v of a particle is calculated using the following equations which show the one dimensional case [19]:

$$x(t + \Delta t) = x(t) + v(t)\Delta t + \frac{f(t)}{2m}\Delta t^2, \tag{2.17}$$

$$v(t + \Delta t) = v(t) + \frac{f(t + \Delta t) + f(t)}{2m}\Delta t. \tag{2.18}$$

$$\tag{2.19}$$

The force $f(t + \Delta t)$ is derived at the new position $x(t + \Delta t)$ calculated in Eq. (2.17). Note, that $f(t)/(m) = a(t)$ is the acceleration.

[3] For an overview of different integration schemes please refer to [19].

[4] The energy performs a random walk around its arithmetic average.

It follows, that the dynamics of the system is determined by the forces acting on the particles. These forces can't only be interaction forces or external forces but also random temperature fluctuations, e.g. implemented in the context of a thermostat.

Thermostat and barostat

In MD simulations, thermostats and barostats are used to generate certain statistical ensembles by monitoring physical quantities like pressure p and temperature T, and correct these quantities such that these values fluctuate around a presetting. A typical thermostat to model temperature fluctuations similar to a real system is the *Langevin* thermostat. For the Langevin dynamics, two additional terms are added to Eq. 2.16, one for the friction in the system and one for the random temperature fluctuations:

$$f(x) = m\ddot{x} = \underbrace{-\nabla U(x)}_{\text{force}} - \underbrace{\xi m\dot{x}}_{\text{friction}} + \underbrace{\mathcal{R}(t)\sqrt{2m\xi k_B T}}_{\text{random fluctuations}}, \tag{2.20}$$

where ξ is the friction coefficient and $\mathcal{R}(t)$ is a Gaussian random process with $\langle \mathcal{R}(t) \rangle = 0$ and $\langle \mathcal{R}(t)\mathcal{R}(t') \rangle = \delta(t - t')$. In this equation a fluctuation is added to the dynamics which can either be positive or negative, and the dynamics is damped by a friction force.

The barostat for a constant pressure in the simulation works by adapting the volume of the simulation box and rescaling of the particle coordinates according to the pressure changes. This leads to a new Lagrangian expression for the system [52]. The interpretation of the additional terms in the new Lagrangian can be seen as a piston of a certain mass acting on the isotropic adaptive system. In the limit of an infinite piston mass, the original system is obtained. For another piston mass, the averages of the observables of the system correspond to the time averages of the thermodynamic ensemble at the desired pressure [52]. Thereby, the dynamics of the volume fluctuations depends on the mass of the piston. For further details and barostat examples refer to [19, 52, 53, 54, 55].

Our main simulations are performed using ESPResSo [17], where most of the simulation techniques are already implemented. For our self-written codes we use the techniques described in this chapter.

2.2.3 Reduced units

For our simulations we use so-called *reduced units*. The practical reason for this is that we would like to have all quantities in an order of magnitude to be numerically friendly, which reduces the numerical issues in a simulation[5].

A great advantage of reduced units is, that they can be adapted easily to a certain scale afterwards, which means that we perform our simulations in the most general

[5] In SI units we would permanently multiply values which are much less or much larger than 1 which can lead easily to numerical instabilities.

way and are able to insert dimensions like [μm] afterwards to e.g. compare with experiments. We use the following basic units in this work:

- ϵ, the unit of energy,
- m, the unit of mass,
- σ, the unit of length.

The unit of time is then given as $\sigma\sqrt{m/\epsilon}$ and the unit of temperature is ϵ/k_B. We define the following reduced units: The reduced potential $U^* \equiv U\epsilon^{-1}$, pressure $p^* \equiv p\sigma^3\epsilon^{-1}$, temperature $T^* = k_BT\epsilon^{-1}$ and density $\rho^* = \rho\sigma^3$ [19]. With the help of these reduced units we are able to relate an arbitrary number of combinations ϵ, ρ, σ and T to the same simulated state, which is called *law of corresponding states* [19].

2.2.4 Inverse Boltzmann: Pair potential from an RDF

The Radial Distribution Function (RDF) is used to characterize the local structure of a system and can be measured in both, experimental systems[6] and simulations. Therefore, it is often consulted to compare simulations with experimental data and vice versa.

The RDF $g(r)$ is determined by the average number density $\rho(r)$ at a certain distance r from a given atom compared to an *ideal gas* with the same $\rho(r)$. From a given $g(r)$, the local density $\rho(r)$ of the system can be obtained via $\rho(r) = \rho g(r)$. For details about the calculation of $g(r)$ refer to [19]. Note, that $g(r) = 1$ for an ideal gas, and $g(r) \neq 1$ is an indication for pair interactions. For a typical system (e.g. with a Lennard-Jones interaction) we expect therefore the RDF to be zero at small values of r, because of the strong repulsion of the potential at these distances, and in the following we obtain peaks and minima at distances where it is likely or unlikely for the system to find particles at a given r, and the RDF ends on a value of 1 for large distances r. Examples of RDFs can be found in chapter 5.

With the *Inverse Boltzmann* method [56, 57] a pair interaction potential can be reconstructed from a given (experimental) RDF $g_{\text{exp}}(r)$ in an iterative way. This is based on the fact, that particles arrange e.g. during equilibration according to their pair potential. From this arrangement the RDF is generated, therefore it is nearby to reconstruct the potential from an RDF. For the Inverse Boltzmann method we start with an initial guess for the potential generated from $g_{\text{exp}}(r)$ with

$$U_0(r) = -k_BT \log g_{\text{exp}}(r). \tag{2.21}$$

Then, we equilibrate our simulation system using this potential[7] and generate the corresponding $g_i(r)$. From the new $g_i(r)$ we calculate the correction of $U(r)$ in an

[6] e.g. from data obtained by scattering experiments or by microscopical imaging.

[7] A tabulated potential is well suited for this purpose. Otherwise a fit of the expected potential for generating the RDF could be tried for example.

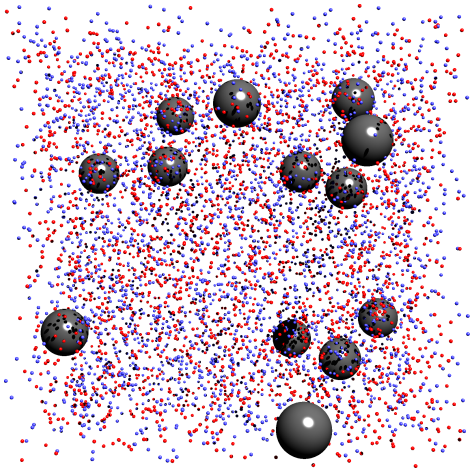

Figure 2.1: Simulation box filled with different particles: Macromolecules (dark), and neutralizing salt-ions with opposite charges (red and blue).

iterative procedure via

$$U_{i+1}(r) = U_i + k_B T \log\left(\frac{g_i(r)}{g_{\text{exp}}(r)}\right) \tag{2.22}$$

until the desired accuracy is reached. This iterative procedure corrects the potential in a way that particles are more attracted at distances where the target RDF would require more particles, and are less attracted at distances where less particles are expected. This means, that a potential reproducing the exact RDF is a fixed point for this procedure. The Inverse Boltzmann method will be applied to an RDF from an experimental system to obtain the corresponding pair interaction potential in chapter 5.

2.3 Model for charged macromolecules

In this section we address the simulation details of charged macromolecules. If we model the macromolecules as particles on a computer, a typical simulation scenario could look like the one in Fig. 2.1, which shows a simulation box with many particles. In this simulation scenario there are larger particles which represent the macromolecules and smaller particles of two different colors, which represent the oppositely charged neutralizing salt. Note, that this is an example image for a situation where we just put the ingredients in a simulation box and would like to figure out what happens. We are interested in the dynamics of that system, more precisely if we are now at timestep t, how does the system look like at $t + \Delta t$? In section 2.2 we have discussed the numerical methods to perform this intention in a simulation, therefore we will now have a look at the interaction forces for our model which influence the

dynamics. We would like to simulate a charged system, which means that forces arise from electrostatic interactions of the charges. These forces are called *Coulomb forces*. In the following sections we present a generic model, which can be used for charged macromolecules like proteins and colloids and captures the influence of long-range electrostatics.

2.3.1 Coulomb electrostatic interaction in different media

We present the details of the electrostatic interaction following the book of Ref. [58]. First, we consider the general case in vacuum, where nothing hinders the propagation of the electrical field. Then we discuss the Coulomb interaction in a polarizable medium like water, and finally tackle the case where the electrostatic interactions are screened by additional charges like salt ions.

Coulomb interaction in vacuum, unscreened

Two charges Q_1 and Q_2 in vacuum with a distance r have the Coulomb interaction potential

$$U_{\mathrm{C,vacuum}}(r) = k\frac{Q_1 Q_2}{r} \tag{2.23}$$

with

$$k = \frac{1}{4\pi\epsilon_0} = 8.99 \times 10^9 N m^2 C^{-2}, \tag{2.24}$$

where ϵ_0 is the *vacuum permittivity*. If we calculate the force $f(r)$ by the derivative of Eq. (2.23), we see that $f(r) \propto r^{-2}$. From the sign of Q_1 and Q_2 it is determined, whether the force is attractive or repulsive: Different charges are attractive and same charges are repulsive. We see that in the vacuum case the interaction is long-ranged because of the slow decay $\propto r^{-1}$ (Fig. 2.2(a)). The field lines are not terminated in this case and head for infinity. For biological, chemical and physical applications in vacuum this would mean that we would have to superpose many of these fundamental long-ranged Coulomb interactions, which is not very practicable in simulations when thinking of the absolute number of interaction pairs, e.g. in a periodic simulation box. Luckily, many applications (including our project) are not situated in vacuum.

Coulomb interaction screened by a polarizable medium

Charges embedded in a dielectric material interact also via the Coulomb interaction. In contrast to the vacuum case, their interaction strength is reduced by the *dielectric constant* ϵ_r,

$$U_{\mathrm{C,dielectric}}(r) = \frac{k}{\epsilon_r}\frac{Q_1 Q_2}{r}. \tag{2.25}$$

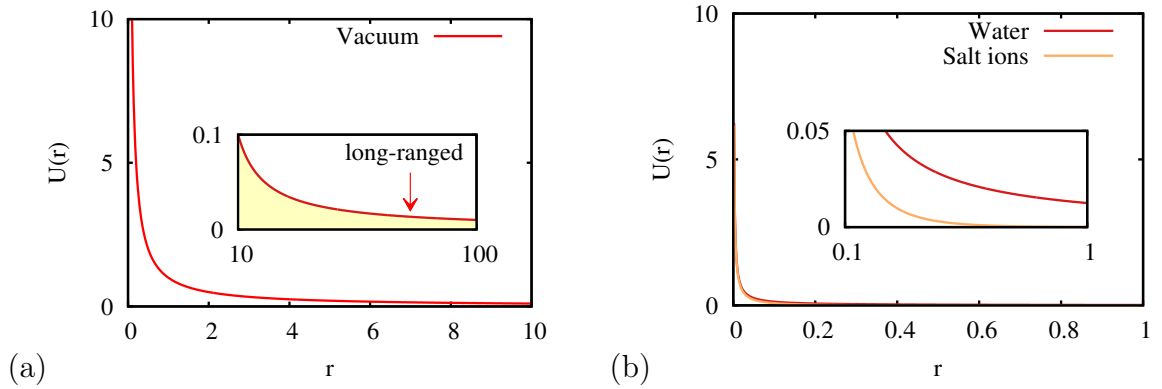

Figure 2.2: (a) Coulomb interaction potential in vacuum, the inset shows that the potential is long-ranged. (b) Coulomb potential for water ($\epsilon = 80$) and for the Debye-Hueckel interaction, where the potential is screened by salt ions. In both cases, the interacting potential is much smaller than in (a). In addition, the Debye-Hueckel potential decays very fast to zero as shown in the inset, which is a good model for screened interactions, where the field-lines are terminated at the counter-charges.

Here, the screening is possible due to the polarizability of particles in the dielectric medium. Thereby, oriented dipoles are induced and some of the field lines are terminated at other charges. In Eq. (2.25) the interaction strength is reduced by a constant factor ϵ_r. However, this is still a long-ranged interaction (Fig. 2.2(b)), but depending on the value of ϵ_r this potential can be cut at an earlier point with a lower error in contrast to the vacuum case.

As an example, pure water at a temperature of $T = 298K$ has a value of $\epsilon_r \approx 80$, which means that the Coulomb interaction is reduced by a factor of 80 relative to vacuum. In this case, every water molecule carries a permanent dipole moment which reduces the pure interaction.

Bjerrum length

The Bjerrum length is the distance of two charges $r = l_B$, where the two unit charges e have an interaction energy of $U = k_B T$. If we put this in Eq. 2.25, we obtain the Bjerrum length via

$$\frac{k}{\epsilon_r}\frac{e^2}{l_B} \stackrel{!}{=} k_B T, \tag{2.26}$$

$$\Longrightarrow l_B = \frac{k}{\epsilon_r}\frac{e^2}{k_B T}. \tag{2.27}$$

As an example, the Bjerrum length of two electrons with a charge $e = 1.602 \times 10^{-19} C$ in water with $\epsilon_r = 80$ is $l_B = 7.0 \times 10^{-10} m = 0.7 nm$. This means that the interaction

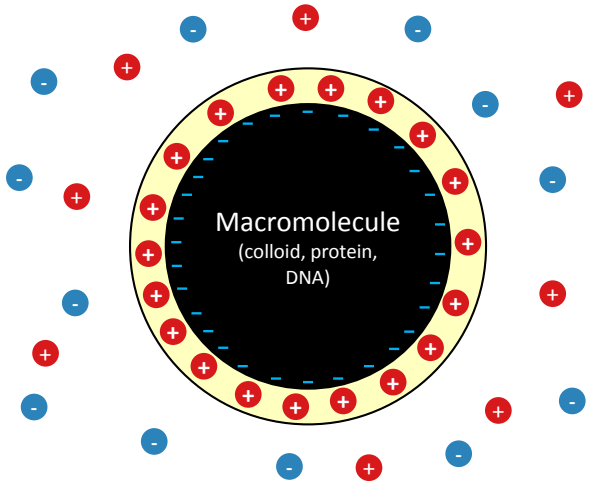

Figure 2.3: Screening of the surface charges of a macromolecule in solution. The negative surface charges of the macromolecule are immediately screened by counter-ions which accumulate at the surface. Outside of this area the interacting potential is mainly determined by the effective ion background.

energy is $U = k_B T$ at an electron distance of 7Å.

Using the Bjerrum length, the electrostatic potential conveniently reduces to

$$U(r) = l_B k_B T \frac{Q_1 Q_2}{r}. \tag{2.28}$$

Coulomb interaction screened by salt ions

Now, we arrive at the very important electrostatic interaction case for biological, chemical and physical applications. In these applications, the interaction is practically always screened and reduced in strength by the presence of other molecules. For example, if we consider a macromolecule which can be e.g. a colloidal sphere in solution which carries a negative surface charge, this negative surface charge is immediately screened by counter-ions from the environment (Fig. 2.3).

Outside this area the potential is mainly determined by the resulting effective ion background. In contrast to the case in the previous section, the screening here is not only a reduction of the Coulomb interaction by a constant factor, but an exponential decay of the interaction potential beyond a distance which is called *Debye screening length* l_D. This length l_D determines, at which distance the exponential scales down the Coulomb interaction and can be calculated for an electrically neutral system by

$$l_D = \left(\frac{\epsilon_0 \epsilon_r k_B T}{\sum_{i=1}^{N} n_j^0 Q_i^2} \right)^{-1/2} \tag{2.29}$$

with the number of different species of charges N, the charge of the i-th species Q_i, and the mean concentration of charges n_i^0. As one can see from this equation, the screening length l_D decreases when increasing the number of ions.

The resulting screened Coulomb potential is called *Debye-Hückel* or *Yukawa* potential[8]. Using equation (2.27) this potential has the form

$$U_{\mathrm{DH}}(r) = l_B k_B T \frac{\exp(-\kappa r)}{r}. \tag{2.30}$$

with the inverse screening length $\kappa = {l_D}^{-1}$. For a visualization see Fig. 2.2(b).

This model can be applied to simulate charged macromolecules as they appear in biophysical soft-matter applications with charges being screened by neutralizing salt ions (see also Fig. 2.3), e.g. many nucleic acids, proteins and other macromolecules are slightly charged when they are put into water. The ions originate from the dissociation of functional groups which are then screened by the surrounding salt. The size of the surrounding cloud is approximately the screening length l_D. This is often called *electric double layer* [59, 60].

2.3.2 The hard core Yukawa potential

The potential for our simulations of the crystallization of charged macromolecules is the *Yukawa* potential and based on the potential described by Eq. (2.30). We choose this potential because it was successfully applied to colloidal systems [9, 61, 62, 63] as well as to be able to compare our results with previous works (e.g. with Ref. [9]).

In addition, we add a *Weeks-Chandler-Andersen* (WCA) potential [64], which is a shifted *Lennard-Jones* potential with a cutoff $2^{(1/6)}\sigma$ and hence without the attractive part, to model the excluded volume of the macromolecules. This WCA potential as a smaller cutoff than the Yukawa potential and therefore only plays a role if the macromolecules come close to each other. Then, this potential avoids overlapping of the macromolecules, as it is the case in real experiments.

The resulting potential for our simulation is a combination of both potentials,

$$U(r) = U_{\mathrm{Yukawa}}(r) + U_{\mathrm{WCA}}(r) \tag{2.31}$$

with

$$U_{\mathrm{Yukawa}}(r) = \epsilon \frac{\exp(-\kappa(r/\sigma - 1))}{r/\sigma} \tag{2.32}$$

and

$$U_{\mathrm{WCA}}(r) = \begin{cases} 4\left(\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6} + \frac{1}{4}\right) & r < 2^{\frac{1}{6}}\sigma \\ 0 & \text{else.} \end{cases} \tag{2.33}$$

[8] Eq. (2.30) is called the Yukawa form of the Debye-Hückel potential.

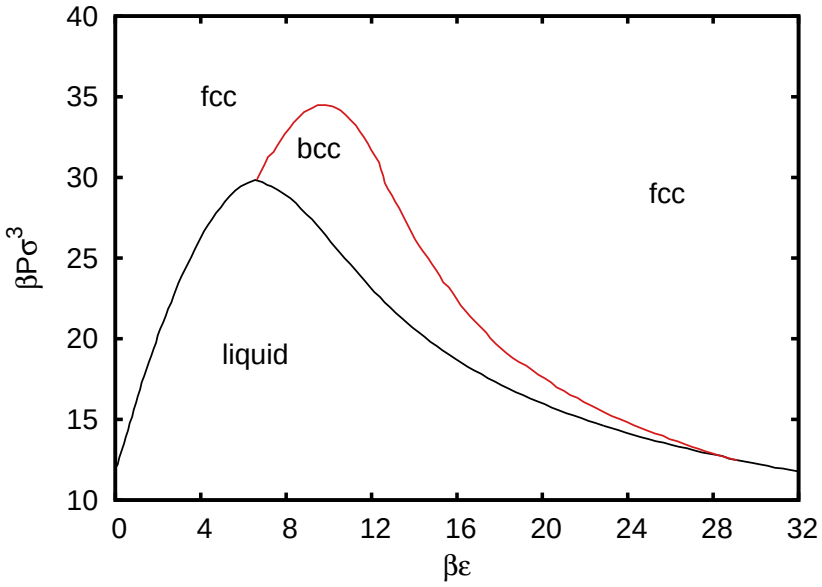

Figure 2.4: Phase diagram of the Yukawa potential for $\kappa = 5$, data from [62]. The phase diagram consists of three regions in this case: a liquid region, and two solid regions, namely a solid region where the system crystallizes to a bcc or fcc lattice. In addition, there are two triple-points where all three phases can coexist.

From a practical point of view, the Lennard-Jones and the Debye-Hueckel potential are already built-in potentials in our simulation tool ESPResSo, therefore we use these potentials and set the cutoff to the WCA cutoff for the former one and the Bjerrum length for the latter one such, that it matches our potential in Eq. (2.32), $l_B = \epsilon\sigma e^{\kappa}$.

2.3.3 Phase diagram of the Yukawa potential

The Yukawa potential which was introduced above has a phase diagram which is depicted for a screening length of $\kappa = 5$ in Fig. 2.4 [62]. As can be seen in the phase diagram, the system has three phases in this case, a liquid phase and two solid phases. The solid phases can have a body-centered cubic (bcc) or a face-centered cubic (fcc) lattice.

There are also two *triple-points*, where all three phases can coexist. In our simulations we try to build crystal clusters from fluctuations as close as possible to these phase coexistence lines. Note, that the size of the solid domain with the bcc structure is determined by the location of the triple points. For higher values of κ, the size of this domain shrinks [62].

Before we discuss the crystallization theory in section 2.4, we discuss briefly the computational considerations and physical limitations of this model.

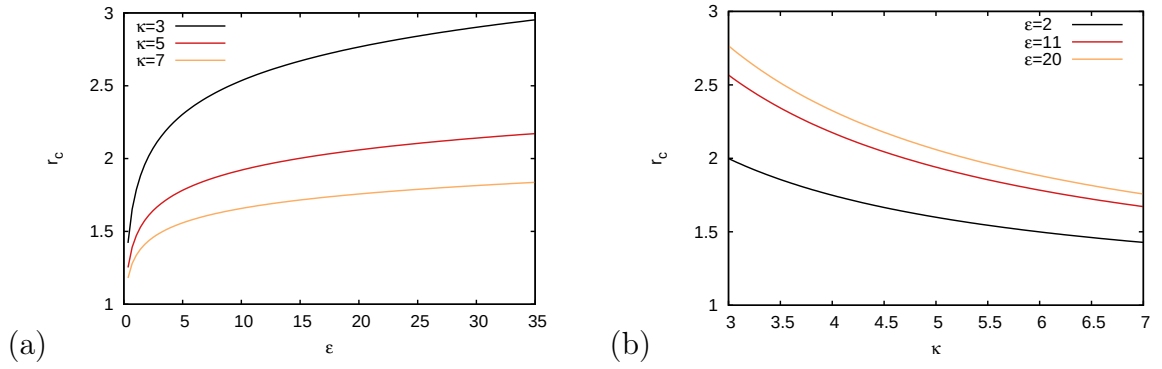

Figure 2.5: Cutoff trend for $r_{\text{cut}}(\kappa, \epsilon)$: (a) Varying ϵ and (b) varying κ. If the contact value ϵ is increased, the potential must be cut at higher distances for the same accuracy and if the inverse screening length κ is increased, the cutoff is lower for the same accuracy.

2.3.4 Computational considerations and physical limitations

The main quantity which can be changed for the computational effort is the cutoff of a long-ranged potential. On the one hand, choosing a higher cutoff results in a better accuracy but more particle pairs must be considered. In Sec. 2.2 we mentioned that calculating the interaction forces is the part which takes most of the computation time, so this increases the computational effort tremendously. On the other hand, a cutoff which is too low leads to defective results.

Choosing a cutoff for the screened Coulomb interaction

The screened Coulomb potential decays much faster than the plain Coulomb interaction (see also Fig. 2.2). However, it has still an asymptotic behavior for $r \to \infty$. For the computer simulations it is necessary to cut the potential at a certain distance r_{cut}. In our case, for the potential in Eq. (2.32) this cutoff depends on the screening length κ and the contact value ϵ of the potential, because the combination of these two values determines the shape and decay of the potential. We choose the cutoff value such, that the error in energy is below a certain value, with

$$r_{\text{cut}}(\kappa, \epsilon) = -\frac{1}{\kappa}\left[\log\left(\frac{\epsilon_c}{\epsilon}\right) - \kappa\right], \tag{2.34}$$

where ϵ_c can be set to e.g. 0.1ϵ. The trend for $r_{\text{cut}}(\kappa, \epsilon)$ in the range where we simulate is shown in Fig. 2.5. It can be seen, that the cutoff varies between approximately 1.5 and 3, which can be tackled with our simulations and the available computation power.

Pressure tail corrections

If the pair potential is cut at a certain distance r_{cut}, the measurement of the pressure in the simulation is defective and can be corrected via a pressure tail correction [19]. For the Yukawa potential this leads to

$$\Delta P_{\text{tail}} = \frac{2\pi\rho^2}{3}\int_{r_{\text{cut}}}^{\infty} dr r^2 \mathbf{r}\cdot\mathbf{f}_{\text{yuk}}(r), \text{ with} \tag{2.35}$$

$$\mathbf{f}_{\text{yuk}}(r) = -\frac{\partial U_{\text{yuk}}(r)}{\partial r} = \frac{\epsilon\sigma e^{-\kappa(r/\sigma-1)}}{r^2} + \frac{\epsilon\kappa e^{-\kappa(r/\sigma-1)}}{r}, \text{ and} \tag{2.36}$$

$$r_{\text{cut}}(\kappa,\epsilon) = -\frac{1}{\kappa}\left[\log\left(\frac{\epsilon_c}{\epsilon}\right) - \kappa\right]. \tag{2.37}$$

We set e.g. $\sigma = 1$ and $\kappa = 5$ in our simulations, which means that the pressure tail correction ΔP_{tail} depends on ϵ and ρ in our case,

$$\Delta P_{\text{tail}}(\epsilon,\rho) = \frac{2\pi\epsilon_c\rho^2}{75}\left[\log\left(\frac{\epsilon_c}{\epsilon}\right)^2 - 13\log\left(\frac{\epsilon_c}{\epsilon}\right) + 43\right]. \tag{2.38}$$

By using the ΔP_{tail} value of Eq. (2.38) and adding it to the obtained value of P in the simulation, the real pressure for the simulation can be calculated, which we performed for our system in Chap. 6. Having treated the computational feasibility we address now the physical limitations of the model.

Physical limitations

The use of a screened Coulomb interaction and a certain cutoff is a simplification to avoid the calculation of the interactions of all charges in the system with each other. Despite of the available computation power which has increased tremendously in the last years [65] it wouldn't be possible until now to investigate crystallization with explicit charges close to the phase coexistence lines because of the high computational effort at reasonable system sizes. For example, if we would like to simulate $10,000$ charged macromolecules carrying a charge which is only 10 times higher than the charge of the salt ions, we would have to simulate at least the long-ranged Coulomb interactions of $100,000$ counter-ions plus the $10,000$ macromolecules to obtain a neutral system. In addition, we would probably like to have not only the counter-ions but also additional charges in the system. This leads to a system which contains much more than $100,000$ charges, and the simulation of the crystallization will only be possible with more computation power in the next few years.

However, as found in previous work [66, 67, 68, 69] and as we will see in Chap. 5, the screened Coulomb potential is a good model to describe many phenomena of systems consisting of charged macromolecules, not only when compared to simulations but also in comparison to experimental systems, see Chap. 5.

We should keep the following things in mind: The screening only works if many charges, which weaken the long-ranged forces, are between the macromolecules. If there are no charges located in between, Eq. (2.30) can't be applied anymore.

In addition, there can be the case where only a few ions are between the macromolecules, e.g. if they are close to each other. This could lead to non-linear charge effects. In such a case, charges of the surrounding solution would move very fast to such a position to compensate for the electrostatic potential[9]. We will take on this point again in the discussion.

2.4 Crystallization

Crystallization can be seen as a combination of two process: *Nucleation* of growth units, when a crystal cluster forms from the parent phase at the beginning until a critical cluster size has been reached, and further *crystal growth*, when the critical cluster size has been overcome.

2.4.1 Phase transitions

Nucleation is a first-order phase transition which is thermodynamically characterized by an equal chemical potential of the old and the new phase and by discontinuous first order derivatives at phase equilibrium (classification according to Ehrenfest, 1933). In general, this phase transition occurs if a metastable state is transformed into a truly stable state. In this section we follow mainly refs. [70, 71].

Stable thermodynamic equilibrium

A fluid is in a stable thermodynamic equilibrium, if the Gibbs free energy

$$G = F + pV \tag{2.39}$$

is minimal. In this relation, V is the Volume and p a certain pressure. F is the Helmholtz free energy, which is given by

$$F = -p(V)dV \tag{2.40}$$

with the volume dependent pressure $p(V)$.

Note, that at certain conditions, two phases are able to coexist as separate phases if they are in contact with each other, which is called *phase equilibrium*. Then, the chemical potential

$$\mu_{\mathrm{old,e}} = \mu_{\mathrm{new,e}} \equiv \mu_{\mathrm{e}} \tag{2.41}$$

of the two phases is the same.

[9] Note, that in our examples the charges of the neutralizing salt are monovalent, but in the general case multivalent charges must be considered, too.

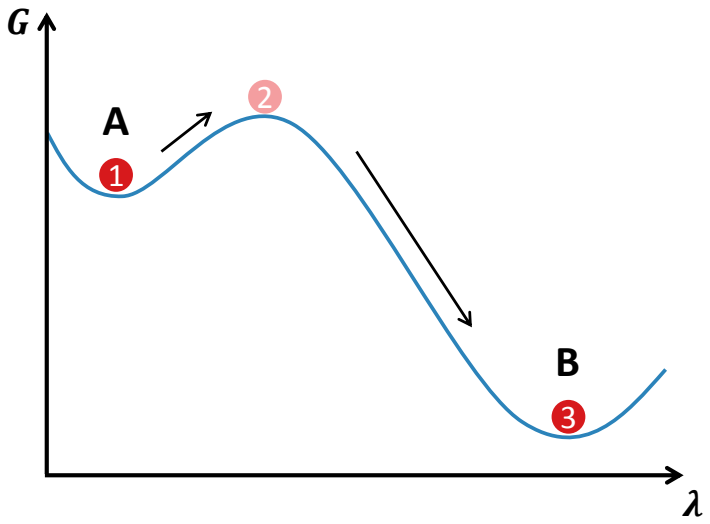

Figure 2.6: Phase transition from a metastable state A (local minimum) to a stable state B (global minimum). The transition is characterized by an order parameter λ (x-axis). Before leaving the metastable state A at position (1), an energy barrier must be overcome at position (2) before the truly stable state B at position (3) can be reached.

Stable and metastable

Above we have mentioned, that if G is minimal, more precisely if G has a global minimum, the state of the system is truly stable. However, if there is a local minimum of G, the system is metastable and is able to perform a transition from metastable to truly stable under certain conditions (fig 2.6). This transition is then called *phase transition of first order*, where the first order derivatives, e.g.

$$V = \left.\frac{\partial G(T,p)}{\partial p}\right|_T \tag{2.42}$$

are not continuous.

The process of the beginning phase transition is called nucleation of the new phase and takes place in the space of the parent phase. A metastable state can be obtained by *supersaturation* or *undercooling* of the system, which we address in the next section.

Driving force for nucleation

From a thermodynamic point of view, the ambition of a system is to occupy a lower energy state [70]. Here, this is the ambition of the first phase in the system. The thermodynamic driving force for such a first order phase transition and hence for the

nucleation process is the *supersaturation* of the system,

$$\Delta\mu \equiv \frac{(G_{\text{old}} - G_{\text{new}})}{N} \equiv \mu_{\text{old}} - \mu_{\text{new}}, \tag{2.43}$$

where N is the number of molecules. The supersaturation $\Delta\mu$ is the gain in Gibbs free energy per molecule when advancing from the (local) minimum G_{old} to the new (e.g. global) minimum G_{new}. The quantities μ_{old} and μ_{new} are also called the *chemical potentials* at these minima.

When there is no driving force, $\Delta\mu = 0$, the phase is called *saturated*. In addition, the old phase is called *undersaturated*, if $\Delta\mu < 0$ with the definition in Eq. (2.43). In both cases, nucleation is not possible. This occurs e.g. when the minimum at G_{old} is lower than the other minimum.

If we regard this the other way round, the nucleated phase would be dissolved again if we advance from a lower minimum to a higher (local) minimum, where the parent phase is metastable again. In our simulations, we are able to perform such operations with the rare event methods described later in this chapter. Before we do this, we introduce a generally used nucleation theory in the next section.

2.4.2 Classical Nucleation Theory

The Classical Nucleation Theory (CNT) [72, 73, 74, 75] is suited for the fundamental theoretical description of the nucleation process. In the last section we have seen that a necessary condition for the first order phase transition and hence the nucleation is the supersaturation of the parent phase. However, this is not a guarantee that this is going to take place within an available observation time, because the system remains a certain time in the metastable state before advancing to the truly stable state. This is originated in the fact that the two states are separated by an energy barrier (see also Fig. 2.6). Beyond that, a real system is able to take different paths for the transition, and each path can have a different underlying energy landscape, e.g. the nucleus can have different shapes during this process, and some of the shapes could be more advantageous for the transition to the stable state.

From an energetic point of view, the preferred path should be the one which requires the lowest energy [70, 76]. An obvious path would be a complete density change from the old phase to the new phase. This would result in a volume change which affects directly the Gibbs free energy G. However, this path is very unlikely because all particles would have to take part in this transition, which has a very high energy barrier of $N\Delta\mu$. Therefore, a path which originates from local density changes is much more likely, e.g. caused by local fluctuations of $n \ll N$ particles (Fig. 2.7). This results in an energy barrier height in the order of $n\Delta\mu \ll N\Delta\mu$, which makes the transition much more probable than in the previous case.

In nature, this can be observed directly: Nano-sized *precursors* grow first due to fluctuations with their density close to the parent phase which are the starting point

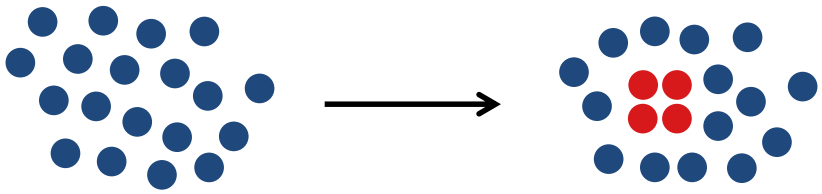

Figure 2.7: Illustration of the cluster formation which is the beginning of the phase transition. The parent phase is shown on the left-hand side. The phase transition begins only in a small domain (red particles on the right-hand side) of the parent phase, induced by local density fluctuations.

for the further phase transition.

Cluster formation

We investigate the cluster formation of n growth units. In general, such a cluster which represents the new phase is separated from its parent phase by a so-called *phase boundary* or *dividing surface* [70]. The free energy ΔG required to form the cluster of size n is given as

$$\Delta G(n) = -n\Delta\mu + G_{\text{ex}}(n), \tag{2.44}$$

with the cluster *excess* energy $G_{\text{ex}}(n)$. $G_{\text{ex}}(n)$ can be determined in a sufficiently large cluster[10] with the help of thermodynamic considerations:

$$G_{\text{ex}}(n) = \phi(V_n) - (p_n - p)V_n + \int_p^{p_n} V_n(P)dP. \tag{2.45}$$

In this equation, ϕ is the total surface energy of the cluster, p and p_n the pressures of the old phase and the n-sized cluster, and V_n the equation of state. For spherical clusters with radius R in the condensed phase, where it is assumed that the cluster is incompressible which leads to the cancellation of the last two terms in Eq. (2.45), this can be simplified to

$$G_{\text{ex}}(R) = 4\pi\gamma R^2, \tag{2.46}$$

where γ is the surface tension. Equation (2.44) can then be expressed for spherical clusters with $n = \frac{4}{3v_0}\pi R^3$ as

$$\Delta G(R) = -\frac{4\pi\Delta\mu}{3v_0}R^3 + 4\pi\gamma R^2, \tag{2.47}$$

where $v_0 = \rho^{-1}$ is the molecular volume[11]. Note, that this expression is only valid for the condensed phase. For the vapor phase, the expression would be more complicated

[10] Then, the cluster can be seen as a distinct phase.
[11] Volume occupied by a single molecule in the cluster.

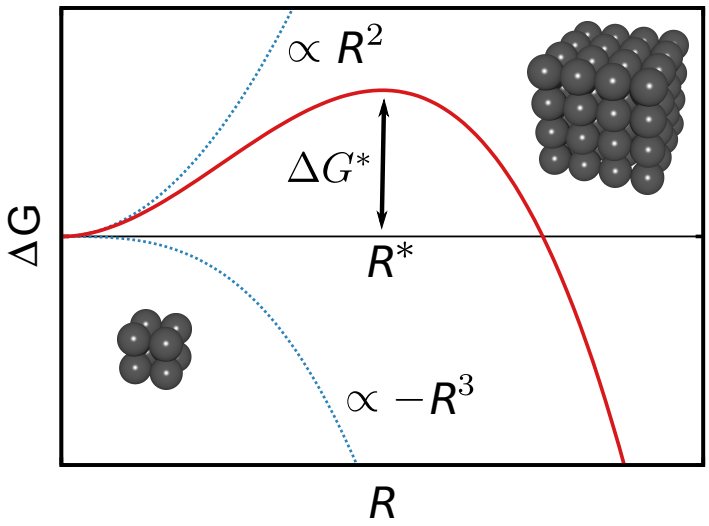

Figure 2.8: Energy landscape $\Delta G(R)$ (solid red line, Eq. (2.47)) of the nucleation as a balance of bulk $\propto R^3$ and surface $\propto R^2$ term (blue dashed lines). This leads to an energy barrier with a barrier height of ΔG^* (Eq. (2.50)) at the *critical nucleus size* R^* (Eq. (2.51)).

because of a different equation of state V_n, where the cluster can not be seen as incompressible any more like in the condensed phase.

Relation (2.47) leads to an energy barrier $\Delta G(R)$ (Fig. 2.8) for the first-order phase transition and hence the nucleation, which is a balance of the bulk $(-\frac{4\pi\Delta\mu}{3v_0}R^3)$ and the surface term $(4\pi\gamma R^2)$ of Eq. (2.47). The energy landscape has a maximum with a height of

$$\Delta G^* \equiv \Delta G(R^*) \tag{2.48}$$

at a certain cluster size R^*, the *critical nucleus size*.

The consequence of this is that clusters with a size $R < R^*$, so-called *subnuclei*, are more likely dissolved than they continue to grow. In contrast, clusters $R > R^*$, so-called *supernuclei*, are capable to grow spontaneously because the free energy $\Delta G(R)$ is then lowered with further growth. For $R = R^*$ the *committor probability*[12] is 0.5, which means that further growth is equally likely with becoming dissolved again.

Therefore, the formation of nuclei is a precondition for the onset of crystal growth and hence the first-order phase transition. The formation of the nucleus is statistically spoken a *random event*, and can even be a *rare event*[13] depending on the height ΔG^* of the energy barrier. The condition to obtain the maximum of the free energy is

$$\left.\frac{d\Delta G}{dn}\right|_{n=n^*} = 0. \tag{2.49}$$

[12]The so-called *committor probability* describes the probability to complete the transition to the new phase from a given phase point, usually characterized by an order parameter λ.

[13]More on rare events in section 2.5

The height G^* can be obtained for the condensed[14] homogeneous nucleation like in our case using equations (2.47), (2.48) and (2.49):

$$\Delta G^* = \frac{16\pi v_0^2 \gamma^3}{3\Delta\mu^2}. \tag{2.50}$$

The critical radius R^* is thereby given as

$$R^* = \frac{2v_0\gamma}{\Delta\mu} \tag{2.51}$$

or in terms of the number of molecules in the critical cluster

$$n^* = \frac{32\pi v_0^2 \gamma^3}{3\Delta\mu^3}. \tag{2.52}$$

We see, that the critical nucleus size and the barrier height decrease with increasing supersaturation $\Delta\mu$. Equations (2.50) and (2.52) yield

$$\Delta G^* = \frac{1}{2} n^* \Delta\mu, \tag{2.53}$$

from which we are able to calculate e.g. the difference of the chemical potential $\Delta\mu = 2\Delta G^*/n^*$ of the new and the old phase if we know the height of the energy barrier and the number of molecules in the critical nucleus.

Cluster size distribution

As mentioned above, the formation of clusters takes place randomly in space and time of the parent phase due to fluctuations. As a consequence, there exist clusters with different sizes at certain locations in the old phase. The *equilibrium cluster size distribution* $C(n)$ is thereby given as [77],

$$C(n) = C_0 \exp\left(-\frac{\Delta G(n)}{k_B T}\right), \tag{2.54}$$

with $C_0 = 1/v_0$, the concentration of cluster locations in the system, without considering cluster-cluster interactions.

Nucleation rate

The *nucleation rate* quantifies the creation of *supernuclei* per volume and time and is therefore measured in $\sigma^{-3}\tau^{-1}$ in our 3D system, where σ is the unit of length and τ the unit of time.

[14] In the vapor phase, the pressure plays a crucial role for the critical quantities.

To quantify the nucleation we assume that clusters grow and shrink via particle attachment [78]:

$$B_{n-1} + B_1 \quad \Longleftrightarrow \quad [k_{+,n-1}][k_{-,n}]B_n \tag{2.55}$$
$$B_n + B_1 \quad \Longleftrightarrow \quad [k_{+,n}][k_{-,n+1}]B_{n+1} \tag{2.56}$$

B_n	Cluster with n particles
$k_{+,n-1}$, $k_{-,n}$	Attachment and detachment rates

Then, the master equation can be formulated for the cluster size distribution $N_n(t)$ [79]:

$$\frac{dN_n(t)}{dt} = N_{n-1}(t)k_{+,n-1} - [N_n(t)k_{-,n} + N_n(t)k_{+,n}] + N_{n+1}(t)k_{-,n+1} \tag{2.57}$$

From this master equation, Becker and Döring have derived the nucleation rate as

$$k = \Gamma e^{-\beta \Delta G^*}. \tag{2.58}$$

For details of the derivation of Eq. (2.58) from Eq. (2.57) please refer to references [70, 75]. In many fields an equation of this form is also called the *Arrhenius* equation. Here, Γ is the kinetic pre-factor and $P_c = e^{-\beta \Delta G^*}$ the probability that a critical nucleus forms. Note, that Γ contains the unit of the rate, $[\Gamma] = \sigma^{-3}\tau^{-1}$ in 3 dimensions. The kinetic pre-factor can be calculated via

$$\Gamma = Z f_+^* C_0, \tag{2.59}$$

where f_+^* is the attachment rate of particles to the critical cluster at temperature T (kinetic part), C_0 contains the spatial characteristics of the cluster, and Z is the *Zeldovich factor*,

$$Z = \left[\frac{-(d^2G/dn^2)|_{n=n^*}}{2\pi k_B T}\right]^{1/2} \overset{\text{spherical}}{\Longrightarrow} \left[\frac{G^*}{3\pi k_B T n^{*2}}\right]^{1/2} = \frac{\Delta\mu^2}{8\pi v_0 (k_B T \gamma^3)^{1/2}} \tag{2.60}$$

for $n^* \geq 10$. Note, that Z is the inverse width of the nucleus region, for further details refer to [70]. Typical values for these quantities are $0.01 \leq Z \leq 1$ for the Zeldovich factor, $1 \leq f_+^* \leq 10^{12}\tau^{-1}$ for the attachment rate, and $10^{15} \leq C_0 \leq 10^{29}\sigma^{-3}$ for volume nucleation as well as $10^{13} \leq \Gamma \leq 10^{41}\sigma^{-3}\tau^{-1}$ in most cases [70]. Smaller values stand for lower attachment rates and the presence of seeds and active centers in the system.

In this work we are going to compare our direct simulations to the Classical Nucleation Theory from this chapter. To this aim, the particles of the simulated system which are in a solid structure and hence in a nucleation cluster must be identified.

2.4.3 Crystallization progress: order parameter

To monitor the progress of crystallization in our 3-dimensional system, a mapping of the system's state to a 1-dimensional *order parameter* is necessary. For convenience and to apply further theoretical methods, this parameter should increase monotonously when the system progresses to its aim, e.g. the particles crystallize successively.

For our work, namely crystallization, the size of the largest cluster of solid particles in the system is a good parametrization of the committor function of the system and is an intuitive quantity which leads to physically plausible pathways [80], e.g. it is possible to track the crystal growth in the system, which wouldn't be possible if we would use a global variable like the system's total energy.

Detecting solid particles

To identify a particle i as *solid-like* or *liquid-like* in the 3D simulation box we have to analyze the spatial orientation of the particles relative to their neighbors. Therefore, we use the averaged local bond $\overline{q}_l$ order parameters which allow not only to determine a solid structure but also the underlying lattice type [81, 82]:

$$\overline{q}_l(i) = \sqrt{\frac{4\pi}{2l+1} \sum_{m=-l}^{l} \left|\overline{q}_{lm}(i)\right|^2} \tag{2.61}$$

with the average $\overline{q}_{lm}(i)$ over all neighbors $\widetilde{N}_b(i)$ and the particle i itself (second neighbor shell),

$$\overline{q}_{lm}(i) = \frac{1}{|\widetilde{N}_b(i)|} \sum_{k \in \widetilde{N}_b(i)} q_{lm}(k). \tag{2.62}$$

The complex vector q_{lm} (first neighbor shell) is given by

$$q_{lm}(i) = \frac{1}{|N_b(i)|} \sum_{j \in N_b(i)} Y_{lm}(\vec{r}_{ij}), \tag{2.63}$$

with the sum over all neighbors $N_b(i)$ of particle i constructed by the spherical harmonics Y_{lm}[15] which are dependent of the spatial orientation of the particles i and j, connected by the vector $\vec{r}_{ij}$. The maximum neighbor distance to determine the neighbors of particle i is determined via the first minimum of the radial distribution function of the system. Depending on the value of $\overline{q}_l(i)$ for a certain order l and a particle i, the 3D structure can be analyzed and quantified, e.g. it can be detected, if particle i is member of a crystal lattice and hence solid-like.

[15] The spherical harmonics Y_{lm} are a complete set of orthonormal eigenfunctions of the angular part of the Laplace operator. They are also used e.g. for atomic orbitals calculations.

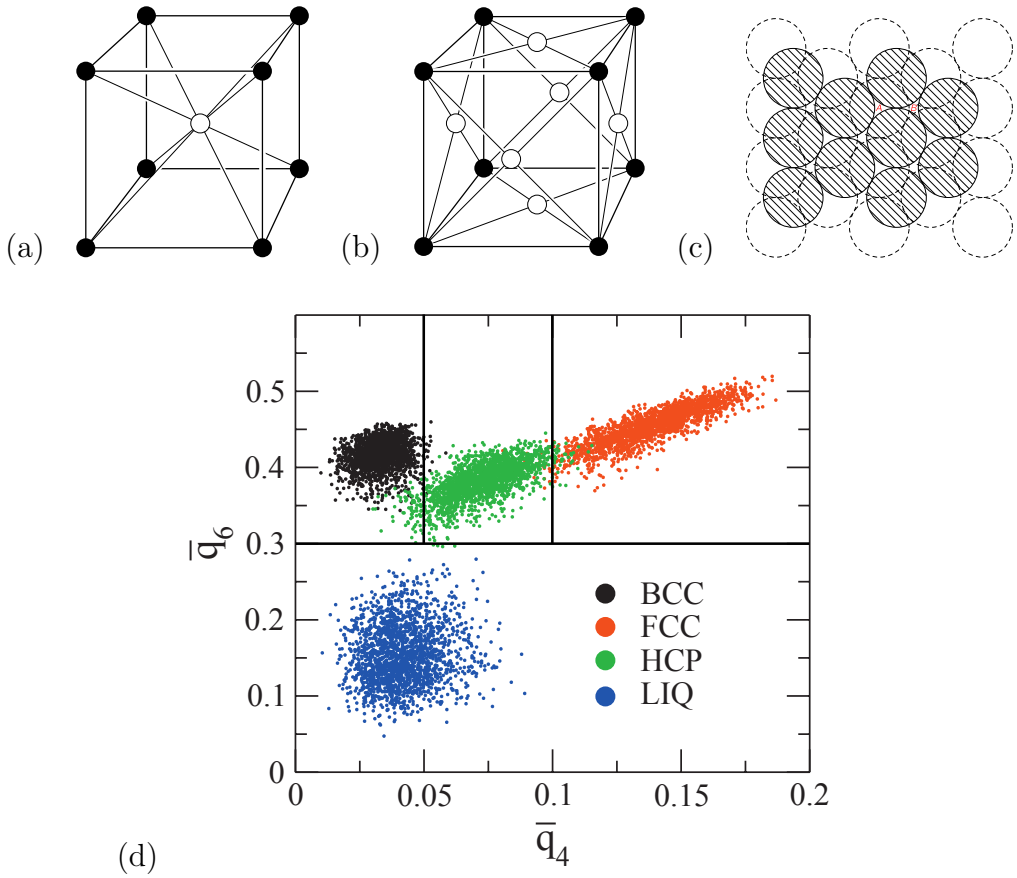

Figure 2.9: Different crystal lattice structures: (a) body-centered, (b) face-centered, (c) depending on the placement of the third layer hexagonal close-packed (hcp) or fcc otherwise. Images (a)-(c) taken from [83]. (d) Scatter plot of the $\overline{q}_4$ and $\overline{q}_6$ values for the different 3D crystal structures. According to this plot the particles can be classified as solid-like and liquid-like using only the $\overline{q}_6$ order parameter (horizontal line). In addition, if they are solid-like, the structure can be determined using the $\overline{q}_4$ parameter (vertical lines). The scatter data is taken from [82].

Solid particles and crystal lattice of the Yukawa system

In our Yukawa system according to the phase diagram (see also Fig. 2.4) the following crystal structures are expected to be possible: face-centered cubic (fcc), body-centered cubic (bcc), and hexagonal close-packed (hcp). Figure 2.9(a)-(c) gives an overview of these structures. To detect the structures in the simulations, we use the $\overline{q}_l$ order parameters with $l = 4$ and $l = 6$. Fig. 2.9(d) shows the $\overline{q}_4\overline{q}_6$ scattering plane of these parameters for the expected crystal structures.

We see, that to identify only solid particles in our system, the $\overline{q}_6$ order parameter

is sufficient, and we can draw a horizontal separating line in Fig. 2.9(d) at $\overline{q}_6 \approx 0.3$. In the post-processing of the simulation, we use in addition the $\overline{q}_4$ order parameter to distinguish between different crystal lattice structures, see the vertical lines in Fig. 2.9(d) at $\overline{q}_4 \approx 0.05$ and $\overline{q}_4 \approx 0.1$.

Note, that the scattering clouds in Fig. 2.9(d) are caused by fluctuations in the system. For the ideal structure, there would only be a single point in the scattering plot. Depending on the magnitude of the fluctuations, there can be overlappings in the scattering clouds, which are in our case already minimized by using the averaged $\overline{q}_l$ order parameters. However, in the transition regions between the domains, the affiliation of a particle to a certain structure is not clearly defined.

Order parameter

In the last section we have identified solid particles. As a next step, we use a cluster algorithm in combination with a neighbor analysis to determine the clusters of solid particles in the 3D simulation box. The size n of the largest cluster in the system is then used as order parameter and therefore to monitor the progress of crystal growth.

Note, that the cluster size is dependent on the choice of $\overline{q}_6 \approx 0.3$ for detecting solid particles. To only characterize the progress of the crystallization, this threshold plays a minor role. However, if we would like to compare to nucleation theory we have to keep in mind that this may not be the real cluster size.

2.5 Rare events

In this work, we are interested in the nucleation processes close to the coexistence line where the chemical potential difference $\Delta\mu$ (Sec. 2.4) is small and where the attachment rate of the growth units is low. At these conditions, the crystallization of the system can be a result of only one crystalline supernucleus [70], and the *mononuclear* crystallization mechanisms can directly be investigated.

However, a small supersaturation $\Delta\mu$ implies a high energy barrier ΔG^* towards crystallization, which means that the growth of a cluster until its critical size is a *rare event*.

In general, a rare event is defined as an event which is short compared to its waiting time (Fig. 2.10). In conventional *brute-force* computer simulations or in experiments we therefore mainly observe the uninteresting waiting time of the system, without observing the fluctuation-driven event itself. If the event is observed by chance, we wouldn't be able to perform a statistical analysis. For the experiments this would mean to wait several years for the event, and in simulations we would calculate useless trajectories where the main effort is spent in calculating the interactions in the system without sampling the event.

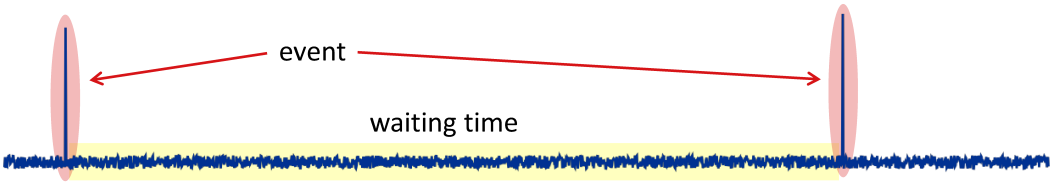

Figure 2.10: Rare event illustration: If the waiting time is much longer than the event itself, the event is difficult to access in experiments and simulations.

2.5.1 Simulating rare events

Recently, a lot of rare event sampling methods have been invented to reduce the amount of unnecessary trajectories[16], e.g. umbrella sampling [18, 19], Bennett - Chandler methods [20, 21, 19], transition path sampling [22, 23, 24], transition interface sampling [27, 28, 29], milestoning [30, 31, 32], nudged elastic band [33, 34], string methods [35, 36], weighted-ensemble methods [37], non-equilibrium umbrella sampling and 'splitting'-type methods [39, 40, 41, 42, 43, 44, 45, 46, 47, 48], where the overall calculation can be split up in many path fragments.

Thereby, the applicability of a method depends on the physics of the system and the research interest, because the methods are only applicable for certain ensembles and aim for extracting different observables from the simulations. For this work, we use the Forward Flux Sampling method [39, 40, 42].

2.5.2 Forward Flux Sampling

Forward Flux Sampling (FFS) is suited to simulate quasi-static systems with stochastic dynamics in equilibrium or non-equilibrium without knowing the phase space density [42] and is therefore applicable to the crystallization of charged macromolecules, where we advance from a metastable state A to a final (stable) state B in phase space and do not have information about the intermediate states in phase space. With FFS simulations the transition rate from A to B can be directly obtained as well as the successful pathways of the system.

To perform such a task, FFS uses an order parameter λ to measure the progress towards the final state. This order parameter is expected to grow monotonously in positive B direction. The way from A to B is then partitioned by a set of non-intersecting interfaces defined at specific values λ_i, where i is the interface index and $\lambda_0 = \lambda_A$ is the border of state A, and $\lambda_n = \lambda_B$ is the border of state B, respectively. The system is in state A, if $\lambda < \lambda_A$ and in state B, if $\lambda \geq \lambda_B$.

[16] 'Unnecessary' in this sense means, that we gain no new information from the trajectories of the system when it is only in the initial state, because we are interested in the transition dynamics to the final state and the corresponding pathways.

The transition rate in this scheme is given as

$$k_{AB} = \Phi P_B, \tag{2.64}$$

where Φ is the so-called escape flux, the flux of trajectories leaving A. P_B is the probability that a trajectory which has successfully left the initial state A manages it to make it to the final state B without returning to A, crossing all successive interfaces. Thereby, P_B is split by the interfaces,

$$P_B = \prod_{i=0}^{n-1} p_i. \tag{2.65}$$

With the help of the above interfaces the system can be driven from state A to state B using a certain algorithm, which is also suited to calculate Φ and P_B. Here, we use the direct FFS algorithm, another possibility would be e.g. the branched growth algorithm [42, 39, 48].

The direct FFS algorithm (DFFS)

The direct FFS algorithm is used to populate the scheme with the set of interfaces λ_i described above in an efficient way, like illustrated in Fig. 2.11. The first step is to launch a conventional MD simulation run with a random generated configuration point from the initial state A with $\lambda < \lambda_A$. Every time, the simulation run crosses λ_A in positive B direction, a configuration point is stored at λ_A (see also the numbered dots in Fig. 2.11). The escape flux Φ is then given by

$$\Phi = \frac{N_0}{t}, \tag{2.66}$$

where N_0 is the number of collected points at λ_0 during the simulation time t of the whole initial MD run. If the system enters the final state B during this run, the trajectory is continued at a random, equilibrated point of A.

In a second step, a random configuration point is chosen from the set of previously collected points which serves as starting point for a new trial trajectory. This trajectory can either fall back to A (failure), or reach the next interface λ_{i+1} (success). The conditional probability p_i from λ_i to λ_{i+1} is then given as

$$p_i = p(\lambda_{i+1}|\lambda_i) = \frac{M_i}{M}, \tag{2.67}$$

where M_i is the number of successful runs and M the number of total trial runs launched. Note, that p_i is the probability to reach interface λ_{i+1}, hence p_0 is the probability to reach λ_1. This second step is now repeated for each interface i in an iterative way until the final state B is reached and P_B can be calculated via Eq. (2.65).

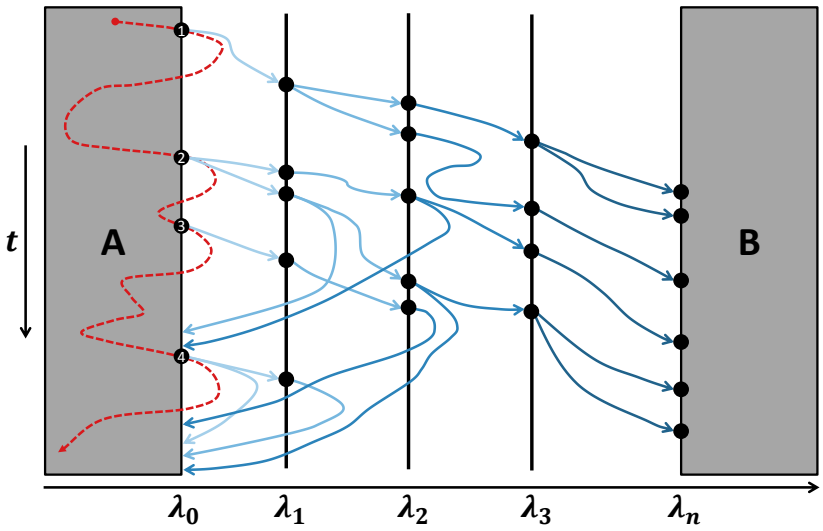

Figure 2.11: Illustration of the DFFS algorithm, schematic view. An initial MD simulation run is started at a random configuration from state A (red dashed line). When λ_0 is crossed in positive B direction a configuration point of the system is stored (serially numbered dots). In a second step, trial runs (blue solid lines) are started at the collected set of points which either reach the next interface λ_{i+1} or fall back to A. This procedure populates the whole sampling scheme from A to B.

In the post-processing, successful transition trajectories can be obtained from the collection of trajectories by backtracking the runs starting at λ_B.

We remark, that FFS is not a panacea for every rare event problem. E.g. it has been demonstrated, that FFS can fail concerning the calculation of the rate constant if there is a slower component of the reaction coordinate in one dimension than in the other [84, 85], or if the folding pathways in protein simulations are not fully sampled, because the system does not explore the complete phase space in such problems [86]. Therefore, the sampling of the phase space should be ergodic at λ_i. In contrast to the previous examples, FFS has been applied successfully for a lot of problems, including crystallization problems, and has also been compared to brute-force simulations [42, 39, 87, 88, 89, 90].

Efficiency of an FFS simulation

The efficiency of an FFS simulation is a balance of computational cost and statistical error [42, 48, 41, 45, 44]. A high computational cost $\mathcal{C}$ and a high statistical error $\mathcal{V}$ lead to a low efficiency

$$\mathcal{E} = \frac{1}{\mathcal{C}\mathcal{V}}. \tag{2.68}$$

Estimates for $\mathcal{C}$ and $\mathcal{V}$ can be obtained analytically by modeling the FFS method as Bernoulli experiment, where trials have the values 'success' and 'failure'[17] [41]. The computational cost $\mathcal{C}$ can be approximated via [41]

$$\mathcal{C} = N_0 R + \sum_{i=0}^{n-1} M_i C_i \tag{2.69}$$

with the cost for a trial run

$$C_i = S[p_i(\lambda_{i+1} - \lambda_i) - q_i(\lambda_i - \lambda_A)], \tag{2.70}$$

the number of interfaces n, $q_i = 1 - p_i$, the cost of generating a configuration R and the fitting constant S. Note, that the computational cost is measured in simulation steps. For the statistical error one obtains [41]

$$\mathcal{V} \approx \sum_{i=0}^{n-1} \frac{(1 - p_i)}{M_i p_i}. \tag{2.71}$$

It has been shown that Eq. (2.71) can be minimized for fixed n, $\{M_i\}$ and P_B by choosing interface positions λ_i such that $M_i p_i$ is equal for all interfaces [91, 48], which results in a constant net flux across the interfaces. Here, in the hypothetical model we keep it at a fixed number. Note, that P_B is determined by the underlying physics of a real system, e.g. by the height of an energy barrier. A measurement function f_i for a constant net flux is [91, 92]

$$f_i = \frac{\sum_{j=0}^{i-1} \log p_j}{\sum_{j=0}^{n-1} \log p_j} \overset{p_j = p}{\Longrightarrow} = \frac{i}{n} \tag{2.72}$$

which is linear at a constant net flux $p_j = p$ when plotted against the interface index i. Using this function f_i the interface positions can be corrected after the analysis of a complete FFS simulation run. In this work we don't use this function to adapt the interface positions after our simulation, but we will use it later as a measurement of the 'quality' of an interface set, which we determine on-the-fly.

2.5.3 Stationary distributions and energy landscapes

With the previously discussed FFS sampling scheme, the stationary distribution and the energy landscape against the order parameter can be calculated by monitoring the whole distribution during a forward (state A to state B) and a backward (B to A) FFS simulation run [43] and weighting this distribution by the corresponding interface transition probabilities p_i. In general, this method is possible for systems

[17] We assume, that the trials are not correlated.

where FFS can be applied. Thus, equilibrium and non-equilibrium systems can be tackled. However, in practice the simulation must be realized for the forward and the backward reaction, which can sometimes be difficult, e.g. a crystal must be dissolved again in the backward run. In contrast, for problems like a 1D particle in a symmetric potential it is sufficient to use only one direction because of the symmetry of the problem.

Stationary distributions

In this section, we follow Ref. [43]. In a brute-force simulation, the stationary distribution $\rho(q)$ can easily be obtained in the states A and B by just letting the simulation run and by monitoring the order parameter q during this run.

However, in the 'barrier' region the sampling is poor because the system stays there from very shortly to never[18]. With FFS, the system is guided through this barrier region and statistics can be obtained for this part in a forward and backward simulation. Therefore, we partition $\rho(q)$ in two contributions for the forward (A) and backward (B) FFS run,

$$\rho(q) = \Psi_A(q) + \Psi_B(q). \tag{2.73}$$

This means, that we must now calculate the two contributions $\Psi_A(q)$ and $\Psi_B(q)$:

$$\Psi_A(q) = p_A \Phi_A \tau_+(q; \lambda_0), \tag{2.74}$$
$$\Psi_B(q) = p_B \Phi_B \tau_-(q; \lambda_n), \tag{2.75}$$

with the probability p_A (p_B) that the system is in A (B) and the escape flux Φ_A (Φ_B) (see also Sec. 2.5.2). The factor $\tau_+(q; \lambda_0)$ is determined by the time which the trajectories remain at a certain q,

$$\tau_+(q; \lambda_0) = \pi_+(q; \lambda_0) + \sum_{i=1}^{n-1} \pi_+(q; \lambda_i) \prod_{j=0}^{i-1} P(\lambda_{j+1}|\lambda_j), \tag{2.76}$$

where the transition probabilities $P(\lambda_{j+1}|\lambda_j)$ for each interface λ_i are directly obtained from the FFS simulation like described in Sec. 2.5.2, which reweight the distribution $\pi(q; \lambda_i)$ in this case. $\pi_+(q; \lambda_i)$ is obtained from the simulations via $\pi_+(q; \lambda_i) = N_q/(\Delta q M_i)$ with the number of trials M_i, the order parameter distribution binning width Δq and a counter N_q for a value of q between q and $q + \Delta q$.

The factor $\tau_-(q; \lambda_n)$ of the backward reaction is calculated as follows:

$$\tau_-(q; \lambda_n) = \pi_-(q; \lambda_n) + \sum_{i=n-1}^{1} \pi_-(q; \lambda_i) \prod_{j=n}^{i+1} P(\lambda_{j-1}|\lambda_j), \tag{2.77}$$

[18] Here, 'never' means that the system is not observed at a certain barrier region position during feasible brute-force simulation time.

where all quantities are now calculated from the backward simulation. For further details, refer to [43]. What is left in our calculations are the quantities p_A and p_B which have the relation

$$p_A k_{AB} = p_B k_{BA}, \tag{2.78}$$

in the steady-state with the transition rates k_{AB} and k_{BA}. Furthermore, in a system with two states A and B and with a low visited barrier region in between, $p_A + p_B \approx 1$, and hence

$$p_A = \frac{k_{BA}/k_{AB}}{1 + k_{BA}/k_{AB}}, \tag{2.79}$$

$$p_B = \frac{1}{1 + k_{BA}/k_{AB}}. \tag{2.80}$$

The free energy profile ΔG can then be approximated for equilibrium systems via

$$\Delta G \sim -k_B T \log\left[\rho(q)\right], \tag{2.81}$$

where $\rho(q)$ is calculated with the forward and backward FFS simulations according to Eq. (2.73).

3 Parallel and optimized rare event sampling

This chapter is the first part of the results, where we extend the sampling methods and describe our optimizations of the Forward Flux Sampling method. Without these extensions it wouldn't have been possible to address the crystallization of charged macromolecules under the conditions which are interesting for us.

The first part of this chapter is about the results of the implementation of FFS in the context of the *Science Experimental Grid Laboratory* (SEGL)[1] for the usage of FFS on high performance computing hardware, which is based on the computational physics part of our collaborative article [93].

The second part is about the general parallelization of the Forward Flux Sampling algorithm, because this algorithm is only described in its serial representation in the FFS literature. We used some tricks in the first part of this chapter to use FFS on high performance computing hardware. Rare event simulations are computationally demanding in the usual case, therefore the serial implementation does not lead to a result in available computing time like in our case.

The last part of this chapter, which addresses the optimized interface placement to increase the efficiency tremendously in FFS simulations, is mainly based on our article [94] with additional information which has become available during application of the methods since the article was published.

- Yevgen Dorozhko, Kai Kratzer, Yuriy Yudin, Axel Arnold, Colin W. Glass and Michael Resch. — Rare Event Sampling using the Science Experimental Grid Laboratory. Civil-Comp. CC2013/2013/00402 (2013).

- Kai Kratzer, Axel Arnold, Rosalind J. Allen — Automatic, optimized interface placement in forward flux sampling simulations. J. Chem. Phys. 138, 164112 (2013),

[1] SEGL is developed at the HLRS Stuttgart, `http://segl.hlrs.de`

3.1 Using FFS on high performance computing hardware

In this chapter we use the Science Experimental Grid Laboratory (SEGL) [95, 96, 97] for running FFS on high performance computing hardware. SEGL is a tool which manages the workflow and dataflow of a simulation and is therefore suited for FFS simulations. In our simulations, the sampling algorithm is described as a workflow, and the storage and selection of configuration points is handled by the dataflow tools. For our intention, an important requirement is the selection of a configuration point at random from the dataspace of the last FFS interface, which has been extra implemented for our purpose by the SEGL group. The great advantage of such a tool is, that the user must not be concerned about the technical details of the implementation of the workflow and dataflow at a certain high performance computing facility, this is performed by SEGL, and the user's experiment can be implemented on a higher level, comparable to e.g. LabVIEW[2].

As the parallelization of the escape flux Φ of the FFS simulation was unclear at this stage (details will be discussed in Sec. 3.2), we generated the configurations on λ_A by one serial MD run which collected the desired number of configurations on λ_A.

The computation of the transition probabilities p_i was performed in parallel by starting a few hundred trial runs at once, e.g. if 1000 trials should be made, we fire the trials in bunches of 5×200 trials, which means that 200 jobs are queued at once. Note, that the duration of the runs is not known in an FFS simulation and this can be a bottleneck in parallelization, but this couldn't be handled in this context so far and will also be discussed later.

The main workflow of the FFS method can be seen in Fig. 3.1. Further details of the implementation of FFS into SEGL with the corresponding dataflow diagrams are given in Ref. [93].

After the simulation run has completed, the simulation data which was produced during the run can be post-processed. The total transition rate $k_{AB} = \Phi \prod_i p_i$ can then be calculated by extracting the escape flux Φ_{esc} from the first calculation stage. Therefore, the number of configuration points on λ_0 in the dataspace is divided by the simulation time which was accumulated in another dataspace of the escape flux calculation block. The probabilities p_i of the second stage are extracted by counting the number of successful configuration snapshots in an interface dataspace and by dividing by the number of total runs launched for this interface. This procedure is repeated for each interface λ_i, respectively, leading to the complete set of p_i.

To extract the successful reaction paths from the datasets we stored the tracking information in the header of each datafile, in this case the name of the originating snapshot. This is sufficient to build the tree of successful pathways.

[2]LabVIEW is a tool of National Instruments, which is widely used in the field of experimental physics to steer experiments (`http://www.ni.com/labview`)

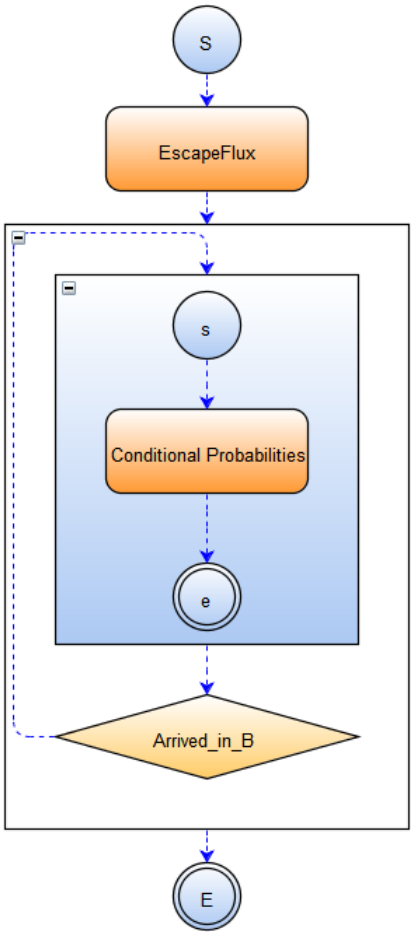

Figure 3.1: Control flow of the FFS method in SEGL: Circles denote start (s,S) and end (e,E) points, where lower-case stands for a "SubExperiment" block. The orange rectangles depict "Solver" blocks, where the main calculation takes place. The "Arrived in B" block is a decision block for proceeding or ending the simulation.

Now, we describe the physical problem which we simulated and present the simulation results. This simulation problem will also be used for further test cases during this work.

3.1.1 Single particle barrier crossing

We simulate a single particle undergoing Langevin dynamics in a 1D potential according to the simulation description of Sec. 2.2.2.

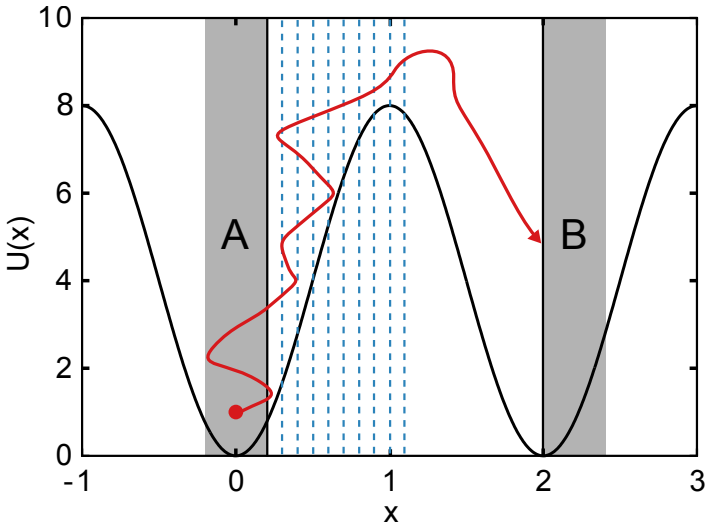

Figure 3.2: Single particle in a 1D potential (Eq. (3.1)) undergoing Langevin dynamics (schematic in this image). For a high barrier, crossings from state A to state B are rare events. We use FFS to push the particle over the barrier using a series of interfaces (blue dashed lines) and calculate transition paths and the transition rate k_{AB}.

Simulation details and order parameter

The potential with height h for this simulation is given by

$$U(x) = \frac{h}{2}\left[1 - \cos\left(\pi x\right)\right]. \tag{3.1}$$

Thereby, the order parameter λ is simply the coordinate x, and a system snapshot is given by the particle's current position x and the current momentum $p = mv$ with the current velocity v and mass $m = 1$. Furthermore, we set the timestep $\tau = 0.001$, the Boltzmann constant $k_B = 1$, the temperature $T = 1$, and the friction $\xi = 1$.

FFS setup

For the setup of FFS, we define the initial state A to be the minimum around $x = 0$ with $\lambda_A \leq 0.2$ and the state B to be the minimum around $x = 2$ (Fig. 3.2), where the final state is reached if $\lambda_B \geq 2.0$. Note, that the location of λ_B is not very sensitive in this case for the transition rate k_{AB}, because if the barrier has been overcome, the probability to reach the minimum around $x = 2$ is $p = 1.0$ if the barrier from A to B is high compared to the energy of the particle. The barrier height can be tuned via the parameter h, which we set to $h = 10k_BT$ in this example. This means that the transition from A to B is a rare event because the barrier is about ten times higher than the fluctuation energy of the particle. We use the FFS method to push

Index i	0	1	2	3	4	5	6	7	8	9	10
λ_i	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	2.0

Table 3.1: Interface positions for the 1D particle in a periodic potential. The interfaces are mainly located at the steep ascent of the barrier with $x \leq 1$.

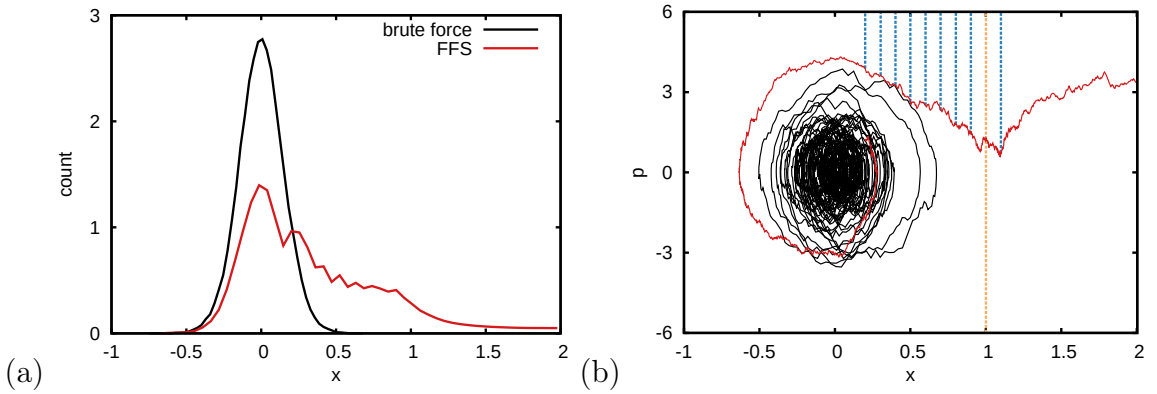

Figure 3.3: (a) Plain histogram of the x-coordinate distribution (not weighted) for the FFS run including the escape flux, and a brute-force run with the same number of simulation steps. The brute-force run is distributed in state A and the FFS run has a distribution also in the barrier region towards B. (b) Phase space plot of a part of the brute-force run (black) and one successful FFS run (red) which manages to cross the interfaces (dashed blue) and reaches B after crossing the barrier maximum (dashed orange).

the particle over the barrier with the help of a set of interfaces, in which the interfaces are located mainly at the steep ascent of the barrier. An overview of the interface locations is given in table 3.1.

3.1.2 Simulation results

Fig. 3.3(a) shows the histogram of the reaction coordinate distribution for the complete FFS simulation and a conventional brute-force simulation with the same number of simulation steps like the overall FFS run. The histogram of the brute-force run is only distributed around state A, where the particle was initially set up, and was not able to leave this minimum within the simulated number of steps. For the FFS run, the histogram is also distributed along the steep part of the energy barrier towards the final state B.

Fig. 3.3(b) is a visualization of the brute-force run and a successful FFS trajectory in phase space (x, p). Note that for better visibility only a part of the brute-force run is shown. As expected, the brute-force run stays in region A with its momentum fluctuating around zero because of the direction changes of the particle at the potential

i	0	1	2	3	4	5	6	7	8	9
p_i	0.324	0.245	0.253	0.199	0.227	0.267	0.385	0.603	0.864	0.917

Table 3.2: Conditional probabilities p_i for each interface transition. By multiplying these values, the overall transition probability from λ_A to λ_B is obtained, $P_B = \prod_{i=0}^{9} p_i$. Note, that an index of i addresses the transition $(\lambda_{i+1}|\lambda_i)$.

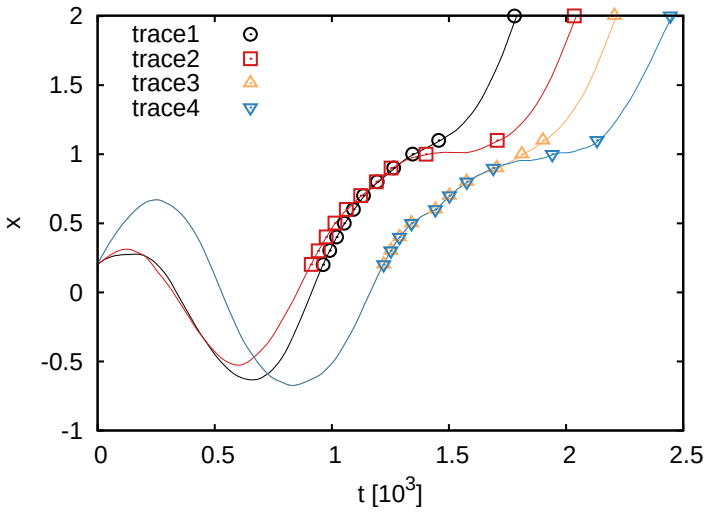

Figure 3.4: Selection of successful trajectories of the single particle barrier crossing simulation. The runs are backtracked until the second last point before finally escaping from A. The dots represent the configuration points on the particular interfaces. Note, that trace 3 and 4 have the same escape trace and split up later.

boundaries. The FFS trajectory is able to leave this basin by crossing the interfaces until the barrier has been overcome.

From the FFS simulation we obtain the escape flux of trajectories leaving state A: $\Phi = 0.424\tau^{-1}$. The results of the particular interface transition probabilities p_i are given in table 3.2, which lead to $P_B = \prod_{i=0}^{9} p_i = 4.455 \times 10^{-5}$ The transition rate $k_{AB} = \Phi P_B$ is then

$$k_{AB} = 1.89 \times 10^{-5}\tau^{-1}$$

for this simulation, which means that the average waiting time for a single crossing event at barrier height $h = 10k_BT$ is $5.30 \times 10^4\tau$ or 5.3×10^7 timesteps.

From the FFS simulation we are also able to extract the successful transition pathways by backtracking the runs which reached B. We show this exemplarily in Fig. 3.4. These successful runs can be used to study the physical transition dynamics, e.g. in this case we see, that the runs gain momentum before they climb up the barrier by

coming from highly negative x values at the left-hand side of basin A.

3.1.3 Discussion

The results of the last section have been obtained using FFS in the context of SEGL on high performance computing hardware. Thereby, SEGL handled the complete workflow and dataflow of the simulation. The advantage of this scheme is, that if all the workflows are defined, the physical simulation can easily be exchanged with other simulation codes without implementing FFS to the particular simulation code itself, allowing to simulate different rare event problems with a low implementation effort.

In the simulation, trajectory fragments of the P_B calculation could be calculated simultaneously by queuing and executing several jobs at once, which was also managed by SEGL. However, as already mentioned above the duration of the runs in an FFS simulation is a priori unknown. Therefore, an interface transition must be calculated completely until the next configuration of the new interface can be drawn. This leads to bottlenecks in the calculation in this *synchronous* parallelization scheme, where e.g. only one job is still running but must complete to be able to advance to the next calculation stage.

The presented scheme in this section is faster than the serial implementation anyways, but there are possibilities for further optimizations. We will now discuss the parallelization of FFS which leads then to a more complex workflow, as we will see.

3.2 Parallelization of FFS

To extend the serial implementation described in the FFS literature [42, 40] and to optimize the synchronous parallel implementation of the P_B calculation (Sec. 3.1), we are going to perform some considerations and tests in this section. FFS is split up in the calculation of the escape flux Φ and in the computation of the transition probabilities p_i and hence P_B to calculate the transition rate $k_{AB} = \Phi P_B$ (see also Sec. 2.5.2). We start with the parallelization of the escape flux Φ.

3.2.1 Escape flux

In a serial FFS simulation, the first step is to start an initial MD simulation run in A to calculate the escape flux Φ. Every time the system crosses the border of λ_A in positive direction of B a configuration is stored. However, in a parallel simulation the parallelization method of this task is unclear at first glance. In this section, we use the 1D particle system of Sec. 3.1.1 to verify our considerations.

A first possibility would be to draw many configurations from state A in parallel and simulate until λ_A is reached. An extension to this would be to collect not only one but $N_{0,p}$ points on λ_A in each parallel run. The question is now, if this leads to

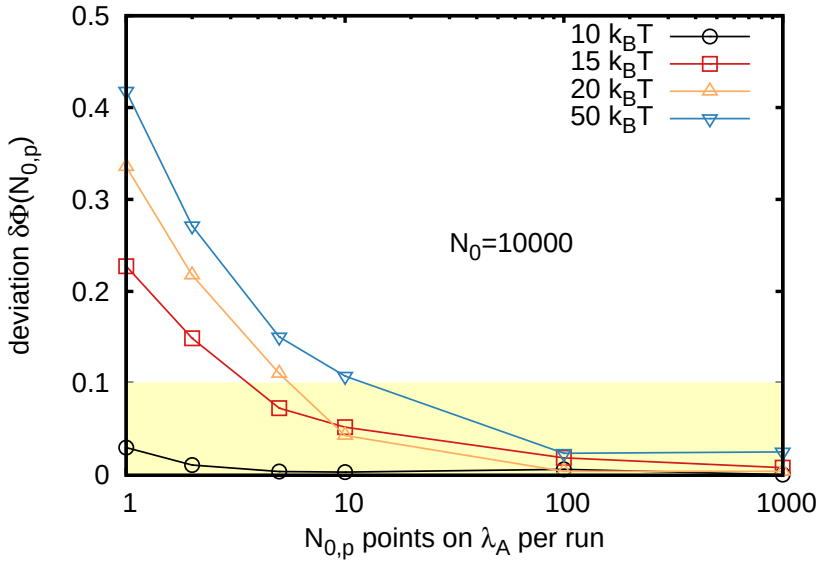

Figure 3.5: Deviation $\delta\Phi$ for the parallelization of the escape flux calculation by ending the parallel runs if $N_{0,p}$ points are calculated on λ_A for each trajectory. The number of parallel runs is $N_0/N_{0,p}$. The shaded area is the acceptable error domain, determined by the mean error in the serial escape flux. Note, that this is a systematic deviation because the fall-back time is not counted if the runs are ended after having collected a configuration point.

the correct result of the escape flux Φ, because we omit a part of the fall-back time in both cases, which should be more severe in the former case, where only one point is collected. Therefore, we performed a standard serial escape flux run and collect N_0 points to calculate a reference flux Φ_{ref}. Then we distribute the collection of N_0 points to $N_0/N_{0,p}$ parallel runs. The deviation $\delta\Phi$ of the parallel flux $\Phi_{N_{0,p}}$ is then

$$\delta\Phi(N_{0,p}) = \frac{\left|\Phi_{N_{0,p}} - \Phi_{\text{ref}}\right|}{\Phi_{\text{ref}}}. \tag{3.2}$$

Fig. 3.5 shows this deviation for a total number of points $N_0 = 10000$. Note, that this graph is simulation specific, but as we use a generic simulation problem this can be transferred to other simulation problems, too. We see, that if we collect only a small number of points on λ_A for each run, the escape flux deviates from the serial one. This is simply due to the fact, that we do not account for the fall-back time of the escape run, which is cropped if we end our runs after collecting a point on λ_A. This error becomes smaller if we collect more points on λ_A. However, a small systematic error remains, which vanishes in the natural fluctuations of the escape run for many points. However, collecting many points means that depending on the number of desired points the count of parallel runs can't be that high any more. We also see that the higher the energy barrier in the system, the higher is the error in the escape

flux calculation on λ_A if the same number of points is collected like at smaller energy barrier heights. This implies, that at high energy barriers, a lot of points must be collected to decrease the error in the escape flux Φ.

This leads to another approach to calculate the escape flux Φ in parallel, namely by fixing the simulation length t_s of the serial escape trajectory. In the next step, we divide the desired length t_s by the amount of parallel trajectories so that every client calculates a total simulation time of t_p. If $t_s = 10^6$, one could e.g. simulate $100 \times t_p$ with $t_p = 10^4$ and use the collected points on λ_A during this time. We suggest to choose a length of t_p such, that in most cases at least 1 point on λ_A is visited during this time[3]. The calculation time t_p of these runs is added to the escape flux in any case, even if no point is collected during such a run. With this procedure, we account for the fall-back time as well which was not considered above. For this method, the number of points on λ_A is not exactly pre-determined in advance, since we have to finish every run. But if we collect $\mathcal{O}(1000)$ points on λ_A a deviation of a few points doesn't matter for the accuracy of the escape flux. We measure the deviation in this case with $\delta\Phi(t_p) = (|\Phi_{t_p} - \Phi_{\text{ref}}|)/\Phi_{\text{ref}}$, where the reference escape flux has the same calculation time as the added partial escape trajectories. Fig. 3.6 shows the result of this type of simulation. The simulations have been performed for different energy barrier heights but with the same order of configuration points on λ_A, $\mathcal{O}(1000)$. Therefore, t_s was adjusted for the different barrier heights to match the number $N_0 \approx 1000$. We see, that we should at least collect 1 configuration point on average for each escape trace within its length t_p to obtain a reasonable estimate for Φ. This means, that if we would like to calculate $N_0 \approx 1000$ points on λ_0, it should be safe to spawn e.g. 100 clients with $t_p = t_s/100$.

3.2.2 Transition probabilities

For the transition probabilities p_i in a serial simulation we would draw a configuration point from the last calculated interface λ_{i-1} and simulate until we reach λ_i or fall back to λ_A, then draw another point and so on, one by one. As long as we calculate the same interface transition, this procedure is independent if carried out in parallel, because we have to draw the points anyways, and it doesn't matter if we perform that in a series or simultaneously. Therefore, we are able to choose as many points as we have parallel clients available and fire trial trajectories starting at the drawn points from λ_{i-1} in parallel, while still trial trajectories are required.

However, there are drawbacks in this scheme. First, the duration of the runs is unknown beforehand, because they are of stochastic nature and only ended if they reach λ_A or λ_{i+1}. Fig. 3.7 shows a distribution of run lengths for different interfaces λ_i. We see, that the run length per interface λ_i can be very different and that there can be a general trend for the run lengths of different interfaces, e.g. the main part

[3]This can be estimated by a short escape trajectory which calculates a few (e.g. $\mathcal{O}(10)$) points.

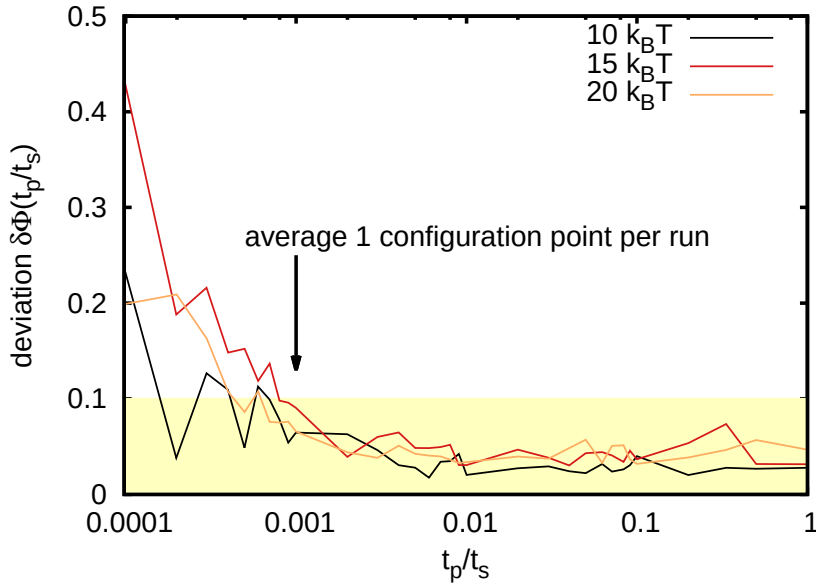

Figure 3.6: Deviation $\delta\Phi$ for the escape flux with different partial run length t_p of the parallel runs, compared to a serial escape run with length $t_s = \sum t_p$. Note, that there is a basic noise in the escape flux when repeating the simulations, therefore the curves do not converge to 0, but the shaded area denotes the acceptable error domain. For the different barrier heights, t_s was chosen such that the same order of configuration points was collected on λ_A, namely $N_0 \approx 1000$. This leads to longer escape runs for higher barriers to collect the same number of points. Note, that at least 1 configuration point should be collected per escape trace within t_p in this case.

of the runs are longer for higher i in this example. This arises from a steeper part of the energy landscape at the beginning, where the runs quickly fall back to λ_A and therefore have a short duration. In contrast, at the top of the energy barrier the free energy landscape is rather flat, leading to longer runs at these points.

For a *synchronous* parallelization scheme this is impossible to implement in an efficient way, because we always have to wait for the last run. In Sec. 3.1 we have performed these runs in bunches of e.g. 5×200 trial runs. However, we will always have to wait for the last run of the interface set to be completed before advancing to the next one, and in the worst case this is a long run which has a duration of e.g. 24 hours. Note, that it isn't allowed to omit trajectories, e.g. aborting the calculation if a certain number of points is collected on the new interface, because this wouldn't account for the information of longer runs. Hence, all runs which have been started by drawing random configurations must be completed.

The aforementioned facts can lead to situations, where calculation clients in paral-

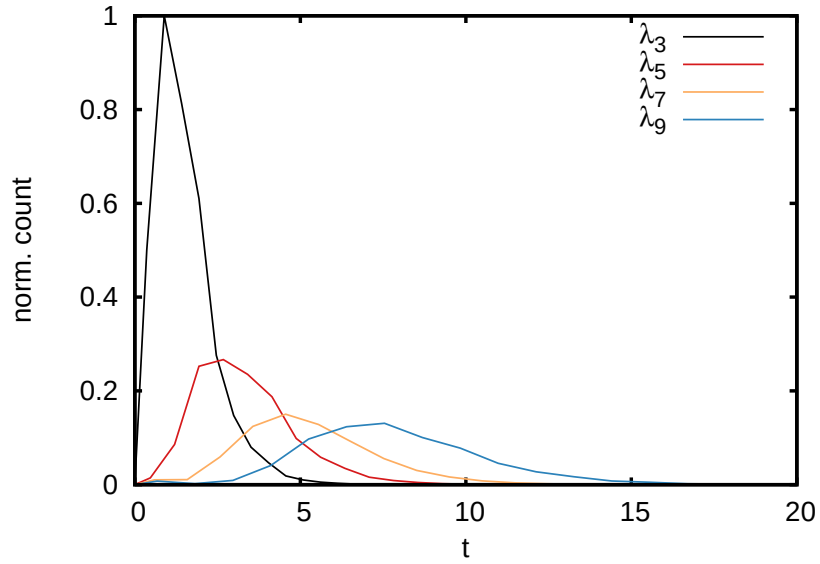

Figure 3.7: Distribution of the duration of FFS runs per interface λ_i for a real FFS simulation, measured in simulation time t ($\tau = 0.01$). The length of the runs is unknown beforehand and differs at a certain interface as well as for different interfaces, e.g. the main part of the runs are longer for a higher interface number i in this example. This must be considered for parallelization.

lel simulations have to wait. In our case, we will steer the parallel simulation *asynchronously*, which means that we start a new simulation run as soon as we have calculation resources available, e.g. if a run is completed and the processing unit is free again.

To bridge waiting times in our simulations we use pre-calculations on already collected points if resources become available, so-called 'ghost runs', which will be described in more detail in Sec. 4.2.1. Before that, we introduce the automatic, optimized interface placement for FFS simulations which will be also implemented in this asynchronous scheme.

3.3 FFS: Automatic, optimized interface placement

In FFS, the efficiency of the simulation depends sensitively on the location of the interfaces, because statistical error $\mathcal{V}$ and computational effort $\mathcal{C}$ depend on the interface placement (see also Sec. 2.5.2). Until now, we only have mentioned that the interface positions λ_i can be adapted after the analysis of a complete FFS simulation run. Borrero and Escobedo developed a procedure where the simulation is performed with an initial (trial) set of interfaces and the positions of the interfaces are corrected

afterwards in an iterative way [91, 92]. However, if the simulation is not even able to complete with such a trial interface set because of a very high energy barrier within the unknown energy landscape and of the simulation being expensive itself (e.g. due to electrostatic interactions like in our case), this strategy can't be applied, therefore the first interface set should be already reasonably optimal.

In this section, we present two alternative methods for determining an optimized set of interfaces automatically and adaptively. These methods can be applied on-the-fly during the simulation run, without knowledge of the underlying energy landscape of the system and without user intervention.

The methods which we developed allow the FFS simulation to progress efficiently through phase space while placing more interfaces in bottleneck regions with the aim of a constant net flux across all interfaces. We will now present the theoretical arguments which are the fundament of the methods. Then, we introduce the methods for the interface placement and discuss the operation scenarios and advantages of each method. Finally, we demonstrate the applicability and performance with two examples, a single-particle test problem and for a computationally demanding crystallization problem of Yukawa particles and then, we draw our conclusions.

3.3.1 Theoretical arguments and optimization principles

While setting up an FFS simulation, the question of the optimal interface positions arises. Interfaces are placed arbitrarily in many cases, or according to an already known underlying energy landscape, e.g. if one knows where the potential is steep, more interfaces can be placed at these parts. However, if the interfaces are placed with a too large spacing, the probability of advancement will be very low, and many trial runs will be wasted because they do not reach the next interface, e.g. firing 1000 trials with 1 successful trial thereof would result in a high computational effort and a high statistical error. In contrast, if the interfaces are placed very close, trajectories can be correlated between the particular interfaces which leads to a bad statistics and little new information can be obtained from this.

Optimal transition probability

Our goal is now to find the optimal transition probability p which maximizes the efficiency $\mathcal{E}(p)$, assuming that the transition probability is the same for all interfaces, $p_i = p$. Note, that in contrast to Borrero and Escobedo [91], the number of interfaces n is not fixed in our case, but determined for our hypothetical rare event problem from

$$n = \frac{\log P_B}{\log p} \tag{3.3}$$

which follows from Eq. (2.65). The computational efficiency is now dependent on p,

$$\mathcal{E}(p) = \frac{1}{\mathcal{C}(p)\mathcal{V}(p)}. \tag{3.4}$$

We have now to calculate $\mathcal{C}(p)$ and $\mathcal{V}(p)$ to determine the dependence of the efficiency on p. Assuming that the number of trials M is the same for each interface we can write

$$\mathcal{C}(p) \approx N_0 R + M \sum_{i=1}^{n-1} C_i(p) \tag{3.5}$$

where

$$C_i \approx \frac{S}{n}[p + i(1-p)], \tag{3.6}$$

with the proportionality constant S/n where S is the cost of a trajectory from A to B. For details refer to [94]. With Eq. (3.5) and Eq. (3.6) we obtain

$$\begin{aligned} \mathcal{C}(p) &\approx N_0 R + \frac{SM}{n} \sum_{i=1}^{n-1} [p + i(1-p)] \\ &= N_0 \left(R + \frac{Sk}{2n} [2p(n-1) + n(n-1)(1-p)] \right) \end{aligned} \tag{3.7}$$

where $k \equiv M/N_0$. Applying Eq. (3.3) leads to

$$\begin{aligned} \mathcal{C}(p) = \frac{N_0}{2 \log p \log P_B} \cdot [&2R \log P_B \log p + Sk(3p \log P_B \log p \\ &+ \log {P_B}^2 - p \log {P_B}^2 - 2 \log p^2 - \log P_B \log p)]. \end{aligned} \tag{3.8}$$

The statistical error is obtained straight forward from Eq. (2.71) with $p_i = p$ and $M_i = M$:

$$\mathcal{V}(p) = \frac{1}{Mp}(n-1)(1-p) = \frac{(1-p)}{N_0 kp} \left(\frac{\log P_B}{\log p} - 1 \right). \tag{3.9}$$

Eq. (3.8) and Eq. (3.9) can be combined via Eq. (3.4) to

$$\begin{aligned} \mathcal{E}(p) = (2kp \log P_B \log p^2)[&(p-1)(\log P_B - \log p)(\log P_B \log p(Sk(1-3p) - 2R) \\ &+ Sk \log {P_B}^2 (p-1) + 2Skp \log p^2)]^{-1} \end{aligned} \tag{3.10}$$

Fig. 3.8 shows the trend of $\mathcal{E}(p)$ for p in the range $(0, 1)$ for a hypothetical rare event simulation with $N_0 = 100$, $M = 200$ and various values of R, S and P_B. The trend of $\mathcal{E}(p)$ is a balance of $\mathcal{C}(p)$ and $\mathcal{V}(p)$, where the former one increases with a higher transition probability p and hence more interfaces, and the latter one decreases with more interfaces at higher p.

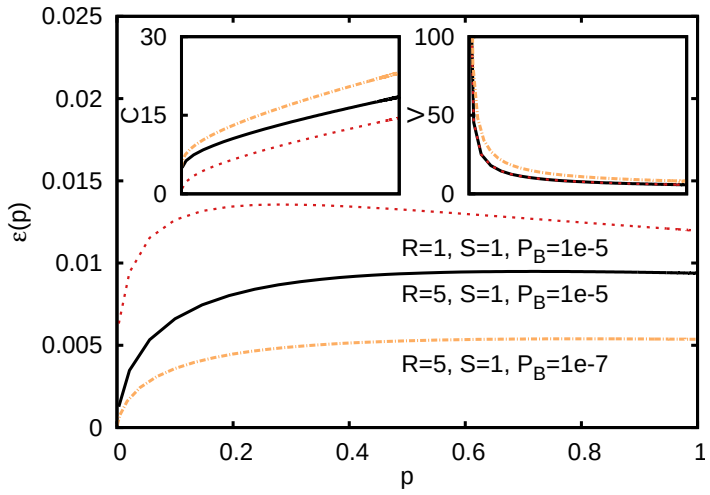

Figure 3.8: Analytical prediction for the efficiency $\mathcal{E}(p)$ (Eq. (3.10)) of a hypothetical rare event simulation (for details see main text). The insets show the predicted computational cost $\mathcal{C}(p)$ (Eq. (3.8)) and statistical error $\mathcal{V}(p)$ (Eq. (3.9)) which contribute to $\mathcal{E}(p)$. The efficiency is rather insensitive over a broad range. However, very low values of p should be avoided.

We find, that low values of p (in this case $p < 0.2$) should be avoided for a good efficiency, but that the efficiency is high in a broad range otherwise. In our case for the hypothetical rare event problem there exists a weak maximum at a certain p. The upper limit for p should be set at a minimal interface distance where trajectories are still able to decorrelate, we will address this later. From these considerations we suggest to choose a value of $0.3 < p < 0.7$ as a rule of thumb, which can be corrected if more detailed knowledge about the particular simulation like decorrelation information is available.

Note, that $p = 0$ would mean that no run reaches the next interface, therefore the interfaces should be placed with an infinite distance which can't be achieved between λ_A and λ_B. In addition, Eq. (3.3) is not defined for $p = 0$. On the other hand, $p = 1$ would mean an infinitely large number of interfaces, because every run should be successful then. In Eq. (3.3) this would be a division by zero.

3.3.2 On-the-fly interface placement algorithms

We now introduce two algorithms to determine the interface positions on-the-fly during the simulation in the context of a desired target transition probability p. The algorithms start at the last known interface position λ_i, which is in the most basic case the border of state A at λ_A, which implies that no interface must be defined by the user when setting up an FFS simulation. Only λ_A and the border of state B, λ_B, as well as a target probability (range) of the desired values of p and a minimal distance

between the interfaces to avoid correlations must be defined. Note, that with these methods, the number n of total interfaces is automatically determined by the choice of p.

Interfaces are placed in an iterative way, e.g. starting from $\lambda_A \equiv \lambda_0$, the interface position λ_1 is determined, next the simulation for the transition $(\lambda_1|\lambda_0)$ is carried out, then the interface position λ_2 is determined, and so on. The stop criterion is reached if the new interface position would be $\lambda_{i+1} \geq \lambda_n \equiv \lambda_B$.

To estimate the transition probability p for placing an interface at an optimized location λ_{i+1}, a small number of trial runs are fired for both algorithms. From the outcome of these 'exploratory' trajectories, the dependence of λ_{i+1} on p can be determined. The difference of the algorithms is the way how these trial runs are performed and how the extracted information is used.

3.3.3 Trial interface method

For the "trial interface" method we allow the transition probability p to be in a certain range, $p_{\min} < p < p_{\max}$, which can be specified by the user according to the requirements of the simulation. To obtain the desired transition probability we place a trial interface at the position λ_{trial}, fire a few number of trial runs and shift the trial interface according to the outcome of these trial runs until the estimated probability p_{est} obtained by this procedure lies in the desired range of p. The algorithm of the "trial interface" method is visualized in Fig. 3.9.

Algorithm

Starting at the last known interface λ_i the trial interface algorithm proceeds in the following way:

1. Begin with a trial interface position λ_{trial} as a candidate for the next interface λ_{i+1} with $\lambda_i < \lambda_{\text{trial}} < \lambda_B$. This initial position is dependent on the simulation problem and could be e.g. $\lambda_{\text{trial}} = \lambda_i + b \times (\lambda_B - \lambda_A)$, where $0.01 < b < 0.1$, or another first guess can be for example $\lambda_{\text{trial}} = \lambda_i + (\lambda_i - \lambda_{i-1})$.

2. Launch M_{trial} trial runs starting from configurations at the last known interface λ_i. The abort criterion is the same like in the standard FFS method, the runs are terminated on the next interface which is λ_{trial} in this case or if they fall back to λ_A. M_{trial} should be much smaller than the number of trials for the real simulation. However, note that the resolution of the estimate depends on this number, e.g. if 10 trials are fired the resolution for p_{des} is 0.1.

3. Calculate $p_{\text{est}} = N_{\text{S}}/M_{\text{trial}}$ with the number of successful trajectories N_{S} which reached λ_{trial}.

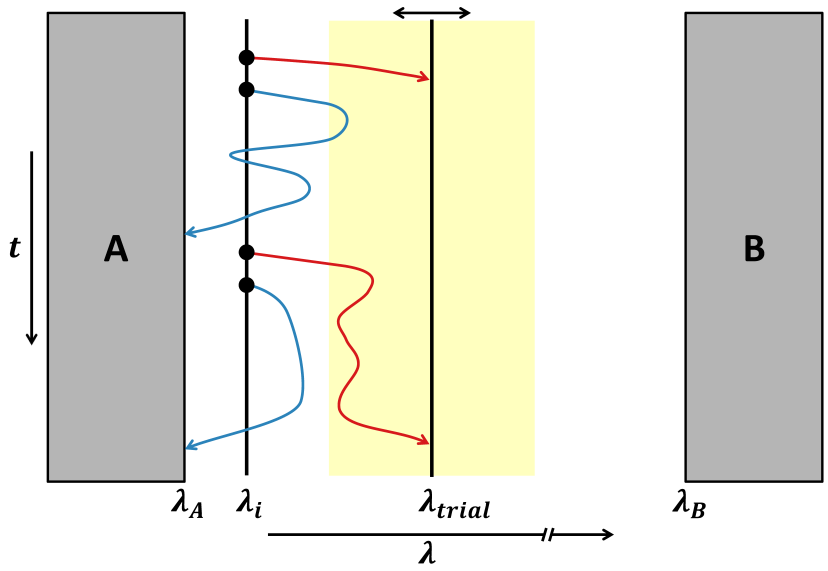

Figure 3.9: Visualization of the "trial interface" method, schematic. A small number of trial runs are launched from the last known interface position λ_i to estimate the probability p_{est} to reach a trial interface at λ_{trial}. In this illustration, two runs (red arrows) reach the interface and two runs (blue arrows) fall back to A, hence $p_{\text{est}} = 0.5$. The trial interface position is varied according to Eq. (3.11) or accepted if p_{est} is in the desired range.

4. If $p_{\min} < p_{\text{est}} < p_{\max}$, the trial interface position can be accepted. Otherwise, a new trial interface position must be chosen via

$$\lambda_{\text{trial, new}} = \lambda_{\text{trial, old}} + \lambda_{\text{step}} \Delta p, \tag{3.11}$$

with

$$\Delta p = \begin{cases} (p_{\text{est}} - p_{\max}) & \text{if } p_{\text{est}} > p_{\max} \\ (p_{\text{est}} - p_{\min}) & \text{if } p_{\text{est}} < p_{\min}. \end{cases} \tag{3.12}$$

Then, the procedure of firing trial runs and checking the estimate p_{est} is repeated until p_{est} is in the desired range. If $\lambda_{\text{trial, new}} < \lambda_i + d_{\min}$, set $\lambda_{\text{trial}} = \lambda_i + d_{\min}$. The above mentioned distance $d_{\min}$ is the minimal interface distance which makes sense in the simulation, e.g. to avoid correlations between the trajectories. If $\lambda_{\text{trial, new}} \geq \lambda_B$ set $\lambda_{\text{trial}} = \lambda_B$.

5. Set the next interface position $\lambda_{i+1} = \lambda_{\text{trial}}$.

6. Proceed with the real simulation by firing the whole M trajectories to λ_{i+1} to calculate p_i and to collect a new set of configuration points on λ_{i+1}. The previous trajectories to this new interface can be included in the number of M trial runs.

Parameters and minimal distance

For this algorithm the parameters specified by the user are $p_{\min}$ and $p_{\max}$, the adjustment step width λ_{step}, the number of trial runs M_{trial} for the estimation of p_{est} and the minimal interface spacing distance $d_{\min}$. This minimal distance $d_{\min}$ depends on the system's dynamics, e.g. a trajectory should not be able to cross several interfaces at once in positive B direction. In addition, $d_{\min}$ should be at least 1 for systems with a discrete order parameter. It is also possible to determine $d_{\min}$ for each new interface position by performing a short study of the systems dynamics, because this can be dependent on the current state in phase space.

A good guideline for the minimal distance is the maximal fluctuation of the order parameter at a certain stage of the FFS scheme. Furthermore, runs usually do not end on the exact value of λ_i, because the stop criterion is $\lambda \geq \lambda_i$, which means that runs are able to overshoot the current interface position, depending on the fluctuations of the order parameter (which is also related to the timestep of the simulation). Therefore, it can be a good idea to use at least the maximal value of the order parameter on the last known interface minus the interface location itself as a minimal distance $d_{\min}$.

The interface shifting criterion of point 4 in the algorithm is expected to work for systems with a steep energy barrier at the beginning, like in our case the crystallization problem. However, it can be flexibly exchanged with other rules which fit the simulated system, e.g. if the system has a flatter barrier it could be useful to use a bisectional scheme, where the trial interface λ_{trial} is placed in the middle of the last known interface λ_i and the border of the final state λ_B. Then the interfaces are placed not that close at the beginning. The shifting can then also be applied according to a bisection by placing the next λ_{trial} between either λ_{trial} and λ_i or λ_{trial} and λ_B. Note, that depending on the system setup (e.g. if state B is far away), the trial interface λ_{trial} could be placed too far apart at the first iteration of the placement algorithm in this case.

Pros and cons of the trial interface method

The great advantage of the trial interface method is, that it works like an original FFS simulation, e.g. runs are fired to a new interface position, which is λ_{trial} in this case, and aborted if the interface position is reached or the runs fall back to A. Therefore, only minor modifications must be made to an already available FFS setup, because the method is conceptually simple. From the computational side the advantage is, that trajectories which are fired to the final accepted interface can be reused directly for the real simulation run to calculate the transition probability.

A drawback of the method is the already discussed step 4 of the algorithm above. If the first estimate of λ_{trial} is not chosen adequately, the algorithm may need some iterations to find the new interface position λ_{i+1}, and therefore preliminary tests should be made to find a good concept for the particular simulation setup.

We advance now to the "exploring scouts" method, where this challenge is avoided,

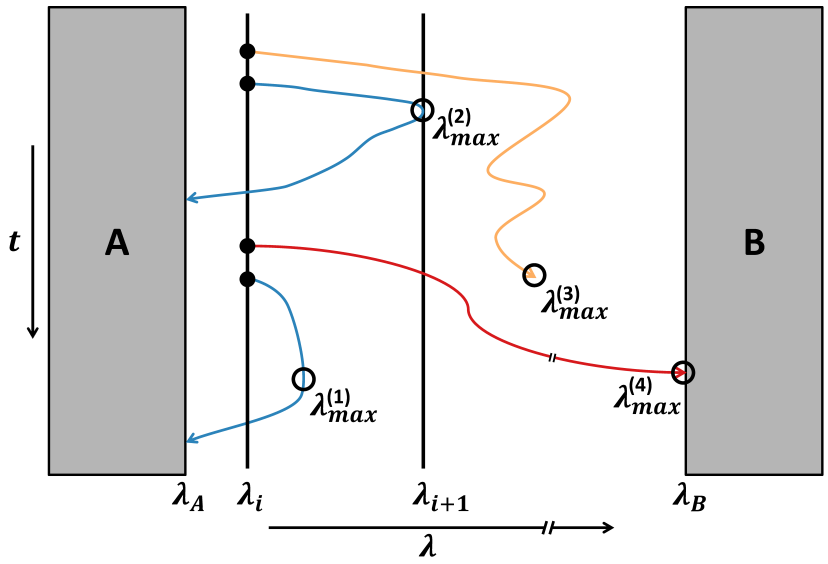

Figure 3.10: Visualization of the "exploring scouts" method. A small number of trial runs are launched from the last known interface position λ_i while monitoring their distribution of the order parameter λ. The abort criteria are either to reach λ_A (blue arrows), λ_B (red arrow) or a user-defined maximum number of calculation steps (orange arrow). Then the particular maximum values of λ are reported (depicted by circles in the illustration), which are used to determine the new interface position (e.g. $p = 0.75$ for λ_{i+1} in this case).

but more settings of the FFS setup must be changed.

3.3.4 Exploring scouts method

The "exploring scouts" method is also based on firing a small number M_{trial} of trial runs from the last known interface position λ_i. However, we do not place a trial interface in this algorithm, the runs are just performed while monitoring the distribution of the order parameter λ. The runs are aborted if they either reach λ_A, λ_B, or a maximum number of calculation steps are performed. This maximum number is defined by the user according to the simulation problem. From these "exploring scouts" the probability of reaching a certain value of λ is obtained, which can be used to place the new interface λ_i. Fig. 3.10 gives a schematic overview of the algorithm. The maximum value of the distribution of λ from each run is used to position the interface such that the transition probability p for the new interface position is close to a desired value p_{des} (which could be the arithmetic average of the acceptable probability range like used for the trial interface method).

Algorithm

From a last known interface λ_i the exploring scouts algorithm proceeds in the following way:

1. Launch M_{trial} trial runs starting at the configurations of the last known interface λ_i. The trial runs are completed, if λ_A or λ_B are reached, or if a maximum number of calculation steps $m_{\max}$ are performed. Store the maximum values of the order parameter λ for each run.

2. Store all maximum values of the trajectories in a ranked list with an index k for each trajectory in the range $0 < k < M_{\text{trial}}$, with $\lambda_{\max}^{(k)} < \lambda_{\max}^{(k+1)}$.

3. Compute the index of the maximum value which is closest to the desired transition probability value, $k_{\text{des}} = \lfloor M_{\text{trial}}(1 - p_{\text{des}}) \rfloor$. The position of the next interface can then be set to $\lambda_{i+1} = \lambda_{\max}^{(k_{\text{des}})}$. If the new value $\lambda_{i+1} < \lambda_i + d_{\min}$, set $\lambda_{i+1} = \lambda_i + d_{\min}$ (where $d_{\min}$ is the minimal interface distance like in the trial interface method).

4. Continue with the standard FFS simulation by firing the complete M trials to λ_{i+1} to obtain the real transition probability and a new set of configuration points on λ_{i+1}.

Note, that for entry k in the ranked list[4], k exploring scouts fail to reach $\lambda_{\max}^{(k)}$ and $M_{\text{trial}} - k$ scouts reach at least $\lambda_{\max}^{(k)}$ or higher values. In the description of the algorithm above, we choose the value of the $\lfloor M_{\text{trial}}(1 - p_{\text{des}}) \rfloor$-th entry which is stored in the ranked list as the new interface position λ_{i+1}. However, one could also think of more advanced selecting algorithms, e.g. by interpolating between the values in the ranked list for the new interface position. In most cases this should not be necessary because the real transition probability will be slightly different from the estimated value in this algorithm anyway.

Parameters

For the minimal distance $d_{\min}$ the same rules from the trial interface method like described above are valid. Further parameters in the exploring scouts method are the desired target probability p_{des}, the number of exploring scouts M_{trial} and the maximum number of simulation steps for a trial trajectory $m_{\max}$. This number should be chosen such that enough information can be gained from an exploring scout. If $m_{\max}$ is set too low, values of larger λ won't probably be explored, and the new interface λ_{i+1} will be placed too close at the last interface λ_i. In contrast, if $m_{\max}$ is chosen too large, the algorithm could become computationally expensive, e.g. if runs are not able to reach λ_A or λ_B within this number of steps, the full number of steps must be calculated.

[4] k runs from zero to $M_{\text{trial}} - 1$

Pros and cons of the exploring scouts method

The great advantage of the exploring scouts method is the fact that it is already predetermined how many trial runs will be required to determine the new interface position λ_{i+1}, because we only need the information of the maximum values of the order parameter which were reached during these runs to place the interface. In addition, the number of parameters which must be specified for this placement method is lower than in the trial interface method. A rather technical drawback is, that the exploring scouts method requires more modifications to a standard FFS simulation, because the distribution of the order parameter λ has to be monitored and the maximum values must be reported by the runs.

3.3.5 Examples

In the following examples we demonstrate the application of the interface placement methods with two examples. First, we present the results of a fundamental problem, namely a single particle moving according to Langevin dynamics (see also Sec. 2.2) in a 1D potential with an energy barrier towards state B. In this example, we also verify the analytical predictions for the efficiency (see also Sec. 3.3.1) by simulations. In the second example, we apply the algorithms to the crystallization of Yukawa particles to demonstrate the usage of the interface placement in a 'real' system.

1D particle in a potential with an energy barrier towards B

In this section we use the same particle system like described in Sec. 3.1.1 with the potential $U(x) = (h/2)\left[1 - \cos(\pi x)\right]$ which has two minima in the x-range $[-1, 3]$, but this time with a barrier height of $h = 12k_BT$ at $x = 1$. For a visualization of the potential $U(x)$ and the simulation problem see also Fig. 3.2. Like in Sec. 3.1.1, we set the border of the states A and B at $\lambda_A = 0.2$ and $\lambda_B = 2.0$. For the simulation we set $k_B = 1$, $T = 1$, $m = 1$, $dt = 0.001$, $\xi = 1$ and as order parameter we set $\lambda = x$.

In contrast to the problem in Sec. 3.1.1, we do not define interface positions λ_i, but use our algorithms to determine the optimized locations. Therefore, we perform DFFS simulations in combination with our methods. The common parameters for the methods are the number of trial runs $M_{\text{trial}} = 50$, the number of real runs $M = 500$, the number of runs for λ_A $N_0 = 250$ and the minimal interface distance $d_{\min} = 0.01$. For the trial interface method we set the acceptance range for p to $[0.4, 0.6]$ and the first trial interface position is calculated via

$$\lambda_i + 0.1(\lambda_B - \lambda_i). \tag{3.13}$$

For the exploring scouts method we use a desired target probability of $p_{\text{des}} = 0.5$ and a maximum number of simulation steps for the exploratory runs of $m_{\max} = 1000$.

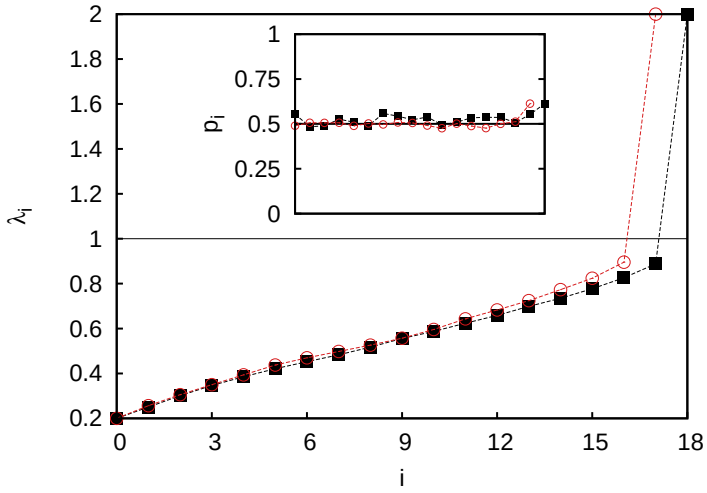

Figure 3.11: Interface locations λ_i and transition probabilities p_i (inset) as a function of i, determined by the trial interface and exploring scouts method in the single particle simulation. The main part of the interfaces is located at the left-hand side of the barrier with $\lambda_i \leq 1$ and the probabilities p_i are distributed around the desired value of $p_{\mathrm{des}} = 0.5$.

The interface positions λ_i which have been found by both our algorithms are shown in the main plot of Fig. 3.11 as well as the transition probabilities p_i. We find, that the bulk of interfaces is located at the ascent domain of the barrier until the top of the barrier. This is a result of the constant flux requirement where it is necessary to place more interfaces at bottlenecks like the steep part of the potential. Therefore, these methods can be used for a hint, where the top of the barrier is approximately located.

In both methods, the transition probabilities p_i are distributed in the accepted domain and around the desired value p_{des}, respectively. The last probabilities are slightly higher because the steep part of the barrier has been left behind for these transitions.

Fig. 3.12 shows the reference functions f_i for constant flux (Eq. (2.72)) for the trial interface and the exploring scouts method. We see, that the trend of f_i is nearly linear in both cases, which means that the interfaces have been placed close to optimal, and further optimizations are not required. Note, that there are statistical fluctuations in the probabilities and therefore they can't be exactly linear.

As we have seen, we are able to obtain a certain transition probability p automatically according to our specification. This can be used to verify the analytical predictions of the p-dependent quantities which we derived above, e.g. the efficiency $\mathcal{E}(p)$ of an FFS simulation. To this aim, we use the exploring scouts method and vary the target probability p_{des} in the range $[0.05, 0.95]$, which means the method has to place a very low number of interfaces at $p_{\mathrm{des}} = 0.05$ and a very large number of inter-

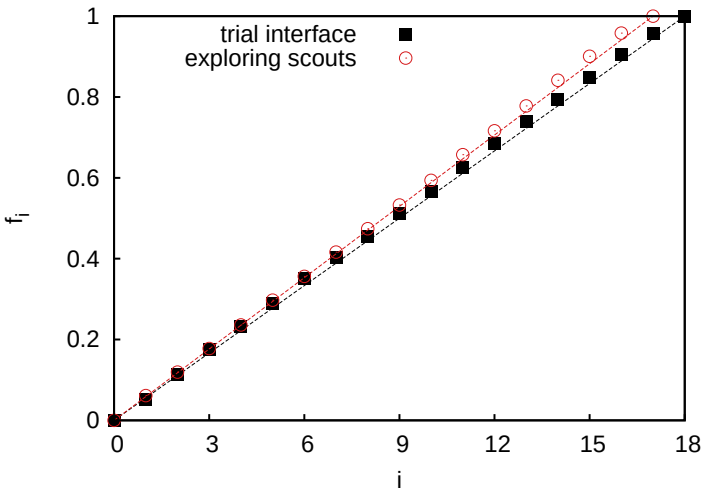

Figure 3.12: Reference function f_i (Eq. (2.72)) for constant net flux of the automatic interface placement methods. The dashed lines would be the optimal case where $f_i = i/n$. Note, that depending on the fluctuations of p, the number of interfaces n can differ.

faces at p_{des} to obtain these probabilities. For a good accuracy we set $N_0 = 3000$ and $M = 1000$. In the simulation we measure the computational cost $\mathcal{C}(p)$ in simulation steps and the statistical error $\mathcal{V}$ in the rate to obtain $\mathcal{E}(p)$ (Eq. (3.4)).

Fig. 3.13 shows the comparison of the result of this simulation to the analytical prediction of $\mathcal{E}(p)$ (Eq. (3.10)). For the comparison we used $P_B = 1.36 \times 10^{-5}$ which was obtained from our simulations. The parameters $R = 1.60 \times 10^7$ and $S = 1.39 \times 10^7$ in simulation steps were obtained by fitting the function of the computational cost $\mathcal{C}$ (Eq. (3.8)) of the hypothetical rare event problem to the simulation data.

As we can see, the results of the simulations are in good agreement with our analytical calculations, which shows that the transition probability p can indeed be used as tuning parameter and is related to the efficiency. Note, that we obtained the same P_B within the error range in our FFS simulations for a different number of interfaces n, which emphasizes the applicability of FFS.

Crystallization of Yukawa particles

With our analytical model for the efficiency of an FFS simulation confirmed and with having the placement methods tested on a fundamental problem in the section before, we move now on to a 'real' simulation problem, namely by applying the methods to crystallization simulations using the model of charged macromolecules like described in Sec. 2.3.2, where we use a combination of Yukawa potential and Weeks-Chandler-Andersen (WCA) potential, $U(r) = U_{\text{Yukawa}}(r) + U_{\text{WCA}}(r)$ (see Eq. (2.32)

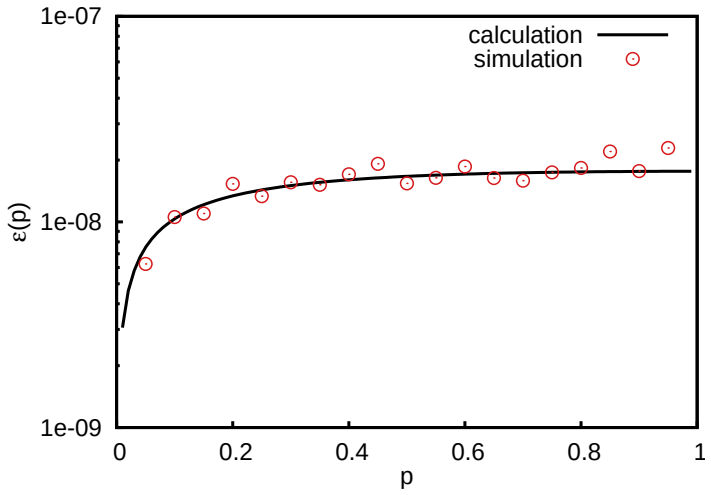

Figure 3.13: Comparison of the simulation results for $\mathcal{E}(p)$ with the analytical calculation (Eq. (3.10)), obtained with the single particle simulations and $P_B = 1.36 \times 10^{-5}$, $R = 1.60 \times 10^7$ and $S = 1.39 \times 10^7$. Thereby, the automatic placement method placed between 3 (for $p = 0.05$) and 671 (for $p = 0.95$) interfaces.

and Eq. (2.33)).

For this simulation, we use the values of the Yukawa potential $\epsilon = 8$ and the inverse screening length $\kappa = 5$ at a pressure of $P = 38$. We perform our MD simulations using ESPResSo [17] with 4096 WCA-Yukawa particles in a 3D box with periodic boundaries in an NPT ensemble and Langevin dynamics[5]. We use our parallel FFS algorithm in the context of the rare event sampling framework FRESHS [98].

With the parameters for the simulation above, the crystallization is already a rare event, and setting up an FFS simulation is difficult because of the unknown underlying energy landscape, e.g. we do not know the size of the critical cluster from which the crystal grows spontaneously. Therefore, we do not know where to place the main portion of the interfaces. Even with 100 evenly spaced interfaces λ_i between λ_A and λ_B this can lead to situations where no single trial of thousand succeeds.

The state A of our system is the liquid phase of the system[6], where the order parameter, which is the number of solid particles in the largest cluster like described in Sec. 2.4.3, is $\lambda < 15$. The final state B of the system is located at $\lambda_B = 3700$, which means that more than 90% of the particles are part of the largest cluster. We collect $N_0 = 80$ configuration points on λ_A and use $M = 50$ trial runs per interface.

[5]The Langevin dynamics includes stochastic fluctuations in the system, which ensures different paths in FFS.

[6]The liquid phase is created by randomly set up the particles in the 3D box and equilibrating the system (first with a capped potential, see also the user's guide from [17]) and verifying the size of the solid clusters in the system.

With these settings, we use FFS to perform the transition from A to B, and hence, crystallize the system. We use the following schemata to place the interfaces:

(i) manual placement of the interfaces *by hand*, using an initial interface set with fixed distances $\Delta\lambda$ of the order parameter, e.g. $\Delta\lambda = 30$ and then moving the interfaces closer if the transition does not succeed,

(ii) manual placement in a *logarithmic* way, because as a pre-knowledge we assume that we have to place more interfaces at the beginning to build up a cluster until the critical size,

(iii) automatic placement with the *trial interface* method,

(iv) automatic placement with the *exploring scouts* method.

With these schemata we are able to compare the quality of our automatic placement methods with conventional (manual) FFS simulation setups, which are used by many other groups to perform their FFS simulations. As a measure of quality, which is determined by a constant net flux across the interfaces (see also Sec. 2.5.2) we use the function f_i (Eq. 2.72). Note, that in this section we are only interested in the interface placement results, the crystallization of the system is analyzed in detail in chapter 6.

Fig. 3.14 shows a comparison of the interface positions λ_i of the manual placement methods and the automatic placement methods. From a practical point of view, the problem of placing the interfaces manual by hand (i) was that our simulations are computationally expensive and we don't know the underlying energy landscape. Therefore, we had to interrupt the simulation when it didn't advance several times to adjust the interface placement, which turned out to be very painful on a high performance computing machine. The result of this procedure was in our case a set of interfaces where – depending of course on the initial set – the interfaces are placed too far apart at the beginning which means a low fraction of successful runs and unnecessary interfaces at the end, when the barrier has been overcome.

The manual logarithmic placement (ii) was performed with the knowledge of the automatic placement methods and hence set up with a comparable number of interfaces ($n = 36$) to be used as a reference. Thereby, the interface spacing was chosen to be closer at the beginning because of the steep part of the energy barrier. The idea of choosing a logarithmic set of interfaces was caused by the fact that the simulation didn't actually succeed with 100 evenly spaced interfaces between λ_A and λ_B.

The automatic placement methods (iii)+(iv) placed interfaces only at positions where they are required, namely at the steep part (bottlenecks) of the barrier which is closer to state A, e.g. in this case there is no intermediate interface beyond $\lambda = 1120$.

Fig. 3.15 shows the corresponding functions f_i and transition probabilities p_i for each case. For the manual placement scenarios (Fig. 3.15(a) and Fig. 3.15(b)) we obtain a characteristics of f_i which is not linear (this would be the dashed line in the figure),

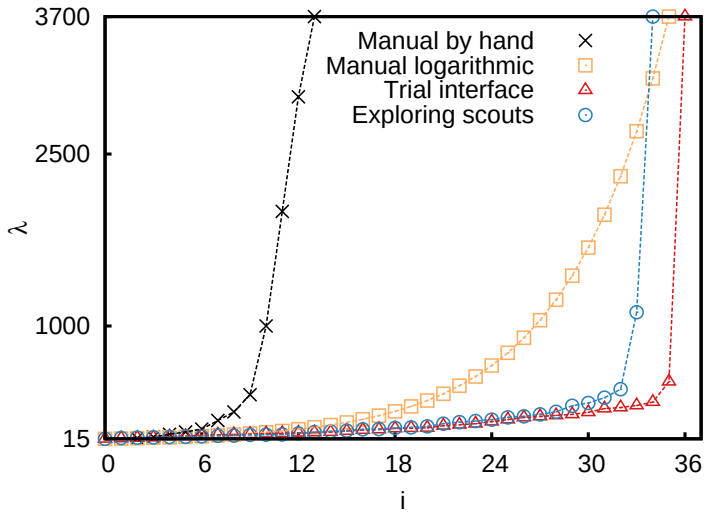

Figure 3.14: Interface positions of the different placement schemata (see main text). By using method (i) by hand, the smallest number of interfaces was placed. Method (ii) is the logarithmic case where more interfaces are placed at the beginning. With automatic placement using (iii) trial interface and (iv) exploring scouts, the larger portion of interfaces is placed at the beginning until the critical cluster size has been overcome, then no more interfaces are necessary for the spontaneous growth.

which means that the net flux across the interfaces is not constant. The associated transition probabilities p_i which are shown in the insets are far from equal because we didn't place enough interfaces in bottleneck regions. This leads to $p \to 0$ and hence a bad sampling with a resulting high computational effort $\mathcal{C}$ and a large statistical error $\mathcal{V}$. In addition, the interfaces placed with $p \to 1$ are unnecessary and lead to a higher cost $\mathcal{C}$ as well as a computational overhead due to storing configuration points on these particular interfaces and launching the appropriate trial runs. Note, that this manual placement method shouldn't be used, because it can potentially bias the FFS simulation due to only shifting the interfaces where we have too few successes, but we don't touch the interfaces on which we collect by chance a large number of successes[7].

The results of the automatic interface placement methods (Fig. 3.15(c), Fig. 3.15(d) and also Fig. 3.14) were generated using $M_{\text{trial}} = 8$ trial runs and a minimal distance of the interfaces $d_{\text{min}} = 3$ to avoid correlations of the runs. For the trial interface method we set the acceptance range to $0.3 \leq p \leq 0.6$ and the trial interface position was chosen via Eq. (3.13). The destination probability for the exploring scouts was set to $p_{\text{des}} = 0.45$, where each exploring scout was allowed to perform a maximum number of $m_{\text{max}} = 10000$ steps. With these parameters, we obtain for both methods a

[7] If such a scheme was used we recommend to repeat the complete FFS simulation after the interfaces have been fixed.

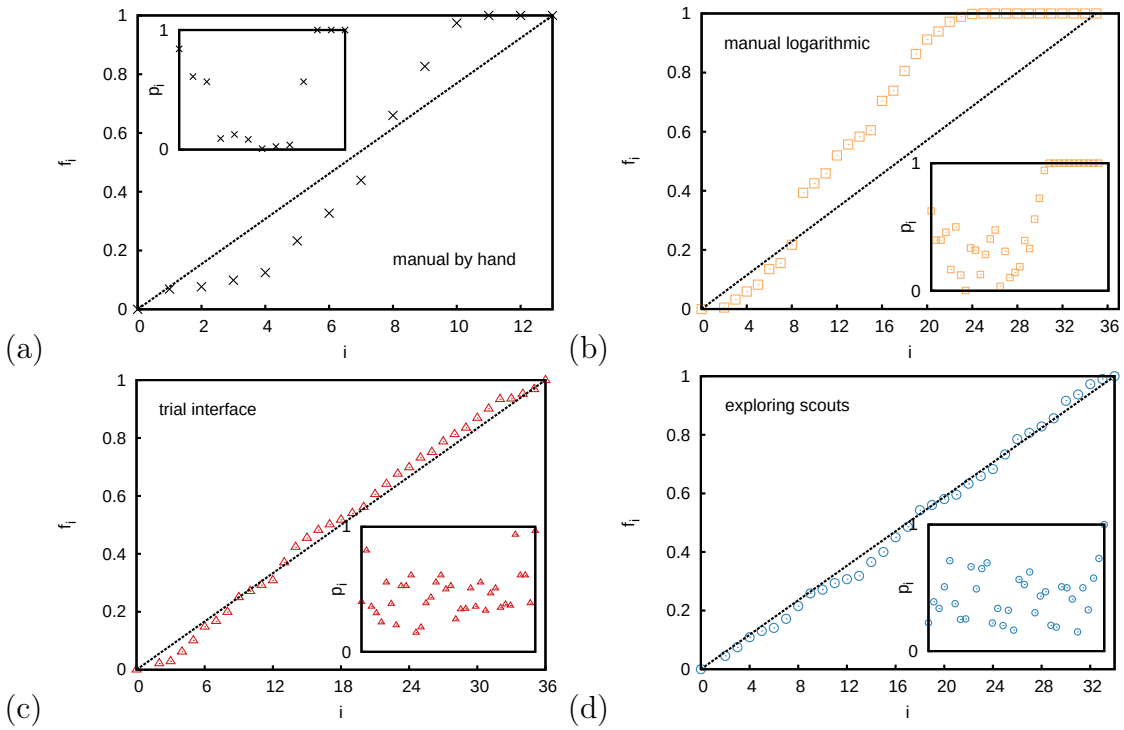

Figure 3.15: Comparison of the quality of the placement methods (i)-(iv) with the help of the net flux function f_i (Eq. (2.72)). The dashed lines show the optimal case, where the net flux across the interfaces would be constant. The insets show the transition probabilities p_i for each interface transition $(\lambda_{i+1}|\lambda_i)$. A probability of $p_i \approx 1$ stands for an unnecessary placed interface, $p_i \approx 0$ leads to a bad sampling and hence a large statistical error. This occurs in the manual placement methods but is avoided with our automatic, optimized interface placement algorithms.

set of interfaces λ_i with a linear shape of f_i on-the fly and automatically, which means that the interfaces are already placed at their optimal positions. In addition, the transition probabilities p_i are nicely distributed around our target probability range or value, respectively, and we do not obtain very low numbers of p_i which would have a low efficiency $\mathcal{E}$. The automatic methods do not place interfaces where they aren't necessary, e.g. when the barrier has been overcome and the crystal growth spontaneously[8].

In table 3.3 we show a detailed overview of the results of all methods. The cost $\mathcal{C}$ is measured in simulation steps and includes the computational effort which was necessary for the interface placement. Note, that this includes also the aborted runs

[8]Note, that if $m_{\max}$ is too small for the exploring scouts method, additional interfaces could be placed if the runs become longer, e.g. when the barrier has been overcome and the crystal just grows, then the runs do not reach λ_B before they reach $m_{\max}$.

method		cost $\mathcal{C}$	error $\mathcal{V}$	efficiency $\mathcal{E}$	k_{AB}
(i)	manual by hand	4×10^7	18652	10^{-12}	$5 \times 10^{-11\pm3}$
(ii)	manual logarithmic	7×10^6	6648	10^{-11}	$1 \times 10^{-14\pm2}$
(iii)	trial interface	4×10^6	188	10^{-9}	$6 \times 10^{-14\pm1}$
(iv)	exploring scouts	3×10^6	251	10^{-9}	$2 \times 10^{-14\pm1}$

Table 3.3: Interface placement results for the different methods (i)-(iv) and the Yukawa simulations: computational cost $\mathcal{C}$ (simulation steps), variance in the rate constant $\mathcal{V}$, resulting computational efficiency $\mathcal{E}$, and rate constant k_{AB} (in $\sigma^{-3}\tau^{-1}$ with the simulation time unit τ).

in (i) when an interface position was not used because of a too low success ratio. We see, that $\mathcal{C}$ and $\mathcal{V}$ are significantly lower for the automatic placement methods which leads to an efficiency which is $2-3$ orders of magnitude better than for the manual placement. Note, that in this case the manual placement method (i) leads also to a biased rate constant with a larger error. The computational cost of the algorithms (iii) and (iv) is about a factor of 2 lower than for the logarithmic case (ii), where we used already pre-knowledge, and one order of magnitude lower than for the manual case (i), where we placed the interfaces by hand without pre-knowledge. For this specific simulation, the exploring scouts method required only 75% of the simulation steps of the trial interface method.

3.3.6 Discussion

The automatic, optimized interface placement algorithms facilitate the set-up of an FFS simulation because only the borders λ_A and λ_B of the initial state A and the final state B must be specified in terms of the order parameter λ. Then, the algorithms ratchet the system from A to B automatically and on-the-fly by placing the interfaces of FFS in their optimal locations λ_i. The efficiency $\mathcal{E}$, which is a balance of computational cost $\mathcal{C}$ and statistical error $\mathcal{V}$, is increased tremendously. This denotes a great improvement of the performance of the FFS simulation, which is particularly helpful for rare event simulations, which are in addition computationally demanding and comprise complex trajectories in phase space with unknown bottlenecks.

The presented algorithms are based on launching a small number of exploratory runs to estimate the new optimal position λ_{i+1} of an interface. The first algorithm works like a standard FFS simulation by placing a trial interface and by monitoring if the exploratory runs reach this interface or fall back to A. The second algorithm doesn't use such an interface, but the distribution of the order parameter λ during the exploratory run: The maximum of the order parameter $\lambda_{\max}$ is reported for each run, which is carried out until it falls back to A, reaches B or reaches a given number of maximum steps.

To find the optimal interface location λ_{i+1}, the dependency of the efficiency $\mathcal{E}(p)$ on the transition probability p for a given P_B was investigated, since this quantity is directly related to the interface spacing. Our analytical expressions were confirmed by fundamental simulations of a 1D particle. We find, that p can be used as a tuning parameter for the interface placement and that for the desired target probability a range of $0.3 \leq p \leq 0.7$ should be suited for many applications. In this domain, the values of the efficiency are high and the change of the efficiency is rather insensitive to p. The lower value of the p-range is due to the fact that the efficiency becomes very poor for low values of p. The upper bound should be set such that the trajectories are not correlated due to a small interface spacing, which can also be avoided by setting a minimal user-defined interface distance $d_{\min}$, which should be e.g. at least 1 for a discrete order parameter λ like in crystallization problems, where the order parameter counts the number of monomers in a nucleus, for example.

We demonstrated the applicability of the methods with two examples, namely a fundamental example of a single particle in a 1D potential and a 'real' simulation problem which is computationally more demanding, the crystallization of charged Yukawa macromolecules with an unknown underlying energy landscape. Thereby, the automatic, optimized interface placement increased the efficiency significantly in comparison to the manual placement methods.

This work was based on the findings of Borrero and Escobedo [91], where the interface set was optimized after a first complete simulation run. The post-optimization from this work can also be applied after the interfaces have been found by the automatic placement, if desired or necessary. However, in our simulations this didn't improve the results any more.

Beyond the application to FFS and its flavors, the interface placement algorithms could be used for other rare event sampling methods with both, one-dimensional or multi-dimensional order parameters to determine the way through phase space with unknown density efficiently.

Last but not least these methods can be used to verify the selection of an order parameter or to adjust an order parameter during simulation by using the information from the exploratory runs, on-the-fly.

The algorithms which we introduced in this section can be parallelized and are already implemented in our Flexible Rare Event Sampling Framework [98] in a generic way, which we present in the next chapter.

4 The Flexible Rare Event Sampling Harness System

In this chapter we present our highly efficient and parallel implementation of rare event sampling algorithms in our *Flexible Rare Event Sampling Harness System* (FRESHS), which was developed starting from scratch during this work. The chapter is based on our article [98].

Some parts of the article have been revised and extended, as well as further important application examples have been added. All the results of chapter 3 have been implemented in context of this framework, which is now publicly available[1] and helps also other researchers to accomplish their investigations using rare event sampling, like e.g. [99].

- Kai Kratzer, Joshua T. Berryman, Aaron Taudt, Johannes Zeman and Axel Arnold — The Flexible Rare Event Sampling Harness System (FRESHS). J. Comp. Phys. Comm. 185(7), 1875-1885 (2014)

4.1 Overview

FRESHS has been developed for simulating rare events with algorithms from the 'splitting family', which comprises all methods which are based on calculating *trajectory fragments*. To be able to simulate both quasi-static and dynamic systems in equilibrium and non-equilibrium we primarily implemented the sampling algorithms *Forward Flux Sampling* and *Stochastic Process Rare Event Sampling* (SPRES) [100] to FRESHS, but other rare event sampling methods (e.g. NS-FFS [101]) can be implemented in a flexible and modular way. In this work, we focus only on the Forward Flux Sampling part of FRESHS and show further relevant application examples which are not included in [98]. For the SPRES part and further simulation examples refer to Ref. [98] and to the contents of the FRESHS package, available at [102].

FRESHS is based on the fact, that the sampling flow can be separated from the calculation of the physics in the simulation, which implies that the simulation tool of the user's choice can be used to simulate the trajectories independent of the sampling algorithms. Thereby, simulation tools like the soft-matter MD tools GROMACS [103], LAMMPS [104], ESPResSo or – beyond that – self-written code can be attached to

[1] `http://www.freshs.org`

FRESHS via a *plugin-system*. This creates also a fundament for a comparison of different sampling algorithms or simulation tools for the same simulation problem.

The parallelization in FRESHS is performed by calculating several trajectories at the same time, with multiple instances of the simulation tool. In section 3.1 we already used such a parallelization scheme together with the Science Experimental Grid Laboratory and discussed the issues which occurred, e.g. the synchronous parallelization. To be maximal flexible, the communication in FRESHS is realized via standard networking[2] which allows the use of FRESHS not only on high performance computing (HPC) hardware but also on *heterogeneous* resources in an *asynchronous*[3] way.

Within FRESHS, the rare event sampling information is stored in a database which can be used to track runs, resume the simulation if aborted, reproduce runs or for a versatile analysis in general like we will see in the example section of this chapter.

4.2 Under the hood

In this section we discuss the inner life of our framework. As already said, for FRESHS we separate the calculation of the trajectory fragments from the rare event sampling method. Therefore, we use a *server-client* scheme, where the sampling method is implemented on the server-side and the clients calculate trajectories. Fig. 4.1 gives an overview of the schematic layout of the FRESHS framework. The partitioning in a server-client scheme is possible, because the calculation of the trajectory fragments is independent of the sampling scheme, only a starting point and an abort criterion must be communicated which is performed via standard networking and the help of a so-called *harness script*[4], which steers the simulation.

The logic of the rare event sampling algorithm with the statistical analysis is located in the server, which we will address now.

4.2.1 The server: optimized rare event sampling

The server is initialized with a configuration file and an appropriate sampling module (see also Fig. 4.1). The information during simulation is stored and read from a database, which can be used to monitor, restart and analyze the simulation.

We focus now on the highly efficient FFS implementation of the FRESHS server. In Sec. 3.2 we already described the parallelization of FFS and the drawback, that a configuration point from an interface λ_i can only be drawn at random in an unbiased way, if the set of configurations of λ_i is complete. As we have seen from the broad distributions in Fig. 3.7, the run lengths in FFS can be very different and are

[2] We use a socket-based communication via the TCP/IP protocol.

[3] An example for an asynchronous communication protocol is a chat protocol, where messages can be sent and received at any time.

[4] Details and examples of the harness scripts can be found on `http://www.freshs.org`.

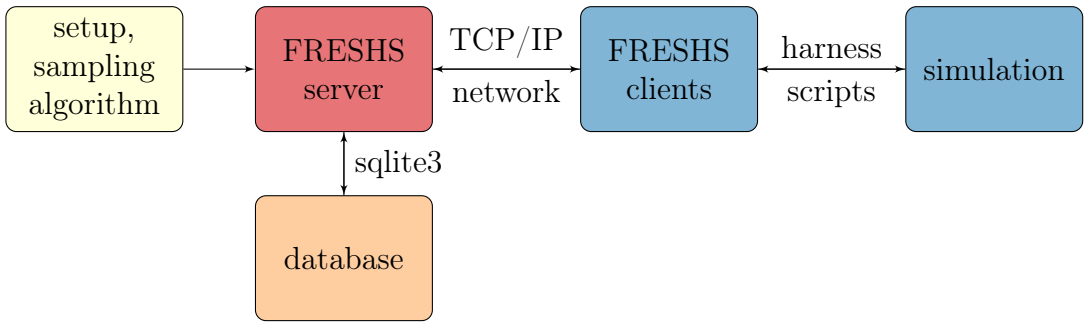

Figure 4.1: Schematic layout of FRESHS. The server (reddish) reads the rare event sampling setup (yellowish) from a configuration file and loads the appropriate rare event sampling algorithm module. During simulation, data is stored in a database (orange), which can be used for on-line monitoring of the simulation and for versatile analysis. Multiple parallel running simulation clients (bluish) are connected to FRESHS via the network, whereas the simulation is steered by the client with an appropriate harness script. Thereby, the simulation itself can also be implemented in parallel.

undetermined beforehand. In the worst case this leads to situations, where one client calculates a long simulation trajectory which is necessary for the last configuration point of the set, while the others would have to wait. Now, we bridge this waiting time using look-ahead runs which we call 'ghost runs'[5].

Ghost runs

To bridge the waiting time in the simulation, we proceed as follows: Assuming that the simulation calculates trajectories for the transition $(\lambda_{i+1}|\lambda_i)$ which is not yet ready because a few runs are missing, we already load configurations from the new interface λ_{i+1} if compute resources become available and calculate the corresponding trajectories. However, the information[6] from these runs is stored in a separate storage location, in our case a second 'ghost' database with the same layout like the original database. Thereby, the starting configuration point for the ghost runs on λ_{i+1} is chosen to be the point with the minimum number of pre-calculated ghosts runs, because our goal is to populate all points on λ_{i+1} with several ghost runs. If the calculation of the transition $(\lambda_{i+1}|\lambda_i)$ is complete, we continue the real simulation be choosing points from the new interface λ_{i+1} on a job call (Fig. 4.2). If a ghost run exists for this point, we copy the entry from the ghost run storage location to the real simulation and with that, use

[5]The name 'ghost' is used to separate these runs from the 'real' runs, because they can't be used directly for the simulation but exist as lead workers which are invisible to the 'real' simulation.

[6]If the run is successful the configuration point with the success information is stored, otherwise the failure information.

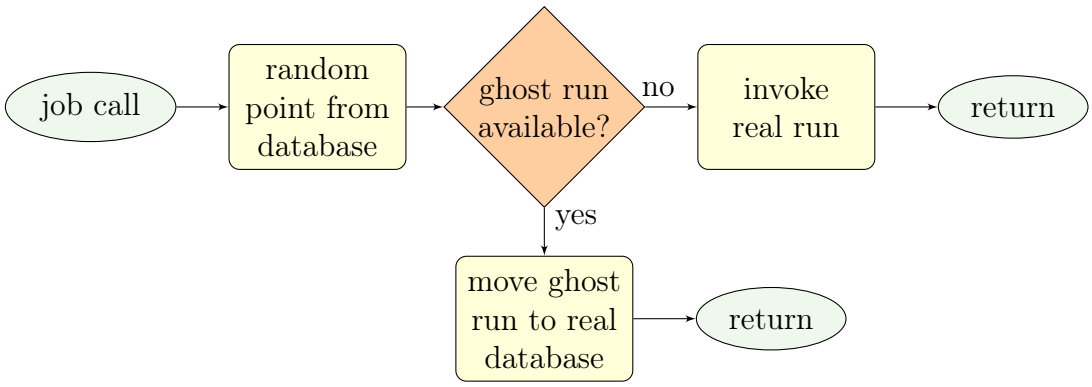

Figure 4.2: Server-side selection of configuration points on a job call with checking for pre-calculated ghost run information. If a ghost run is available on the drawn configuration point the data is used for the real simulation by copying the information from the ghost database to the real database. The call of this scheme is repeated until no more simulation starts are required for the current interface transition.

the pre-calculated information. In a rare event simulation, the configuration points are sampled several times which results in a great efficiency using this scheme. In the extreme case of a lot of clients, the main simulation task would then be reduced to simulate the last (long) runs on the current interface transition.

Fig. 4.3 shows the ratio of calculated integration steps by ghost runs compared to the calculated steps of the real runs in a typical simulation with 20 simulation clients in this case. Here, about 10% of the simulation steps were calculated using ghost runs. Note, that the fraction of calculation steps which is performed by the ghost runs differs in each simulation, because the number of ghost runs performed depends on the available computing resources and hence on the number of clients which are connected to FRESHS, as well as on the randomly distributed trajectory run length for the particular interface of the simulation.

A great advantage of this ghosting scheme is, that clients don't need to know that they are considered as ghosts. In addition, if ghosts are still calculating while the last configuration point is collected on λ_{i+1} they can easily be converted to real simulation runs if the starting point of them is selected during the random draw.

The ghost runs can also be used to adjust the number of sampling points after the complete FFS simulation has already been performed. Conventional FFS simulations must be performed from scratch again, because if one increases only the number of points per interface and simulates again keeping the already sampled points, there would be a large bias because as already mentioned, the configuration points on an interface must be drawn from the *complete set*. The bias can simply be avoided by storing the already calculated trajectories as ghost trajectories and only use them if

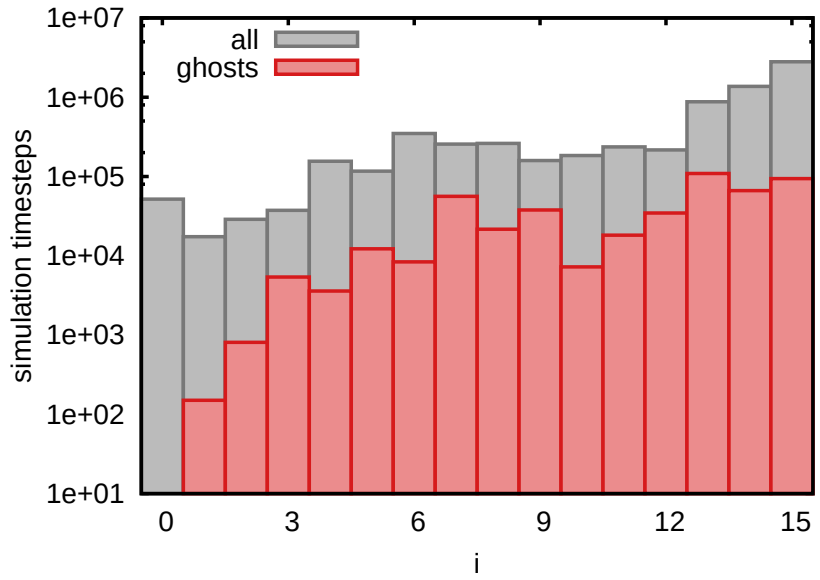

Figure 4.3: Ghost usage (red boxes) in simulation timesteps compared to the real runs (grey boxes) per interface λ_i for a typical simulation scenario consisting of 20 calculation clients in this case. The fraction of ghost run calculation steps depends on the number of clients, the length of the runs per interface and the random distribution of the run lengths, so this ratio differs in every simulation. Here, about 10% of the simulation was performed by ghosts.

the particular point is drawn.

Now, we've introduced a powerful scheme to involve the simulation clients into the calculations and to keep them busy, but a requirement for starting ghost runs is the knowledge about the next interface location λ_{i+1}. In Sec. 3.3 we presented methods to determine the optimal interface location λ_{i+1} automatically and on-the-fly during simulation. Therefore, we address now the point how this can be combined with the ghost runs which leads us to the final calculation job selection scheme.

Automatic optimized interface placement in FRESHS

For a maximum efficiency of the FFS simulation we implemented the methods from [94] which are also described in Sec. 3.3 to FRESHS. At this point, if a calculation client connects to the server, we have the following jobs which must be launched at the appropriate state in the rare event sampling calculation scheme: (i) normal (real) simulation runs, (ii) ghost runs, and (iii) exploring runs which can either be the trial interface runs or the exploring scout runs (see also Sec. 3.3) which can be specified in the server's configuration.

The following is an example, how a highly optimized FFS simulation using FRESHS

can be performed[7], assuming that the physical simulation is set up and an order parameter λ is available to characterize the state of the system. First, we specify our states A and B and define the borders λ_A and λ_B. We enable ghost runs and automatic interface placement in the configuration, and connect a couple of clients to the server. Then, our simulation proceeds as follows:

1. The escape flux Φ is calculated in parallel by the clients according to Sec. 3.2.1 with a fixed number of calculation time t_p per client.

2. As the next interface position is unknown (we only specified λ_A and λ_B) the server starts exploring runs according to the configuration and if a user-defined threshold of points is collected on the current calculated interface.

3. If the new interface position is determined and the last interface has still calculations which are left open, the server starts ghost runs to the new interface if calculation resources are available.

4. As soon as the calculations on the last interface are completed, real runs are started and ghost runs are transferred according to Fig. 4.2.

5. These steps are repeated until the new interface is $\lambda_{i+1} = \lambda_B$. Then, the last transition to λ_{i+1} is performed to obtain all p_i, and finally P_B and hence $k_{AB} = \Phi P_B$ can be calculated.

In the scheme above, clients have to wait very rarely, e.g. if not enough points on the last interface are available to start an exploring run or a ghost run. This would only happen, if a large number of clients are connected and only a low number of points is collected on the interfaces and the threshold for starting the latter mentioned runs is set very high.

Having discussed an efficient scheme for the server side, we will now have look at the client side of FRESHS.

4.2.2 The clients: massively parallel calculations

The task of the FRESHS client is to communicate with the server and to steer the simulation of the physics. Fig. 4.4 gives an overview of the internal structure of a client. When connected to the server, the communication part of the client receives a certain message which is translated to a job request. Then, a specific *harness script* which launches the simulation with the appropriate parameters is called. The harness script can be any type of an executable file like a bash or python script or even compiled

[7]The FRESHS configuration possibilities allow also to perform the very basic simulation like it is described in the literature for the serial case by turning off the extensions and by connecting a single client to the server.

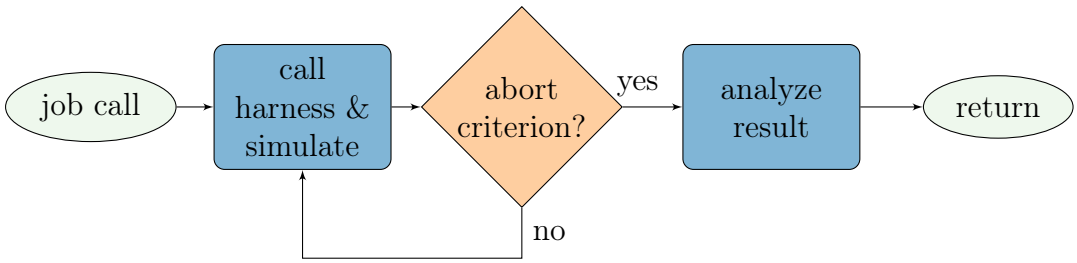

Figure 4.4: Client scheme: If the client receives a job request from the server, the harness script is called with the appropriate parameters and launches the simulation. During simulation, the order parameter is monitored and if the abort criterion (initially given by the server) is reached, the simulation is ended and analyzed. Thereby, the abort criterion can be checked by the harness script or by the client [98]. In FFS, a configuration point is stored if the next interface is reached, otherwise only the metadata like calculation steps and the outcome of the trajectory is reported to the server.

code. Examples for different simulation tools like GROMACS, LAMMPS or ESPResSo are available [102]. The simulation aborts according to the sampling method's abort criteria. If the simulation tool doesn't support order parameter checking, this can be also performed by the client, but this results in a lower performance in general, e.g. when the startup time of the simulation is long. However, as many large simulation packages have this peculiarity, we developed a scheme to maximize the performance in such cases. For details refer to [98]. In our case, the simulation is script language driven, which means that we are able to check the abort criterion between a desired number m of integration steps without restarting the whole simulation program. When our FFS simulation is ended due to the abort criterion, a configuration point is stored if the trajectory was successful, or otherwise only the metadata like the performed steps and the outcome of the trajectory is reported to the server.

Many of these clients can be connected to the server to achieve parallelization. The simulation of the clients itself can be also carried out in parallel, if the architecture of the underlying hardware allows for that. E.g. one could use 1 node with 8 CPU cores for a client, and connect 100 such clients to the server to obtain a calculation power of 800 CPUs.

Now, we will have a look at the quality of the statistics of an FFS simulation and at the backtracking of successful transition pathways.

4.3 Analysis of the statistics

In our simulations, we are not only interested in calculating the transition rate k_{AB} for different physical conditions, but also in the transition pathways, whereby we are

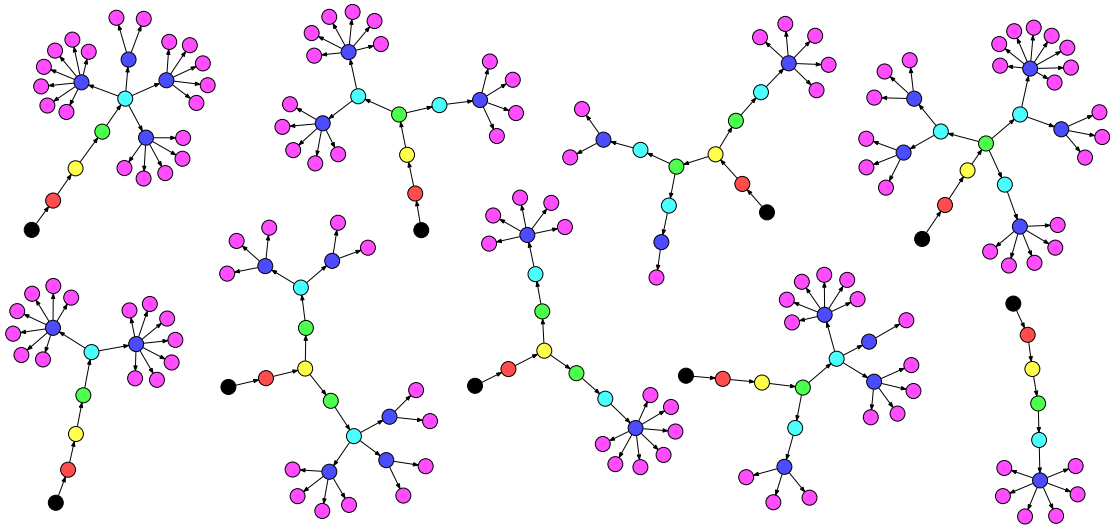

Figure 4.5: Successful transition pathways obtained via backtracking from interface λ_B. The arrows start at λ_A (black circles) and point towards λ_B, with the color coding per interface. The flower-like look arises because of $p_i \approx 1$ for the last transition. The more pathways are obtained, the better is the sampling of the statistics.

able to investigate the crystal growth.

For FFS, it is important to have multiple transition paths with many branches. If all successful runs would cross only a single configuration point on an interface λ_i, a bad sampling statistics would be obtained. If we have for example one configuration point on interface λ_i which has a higher *escape velocity* $\dot{\lambda}$ towards B, this point could be preferentially sampled, e.g. if every run launched from this point reaches the next interface λ_{i+1}. This is especially important for λ_A, because the quality of the sampling on this interface determines the chances of different configuration points being chosen to advance in positive B direction. The number of different available trajectories on an interface λ_i can be obtained via backtracking the runs till λ_A, starting at the configuration points on λ_i. Fig. 4.5 shows the result of such a backtracking. In this case, we backtracked the runs of a crystallization problem starting from interface λ_B. The flower-like look arises, because $p_i \approx 1$ for the last transition. In this case, we obtain a quite large number of different successful trajectories from λ_A to λ_B, like it is desired.

In general, the number of different origin points $N_{0,i}$ on λ_A of the backtracked runs from interface λ_i is the same as the number of different successful trajectories for the simulation. This number of different origin points can be tracked during simulation to determine the decay of the number of different origin points which we define as the fraction of the different points $N_{0,i}$ obtained via backtracking from interface i, related

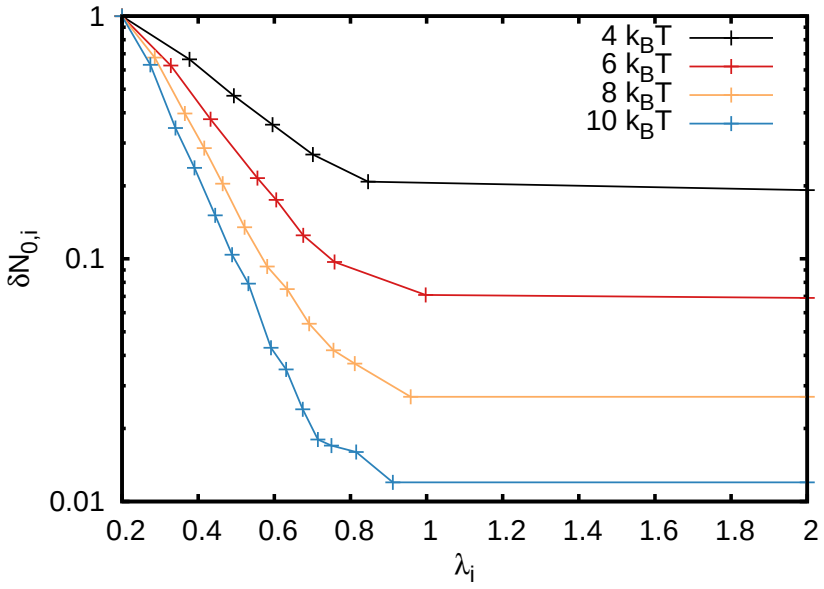

Figure 4.6: Decay $\delta N_{0,i}$ (Eq. (4.1)) of different origin points on λ_A when backtracking runs from different interfaces λ_i for the 1D particle example with $N_0 = 1000$ at $\lambda_A = 0.2$ and the maximum of the barrier at $\lambda = 1.0$ (see also Sec. 3.1.1).

to the number of initial configuration points N_0 on λ_A,

$$\delta N_{0,i} = \frac{N_{0,i}}{N_0} \tag{4.1}$$

Fig. 4.6 shows this decay with the help of the simulation of the 1D particle for a starting number of $N_0 = 1000$ configuration points on λ_A.

Note, that the decay depends on the simulation and the interface spacing, but the general message is, that we should use more points on λ_A and a higher sampling at larger barrier heights, because we loose origin points and hence different successful trajectories during the advancement to B.

FRESHS has a built-in feature to monitor this number on-line and per interface λ_i already during simulation and to adjust the number of configuration points on an interface if the presetting can't be fulfilled. With this, many successful physical transition pathways can be obtained, not only for the purpose of a good sampling but also for the investigation of multiple transition mechanisms.

4.4 Calculating stationary distributions

In general, according to section 2.5.3 the stationary distribution $\rho(q)$ of an order parameter q can be calculated in an FFS simulation by splitting up the calculation of

the distribution in a forward simulated part and a backward simulated part (see also Eq. (2.73)).

Concerning our simulations this means that the crystal of macromolecules must not only be built up in a forward simulation but also be dissolved again in a backward simulation. Thereby, the automatic, optimized interface placement for FFS simulations which was discussed in section 3.3 facilitates such an intention tremendously, because the simulation must simply be set up in A or in B, the rest of the reaction scheme (forward and backward) is populated automatically [94].

In FRESHS, the backward reaction is realized such that we negate the order parameter. Thus, we are able to use the same order parameter (in our case the cluster size) but the reaction is driven in the other direction. The great advantage of this scheme is, that we do not need to change the logic in our routines, e.g. all comparators can be used like they are. This backward reaction scheme via negation is already implemented in FRESHS and can be activated via the configuration options. Thereby, the configurations in B can be loaded from an existing database of a forward simulation run, which is important if the system states in B can't be set up ad hoc like in the crystallization case, where we can't simply set up the ready crystal cluster, e.g. because of its unknown shape.

To calculate the stationary distributions, all order parameter values of all trajectories are required. FRESHS provides a field in the database to store custom data, which we can directly use for this purpose. The advantage of this is, that we are able to store the order parameter values directly assigned to the runs and that we can use database functions to extract all values which belong to an interface[8], which is required in this case because the histogram per interface is then weighted with the corresponding transition probabilities.

Now, we will show the calculation of the stationary distributions on a fundamental simulation problem.

Stationary distributions of a dimer molecule

To demonstrate the calculation of an energy profile via FFS simulations using stationary distribution we simulated a 1D particle in a double well potential using FRESHS. For the simulations[9], we use a velocity verlet integrator and a Langevin thermostat like described in Sec. 2.2 and set $\gamma = 1$, $k_B = 1$, $T = 1$, and $\tau = 0.01$. The potential from which the forces are derived is given by

$$U(x) = x^4 - 7.5x^2 + 14.0625. \tag{4.2}$$

The particle is set up at the left-hand side of the barrier in region A with $x < -1.5$. We are interested in pushing the particle over the barrier to state B using FFS.

[8] An example would be 'SELECT customdata FROM configpoints WHERE lambda = 1'

[9] The harness script is available in the FRESHS package (`http://www.freshs.org`)

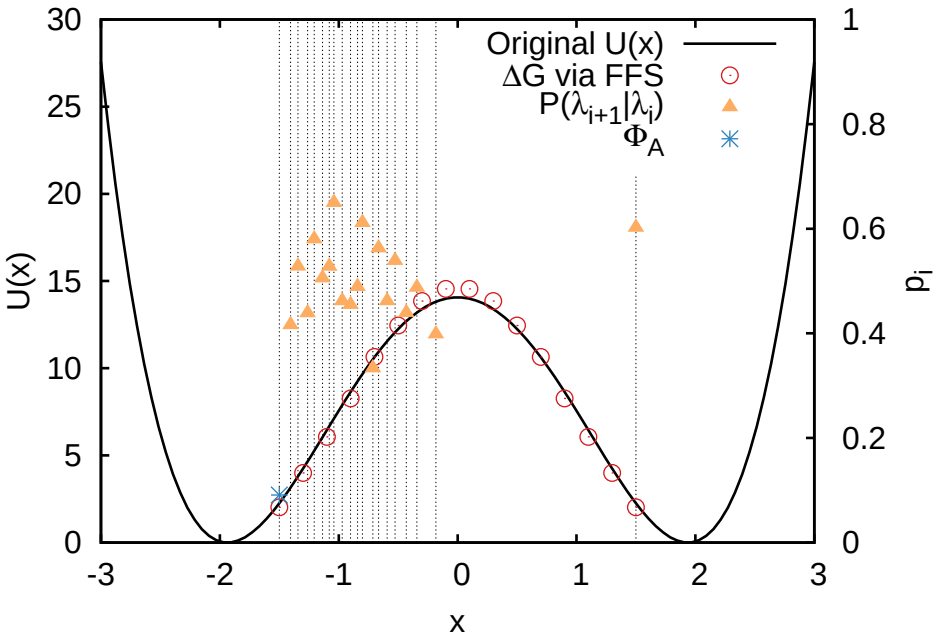

Figure 4.7: Computing the energy landscape with FFS using the stationary distribution theory: The solid line is the underlying 1D potential of the simulation (Eq. (4.2)). The blue datapoint represents the escape flux $\Phi_A \approx 0.09\tau^{-1}$ and the orange triangles the forward probabilities $P(\lambda_{i+1}|\lambda_i)$. The dashed vertical lines show the interface positions λ_i. Since the problem is symmetric, the data of the forward run is inversed and used for the backward run leading to the reproduced energy profile ΔG from Eq. (2.81) (red circles).

Therefore, we set $\lambda_A = -1.5$ and $\lambda_B = 1.5$. We use the automatic optimized interface placement [94] with exploring scouts and a target probability of $p = 0.5$ to advance from A to B automatically and efficiently.

An overview of the results of the simulations is given in Fig. 4.7 and table 4.1.

Since our potential and simulation problem are symmetric, we only need to perform the simulation in one direction (A to B) and can directly use these results for the other direction (B to A) by mirroring the order parameter range and the results at $x = 0$. Therefore, in Fig. 4.7 we only show the forward simulation FFS results with the automatically determined interface set and transition probabilities, which are distributed around our target probability $p = 0.5$. The overall transition rate was determined to $k_{AB} = 1.3 \times 10^{-7}\tau^{-1}$ for this transition. Table 4.2 gives an overview of the computational details of the simulation.

Using the calculations of Sec. 2.5.3, the energy profile ΔG was calculated and matches nicely the given potential shape of Eq. (4.2) (red circles in Fig. 4.7). Note, that this is already a rare event with a relatively high barrier, to further improve the statistics and thus the shape of ΔG one has to collect a lot of points in the FFS

i	0	1	2	3	4	5	6
λ_i	-1.50	-1.41	-1.34	-1.26	-1.21	-1.14	-1.08
p_i	0.09^*	0.42	0.53	0.44	0.58	0.51	0.53
i	7	8	9	10	11	12	13
λ_i	-1.04	-0.97	-0.90	-0.84	-0.80	-0.71	-0.67
p_i	0.65	0.46	0.45	0.49	0.61	0.33	0.56
i	14	15	16	17	18	19	20
λ_i	-0.59	-0.53	-0.43	-0.34	-0.18	1.47	1.50
p_i	0.46	0.54	0.44	0.49	0.40	0.60	1.00

Table 4.1: Interface locations λ_i and transition probabilities p_i of the 1D particle in a double well potential, identified by the automatic, optimized interface placement. The quantity with an asterisk (*) denotes the escape flux.

$k_{AB}[\tau^{-1}]$	$\Phi[\tau^{-1}]$	P_B	$\mathcal{C}$	$\mathcal{V}$	$\mathcal{E}$
1.3×10^{-7}	9.1×10^{-2}	1.5×10^{-6}	1.2×10^{7}	1.7×10^{1}	4.9×10^{-9}

Table 4.2: Computational details of the 1D particle in a double well potential. The error $\mathcal{V}$ in the rate k_{AB} is very low because of the automatic, optimized interface placement, leading to a good efficiency $\mathcal{E}$. The computational cost $\mathcal{C}$ is measured in simulation steps.

simulations.

The branching trees of the trajectories can be extracted by backtracking the successful runs from λ_B in the database. In an FFS simulation, many of these trees are obtained, starting at different origin points. Fig. 4.8 shows such a tree for the 20 interfaces of this simulation, exemplarily.

During the transition from λ_A to λ_B, the tree branches several times. Here, we only show the successful branches which reach the last interface λ_B, but there are also many branches which do not reach the last interface and which are ended in between. The FRESHS package provides scripts to analyze this branching behavior in detail, which can be used to visualize the dynamics of the simulation.

The demonstration in this section shows, how FRESHS with the built-in datahandling can be used for versatile analysis, e.g. to calculate the free energy landscape from custom data and to analyze transition rates. We will apply the same analysis of ΔG to our crystallization of charged macromolecules simulation in chapter 6.

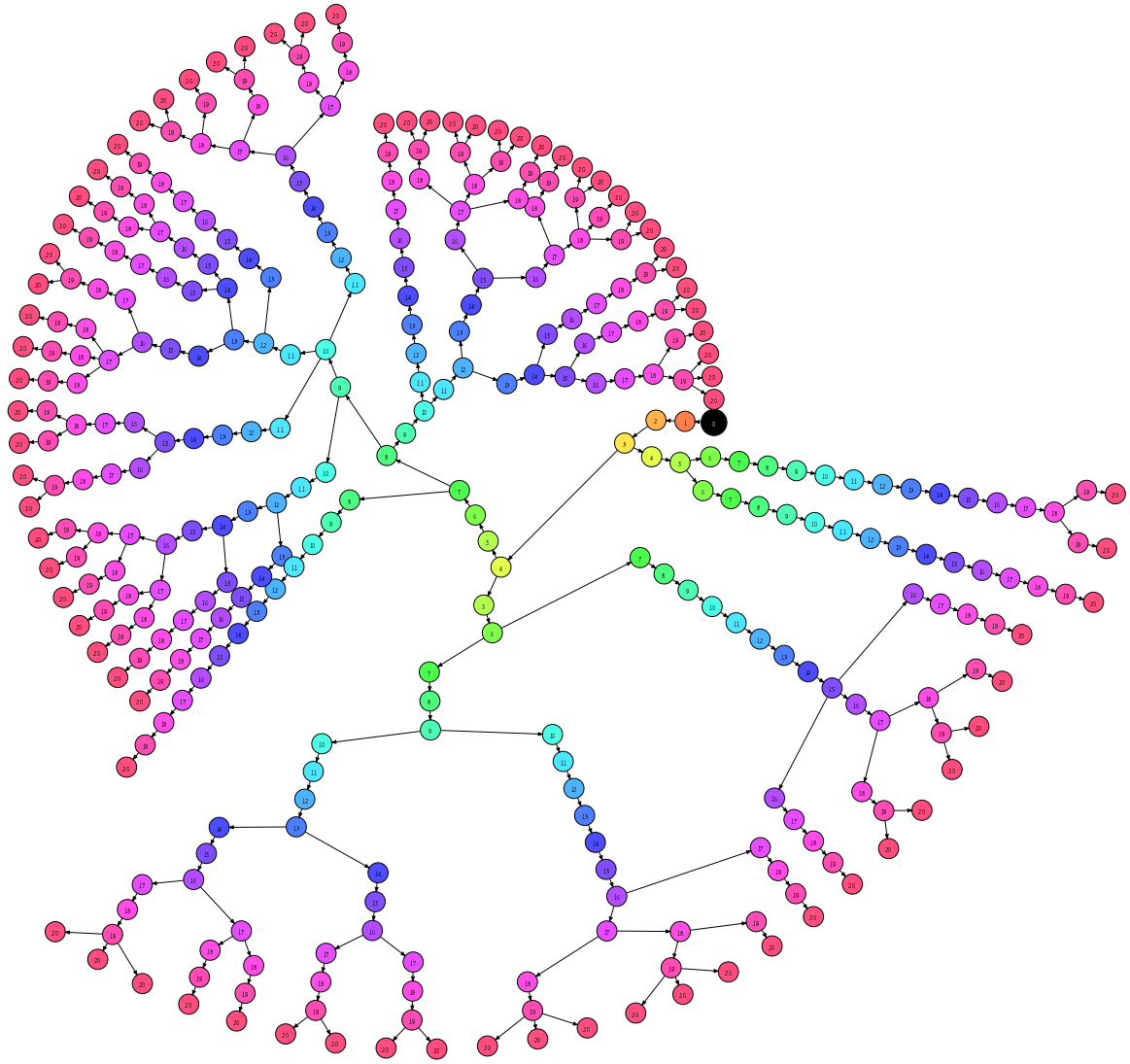

Figure 4.8: Successful branching tree of the 1D particle in a double well potential, exemplarily. In the FFS simulation, multiple trees of this kind are obtained. The color coding is per interface, the arrows point is positive interface direction and the starting point is the larger black circle, which is a configuration point on λ_A.

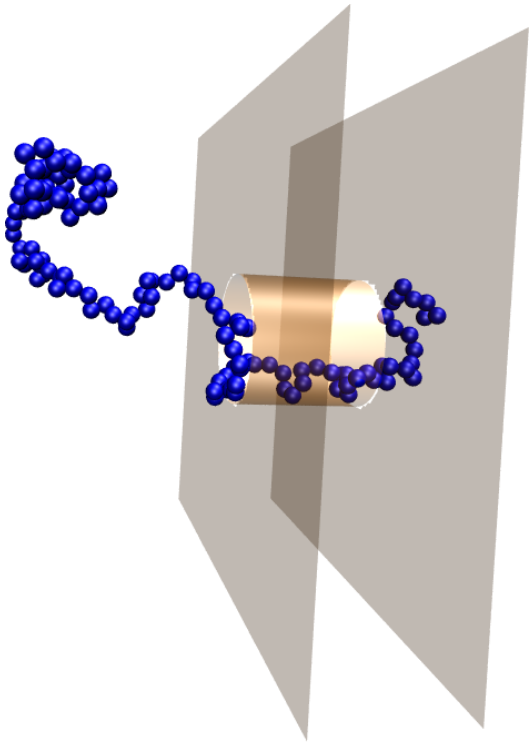

Figure 4.9: Snapshot of the rare event sampling simulation of a polymer which is translocating through a nanopore. The polymer was set up at the entrance of the nanopore and is pushed through the nanopore using the framework FRESHS with FFS, ghost runs, and automatic, optimized interface placement in parallel.

4.5 Translocation of a polymer through a nanopore

In this section, we use the software package ESPResSo [17] together with the Forward Flux Sampling method, ghost runs and automatic, optimized interface placement [94] to push a polymer through a nanopore (Fig. 4.9). This process is also a rare event due to the free energy barrier towards the translocation and is important for many process in nature or technical applications, e.g. when sequencing DNA [105].

This use case is not part of our article [98]. For other examples of the usage of FRESHS, e.g. together with the simulation tools GROMACS, LAMMPS or with the sampling method SPRES for dynamic systems, please refer to refs. [98].

4.5.1 Order parameter

As order parameter, we use the z coordinate in direction of the pore axis of the center of mass of the polymer chain,

$$\lambda = \frac{1}{N} \sum_{i=0}^{N-1} z_i, \tag{4.3}$$

where N is the number of monomers in the polymer chain. Note, that this is maybe not the best order parameter, but an intuitive one which increases monotonously when the polymer translocates through the nanopore.

4.5.2 Simulation and rare event sampling details

For the initial state A, we set up a polymer with the first bead located at the entry of the pore. The rest of the polymer chain is placed at random, using the capabilities of ESPResSo to set up random polymers. Then, the order parameter of this state is verified. If the order parameter value is in the allowed range for state A, we begin the initial MD run in A to calculate the escape flux Φ. If the value is not in A, we delete the polymer and try a new one.

For the simulation setup, we use a 3D box with dimensions $30 \times 30 \times 60$ and periodic boundary conditions. The center of the pore is located in the middle of the box, with a length of $l = 10.0$ in z-direction. The radius r of the pore can be varied, for smaller pores the translocation event becomes very rare, because there is a huge energy barrier towards pushing the polymer into the pore. Here, we simulate for different r and different monomer numbers n_p of the polymer to show the dependence of these quantities on the energy barrier and transition rates.

The entry of the pore is located at $\lambda = 25$ in our case and the center of the pore is at $\lambda_C = 30$. For the border of the initial domain A we choose $\lambda_A = 17$, which is at the beginning of the ascent of the free energy curve towards the entry of the pore. We set $\lambda_B = 43$, which is the mirrored value to λ_A at λ_C. Note, that we do not consider the x and y directions in our reaction coordinate, which can lead to situations where the polymer is located away from the center of the pore. However, to make the translocation event more probable, we set up the polymer at the entrance of the pore. In addition, if the polymer diffuses too much away from the pore, we have to interrupt the simulation before the center of mass is at the other side of the simulation box because of the periodic boundary conditions, which is a simulation specific boundary of our state A. If this happens we initialize our system again in A. This can be seen as fixing the volume of the box, which is necessary in this case to obtain a translocation rate. In an infinitely large box, the rate would be zero.

The polymer uses so-called FENE interactions for the bonds. The other interactions (e.g. polymer-pore) are set to Weeks-Chandler-Andersen (WCA) interactions with $\sigma = 1$ and $\epsilon = 1$, like they are used for the excluded volume of our macromolecules.

n_p	r	$k_{AB}[\tau^{-1}]$	n_p	r	$k_{AB}[\tau^{-1}]$
64	3	5.7×10^{-11}	32	5	1.1×10^{-6}
64	5	2.3×10^{-7}	64	5	2.3×10^{-7}
64	7	2.4×10^{-6}	96	5	6.3×10^{-8}
64	9	9.6×10^{-6}	128	5	2.4×10^{-8}

Table 4.3: Overview of the results of the translocation experiments. On the left-hand side the pore radius r was varied, and on the right-hand side the polymer length n_p. The transition rate k_{AB} becomes lower for smaller pore radii and longer polymer chains.

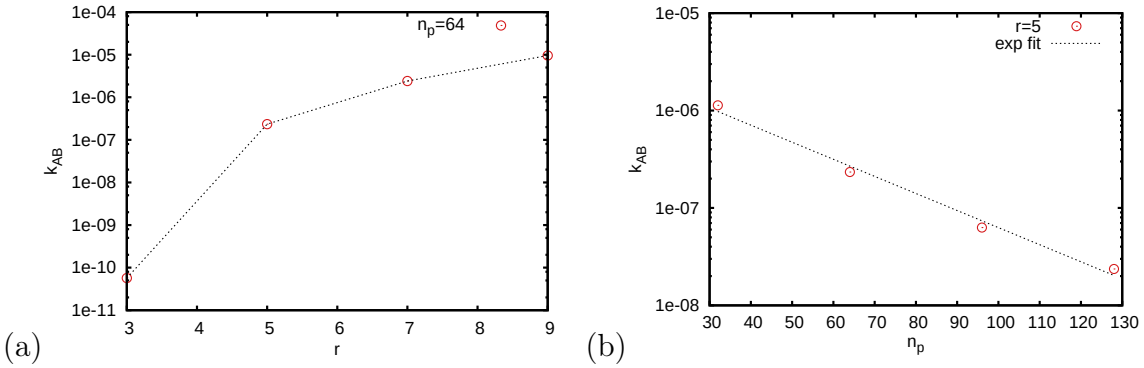

Figure 4.10: Transition rates for the translocation simulations: (a) Varying the pore radius r for a polymer of the length $n_p = 64$, (b) Varying the polymer length n_p for a fixed pore radius $r = 5$.

Furthermore, we simulate at $T = 1$, $k_B = 1$ and with a timestep of $\tau = 0.001$.

4.5.3 Results - transition rates

Table 4.3 gives an overview of translocation simulations with the resulting transition rate k_{AB} for different pore radii r and different polymer lengths n_p. These values are visualized in Fig. 4.10. Fig. 4.10(a) shows the dependence of the transition rate k_{AB} on the pore radius r. The transition rate of the polymer is much lower for small pore radii, because pushing the polymer into the pore costs more free energy in this case. The same is true for longer polymer lengths n_p, as shown in Fig. 4.10(b), where we find an exponential dependence of the rate on the polymer length $k_{AB} \propto \exp(-\Gamma_b n_p)$, where Γ_b is related to the cost of free energy for a bead of the polymer entering the pore. This is consistent with theories from literature, e.g. [106].

Note, that the error of the simulations increases with a smaller pore diameter, because we obtain less successful traces with the same number of collected points in FFS. The error of the rate at $r = 3$ is approximately two orders of magnitude and at

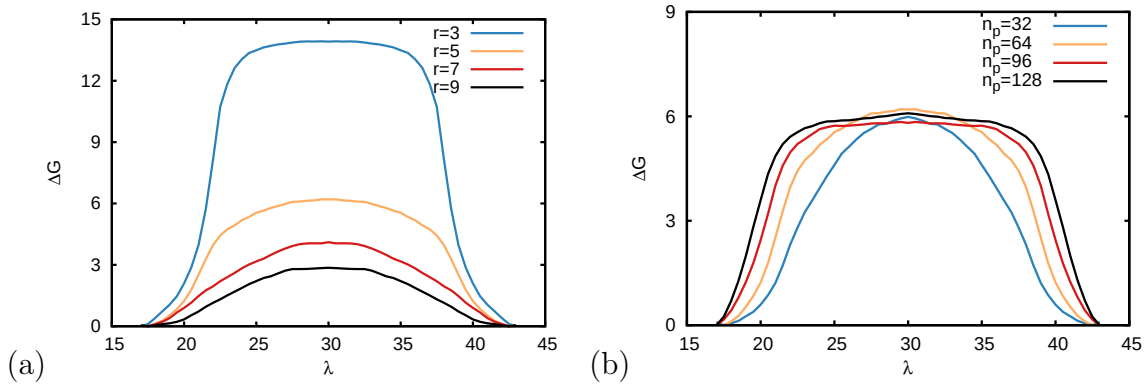

Figure 4.11: Free energy landscapes $\Delta G(\lambda)$ for the translocation simulations, symmetry-mapped from $\lambda_A = 17$ to $\lambda_B = 43$. (a) $\Delta G(\lambda)$ for different pore radii r, simulated for a fixed length $n_p = 64$ and (b) for different polymer lengths n_p at a fixed radius $r = 5$. The free energy landscape has a plateau domain along the pore, as well as a steep ascent at the entrance. Thereby, the ascent is steeper for longer polymer chains and for smaller pore radii. Note, that for the absolute barrier heights a brute-force sampling term must be added to these curves (not shown here).

$r = 9$ one order of magnitude, determined by repeated simulations.

4.5.4 Results - free energy landscapes

With the techniques discussed in Sec. 4.4 we are also able to calculate the energy landscape $\Delta G(\lambda)$. Because of the symmetry of the problem, we can also use the mirrored data for the backward FFS run, where we mirror at $\lambda = 30$ which means that our simulations cover the range $\lambda_A = 17$ to $\lambda_B = 43$. The results of these calculations are shown in Fig. 4.11. Fig. 4.11(a) shows the shape of $\Delta G(\lambda)$ for varying the radius of the pore while fixing the polymer length, and Fig. 4.11(b) for varying the polymer length and fixing the radius. As expected for these kind of experiments, we obtain a shape of the energy landscape which is steep at the entrance of the pore and has a flat part along the pore. This becomes more significant for smaller pore radii, as shown in Fig. 4.11(a). Furthermore, the ascent of the barrier is much steeper for longer polymers, which means that it is more difficult for them to enter the pore, which can be seen in Fig. 4.11(b). Note, that for the absolute barrier heights, a brute-force term must be fitted to these curves, which is not shown in here.

4.6 Summary

With the help of the powerful framework created during this work and presented in this chapter, computationally expensive rare event sampling simulations can be performed optimized, efficient and in a parallel way on high performance computing and heterogeneous computing resources. Thereby, the following optimizations and extensions have been made beyond the state of the art (chapter 2):

- The rare event sampling method is now applicable in parallel and in combination with an itself parallel simulation.
- Simulation tools and rare event sampling techniques can be exchanged in a flexible way, allowing for a general comparison of tools.
- Waiting times in the *asynchronous* parallelization have been bridged successfully using ghost runs.
- The efficiency of FFS was increased tremendously by the automatic, optimized interface placement, which is also implemented in our parallel framework and simplifies a simulation such, that only the states A and B must be defined in terms of an order parameter λ.
- Simulating backwards or starting a new simulation from the final state of a previously calculated transition $A \rightarrow B$ is possible by using the database of the previous run. Thereby, simulating backwards is possible by internal negation of the order parameter.
- The quality of the statistical sampling can be monitored and tuned on the fly, as the simulation progresses and according to simulation-specific parameters.
- The ghost runs can be used to increase the number of sampling points in the complete FFS scheme after the simulation has already been performed.
- An intelligent data storage scheme using databases has been implemented allowing to interrupt, adjust and resume simulations from different stages.
- Versatile analysis tools which are based on the intelligent data storage can be applied, e.g. to analyze the tree of successful pathways, histograms of the run lengths, stationary distributions and hence free energy landscapes.

Without the above mentioned achievements the simulation of charged macromolecules wouldn't have been possible under the desired conditions which are interesting for us. Before advancing to the rare event simulations of the charged macromolecules, we compare the applicability of the simulation model to real experimental data in the next chapter.

5 Comparison of the Yukawa model to an experimental colloidal system

In this chapter we compare results of experiments with colloidal particles to numerical simulations of a comparable system. The colloidal experiments were performed by the Bechinger group[1] at the second institute of physics PI2, University of Stuttgart.

The aim is to determine the pair interaction of the colloids by simulations and to verify the applicability of the screened Yukawa interaction model.

5.1 Introduction

In experimental physics colloidal systems, e.g. consisting of spheres made of Polystyrene in solution, are used to study different effects like entropic forces, many-body interactions or hydrodynamic coupling [107]. In such systems, the radial distribution function (RDF) is often used to characterize the spatial organization of the particles. The advantage of using an RDF is that the RDF is accessible in both, experiments and simulations, because it is calculated from the coordinates of the particles in the system. In turn, these coordinates depend on the underlying pair interaction of the particles in a unique way [56].

In experimental systems, these interactions are difficult to characterize, because they depend on the particles' local surrounding which is influenced e.g. by the ion density, thermal fluctuations, and impurities. Theoretically, the pair interaction can be described by a pair interaction potential, which models an effective interaction.

The task in this chapter is now to find the pair interaction which leads to the distribution of particle coordinates in the simulation to obtain the same RDF as in the experiments. To this aim, we use two approaches: (i) We simulate a system with the same density as the experimental one using the Yukawa model (see also equation (2.30)) and try to find the screening length κ and the contact value A such, that we obtain the correct RDF, and (ii) we reconstruct the pair interaction potential directly from the RDF using the Inverse Boltzmann method (Sec. 2.2.4) in combination with a tabulated potential. All the simulations are MD simulations and performed using the software package ESPResSo [17].

[1] Zaidouny et al.

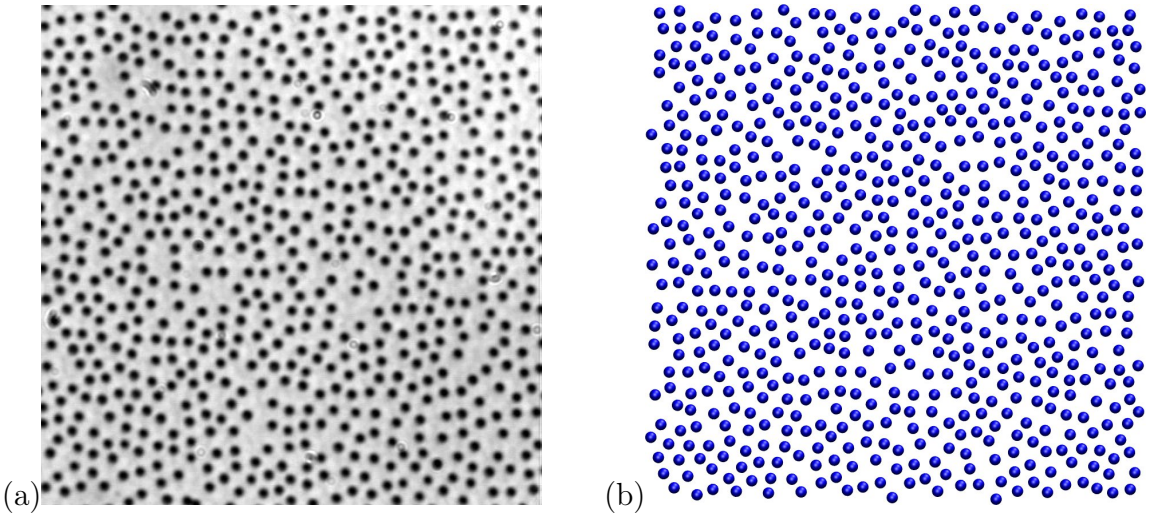

Figure 5.1: (a) Detail of the 2D experimental Polystyrene colloidal system (photomicrograph, Zaidouny et al.). The colloids have a diameter of $\sigma = 3.9\mu m$. (b) Snapshot of the 2D simulation of colloidal particles for comparison with the experimental data. The comparison is performed via matching the RDF of both systems by tuning the pair interaction.

5.2 Experimental details

Fig. 5.1(a) shows a microscopical image of the 2D experimental system with Polystyrene colloids in water. The colloids in the system have a diameter of $\sigma = 3.9\mu m$. The number of particles plays a minor role, the important quantity is the density ρ of the system.

For the system shown in Fig. 5.1(a), the radial distribution function can be calculated from the N particle coordinates in volume V by [108]

$$g(r) = \frac{V}{N} \left\langle \sum_{i=1}^{N} \delta(\mathbf{r} - \mathbf{r_i}) \times \sum_{j=1}^{N} \delta(\mathbf{r} - \mathbf{r_j}) \right\rangle \tag{5.1}$$

with Dirac's δ-function $\delta(x)$. In practice, several frames ($\mathcal{O}(1000)$) are recorded with the camera of the optical microscope. Then, the coordinates of the colloids are extracted from the frames and can be used to sample the corresponding RDF.

The system was prepared for different densities ρ. The lower densities are expected to be easier to compare to the Yukawa model, because at high densities the particles are located at closer distances and many-body interactions can play a role [107]. Thereby, the investigated densities are in the range $0.00215\mu m^{-2} \leq \rho \leq 0.02\mu m^{-2}$.

5.3 Simulation details

Fig. 5.1(b) shows a visualization of the simulated 2D system. We use a Langevin thermostat and periodic boundary conditions to realize a sufficiently large NVT ensemble. In the simulations, the particle coordinates are already known and can be used directly to sample the RDF in a time evolution of the system, which is comparable to the method in the experimental system by recording several frames with a microscope in a time series.

Since the experimental RDFs suffer from finite size effects and do not end with their tails at values of 1, we correct the experimental RDFs by dividing the whole RDF by a factor which was determined by the average value of the tail of the particular RDF for better convergence of the simulations.

For the simulations of the colloids we use the Yukawa pair interaction potential in the following notation:

$$U_{\text{Yukawa}}(r) = Ak_BT\frac{\exp\left(-\kappa r\right)}{r} \tag{5.2}$$

with the pre-factor A, which is given according to DLVO theory[2] as

$$A = l_BZ^2\frac{\exp(\kappa\sigma)}{(1+0.5\kappa\sigma)^2}, \tag{5.3}$$

where Z is the charge and σ the diameter of a colloid.

To determine the parameters A and κ in the pair interaction potential (Eq. (5.2)) we use two approaches, the first is based on tuning the Yukawa interaction itself and the second is the Inverse Boltzmann approach for the direct reproduction of the potential from the RDF without pre-knowledge like described in Sec. 2.2.4.

5.4 Results

In this section we present the results concerning the reconstruction of the pair interaction potential from the experimental datasets. The first part is about tuning the Yukawa interaction and the second part about the reproduction of the potential by the Inverse Boltzmann method.

5.4.1 Optimizing the Yukawa potential

Here, we use the Yukawa model of Eq. (5.2) for our simulations to compare with the experimental data.

[2]Derjaguin, Landau, Verwey and Overbeek

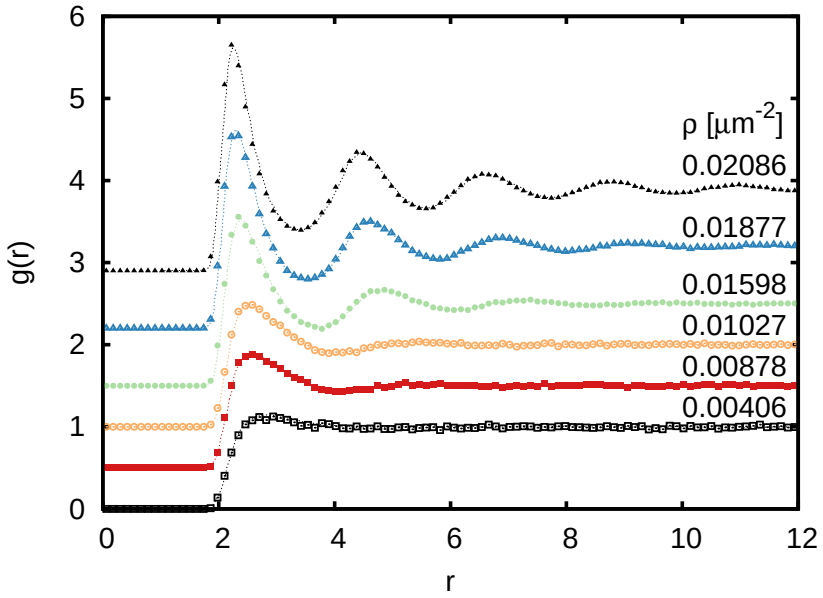

Figure 5.2: Reproduction of the data of [107] for $A = 750000\mu m$, $\kappa = 1.83\mu m^{-1}$ and different densities ρ. For better visibility the RDFs are shifted in y-direction. The dashed lines depict the digitized data from [107], the datapoints are the reproduced data. The RDFs can be reproduced by our simulations in great detail. Note, that the particle diameter is different in this work.

$\rho[\mu m^{-2}]$	$a[\mu m]$	$U(r=a)[k_BT]$
0.00406	2.83	0.015
0.01027	2.57	0.071
0.01877	2.29	0.370
0.02086	2.24	0.498

Table 5.1: Values of the simulated potential at the peak location $r = a$ of the RDFs for the different densities ρ.

Comparison to previous data

To verify the correctness of the implementation of the simulation we reproduce the experimental data of Ref. [107]. To this aim, we set $A = 750000\mu m$ and $\kappa = 1.83\mu m^{-1}$ in Eq. (5.2), which correspond to the values of Ref. [107] with $\sigma = 3\mu m$. Fig. 5.2 shows the results of our simulations for 6 different densities ρ. The data of [107] could be reproduced in great detail for all densities with our simulations.

As a reference, table 5.1 shows the values of the potentials $U(a)$ at the position of the first peak $r = a$ of the RDF.

Sensitivity of the parameters

To find the values of A and κ which lead to a simulated RDF with the minimal distance to the experimental one we varied these values in a step-wise fashion, where we started with guessed values first. As a measure of quality, we use the L^2 distance d of the experimental RDF and the simulated RDF which we define as

$$d = \int_{r_t}^{\infty} \frac{1}{r}(g_{\text{exp}}(r) - g_{\text{simu}}(r))^2 dr, \tag{5.4}$$

with the value r_t where the RDF begins to take off from 0. Thereby, we weight the distance with r^{-1} because the first part of the RDF is more significant for the pair potential than the rear part, as it is related to the main part of the interaction of two particle pairs. Then, we iterate A and κ to minimize the distance function Eq. (5.4) in a loop. This should lead to values of the Yukawa potential which fit the experimental RDF best.

However, this method is complex because the guessing part is based on a trial and error approach, and if the method leads to a deviating RDF it is unclear if the wrong parameterset of A and κ is chosen or if the system can't be matched with a Yukawa model.

Matching an experimental RDF

In this section we use the tuning method from above to reproduce an experimental RDF created from $N = 612$ particles at a density of $\rho = 0.011125\mu m^{-2}$. Fig. 5.3 shows the result of the minimization procedure with Eq. (5.4) and the sensitivity of the parameters A and κ. For the given system we found with that procedure a $A = 10^{25}\mu m$ and an inverse screening length of $\kappa = 7.35\mu m^{-1}$. The resulting *Debye screening length* is then $l_D = \kappa^{-1} \approx 136nm$.

Because of the high sensitivity of the RDFs to the parameters and the fact, that different combinations of A and κ can lead to a similar quality of RDFs when using the distance d (Eq. (5.4)) due to $\kappa \propto \log A$ for similar contact values at the main distance of the particles, we use the Inverse Boltzmann for reproducing the potential in the following sections.

5.4.2 Inverse Boltzmann

Here, we use the Inverse Boltzmann method from Sec. 2.2.4 to reproduce the pair potential from RDFs at different densities.

Low densities

To exclude many-body interactions the experimental systems have been prepared at the densities $\rho = 0.00215\mu m^{-2}$ and $\rho = 0.005\mu m^{-2}$. Here, we use the Inverse Boltz-

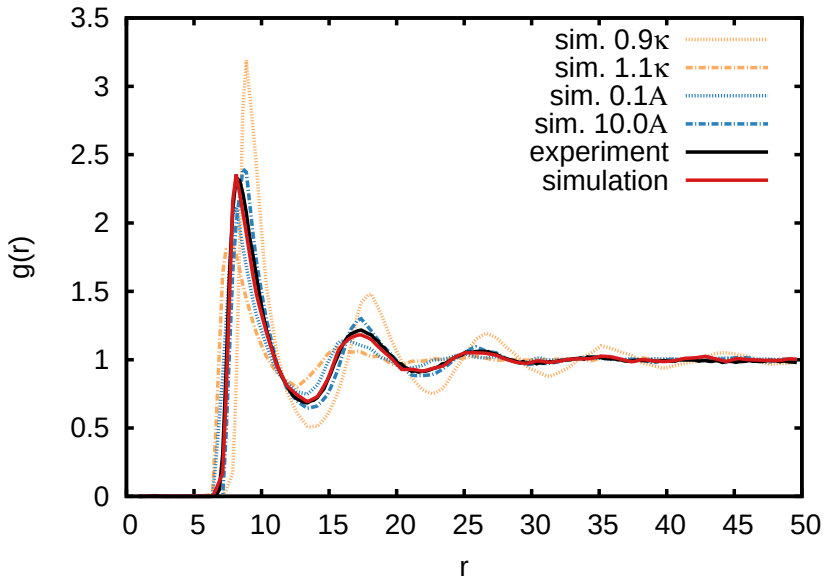

Figure 5.3: Radial distribution function $g(r)$ for the experimental system and the simulation with $A = 10^{25}\mu m$ and $\kappa = 7.35\mu m^{-1}$. The dashed curves show the behavior of the RDF for $0.9 \times \kappa$ and $1.1 \times \kappa$ as well as for $0.1 \times A$ and $10.0 \times A$. The resulting screening length is $\kappa^{-1} \approx 136nm$ in this system.

$\rho[\mu m^{-2}]$	$A[\mu m]$	$\kappa[\mu m^{-1}]$
0.005	2.2×10^9	2.15
0.00215	11.3×10^9	2.70

Table 5.2: Inverse Boltzmann Yukawa fitting results. The values for κ result in a screening length of $400nm - 500nm$, similar to the results in Ref. [107]. The prefactors differ due to the fact that the RDFs start to take off at different distances ($7\mu m$ vs. $8\mu m$).

mann method to determine the parameters. For these low densities we obtain potentials that fit well to a Yukawa-like shape for both densities which are depicted with a corresponding RDF in Fig. 5.4.

Note, that at the lowest density $\rho = 0.00215\mu m^{-2}$ the results are more noisy because at this low density the simulations become computationally more expensive. It takes around 50000 simulation steps for a particle to diffusive over the mean particle distance which is important for the sampling of the RDF.

The results for A and κ for both low densities are given in table 5.2. The screening length κ^{-1} is in the range of $400nm - 500nm$ which is comparable to the data from [107]. However, the prefactors differ due to the fact that the RDFs start to take off from 0 at different distances ($7\mu m$ vs. $8\mu m$) and this must be compensated by the prefactor (Eq. (5.3)), which depends on the salt concentration Z. The factor $\exp(\kappa\sigma)$ in Eq. (5.3)

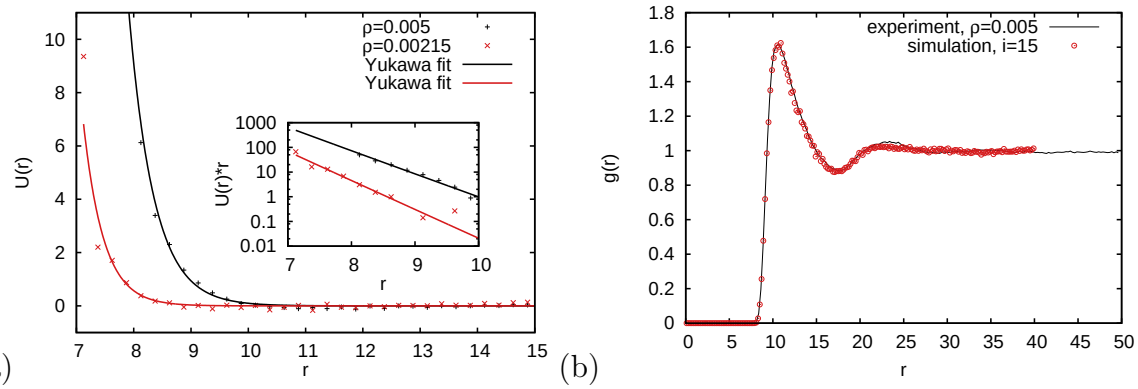

Figure 5.4: (a) Datapoints of the potentials for the two densities $\rho = 0.005\mu m^{-2}$ and $\rho = 0.00215\mu m^{-2}$ and corresponding Yukawa fits (the values of the fit are given in table 5.2). The inset shows the first part of the potentials as $U(r) \times r$ plot. (b) RDF for $\rho = 0.005\mu m^{-2}$ obtained with the tabulated potential (datapoints) from (a). The experimental RDF can be reproduced nicely and the obtained potential has a Yukawa-like shape.

is quite large, because the screening length κ is small compared to the colloid diameter σ. For the pre-factor in Eq. (5.3) we obtain $l_B Z^2 = 1.19 \times 10^7 \mu m$ and $l_B Z^2 = 1.35 \times 10^7 \mu m$ for the two densities $\rho = 0.00215\mu m^{-2}$ and $\rho = 0.005\mu m^{-2}$, respectively. Theses values are consistent, and the differences in the pre-factor A are mainly due to the different screening lengths for the two densities.

High densities

The RDFs of the higher densities $\rho = 0.0133\mu m^{-2}$, $\rho = 0.0175\mu m^{-2}$ and $\rho = 0.02\mu m^{-2}$ are much more peaked than the lower density cases in the previous section. Therefore, particles must be located more precise at the peak positions, which can lead to attractive potentials when simulating pair interactions.

Fig. 5.5 shows the radial distribution functions for the densities $\rho = 0.0133\mu m^{-2}$ and $\rho = 0.0175\mu m^{-2}$. In this case, the Inverse Boltzmann method succeeds in matching the RDFs nicely. With the Yukawa tuning method this was not possible satisfactorily which indicates that the exact RDFs can't be matched with a Yukawa potential. Since we do not assume any kind of potential in the Inverse Boltzmann method, we are able to match the RDFs with a potential that has not only a repulsive part like the Yukawa potential but also an attractive part which reminds of a Lennard-Jones potential. Fig. 5.6(a) shows the corresponding potentials for the two densities $\rho = 0.0133\mu m^{-2}$ and $\rho = 0.0175\mu m^{-2}$. We do not show the RDF and potential for $\rho = 0.02\mu m^{-2}$, because the Inverse Boltzmann didn't converge in this case.

Fig. 5.6(b) shows the comparison of a simulation with the adapted Yukawa parameters from Ref. [107] for our current colloids, $A = 750000\mu m$ and $\kappa = 1.83\mu m^{-1}$

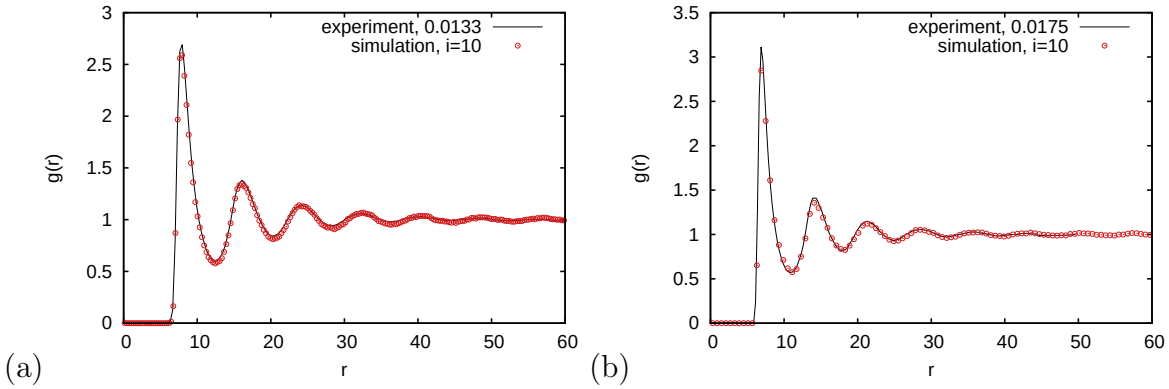

Figure 5.5: Inverse Boltzmann for high densities: (a) $\rho = 0.0133\mu m^{-2}$ and (b) $\rho = 0.0175\mu m^{-2}$ for iteration 10, respectively. The RDFs can be matched nicely with this method.

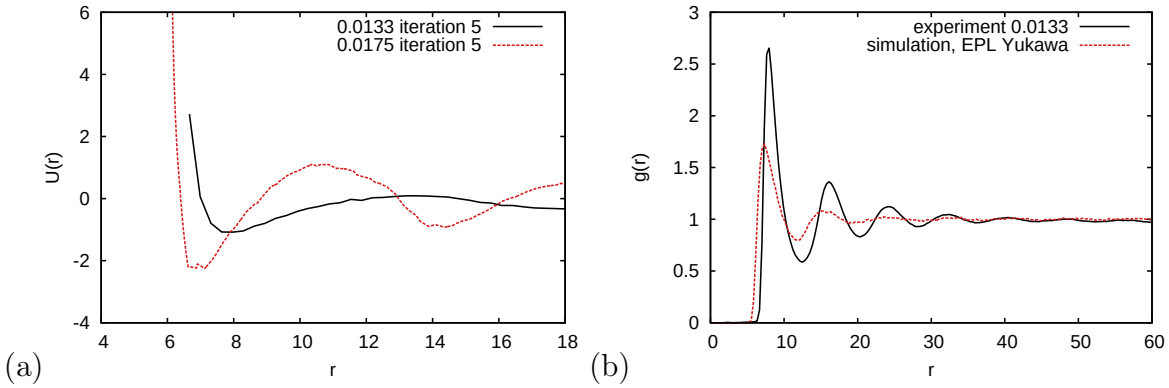

Figure 5.6: (a) Potentials identified by the Inverse Boltzmann methods for the densities $\rho = 0.0133\mu m^{-2}$ and $\rho = 0.0175\mu m^{-2}$. The potentials show in addition to the repulsive part also an attractive part which is necessary to obtain the sharp peaks in the RDFs. (b) Comparable simulation to Ref. [107] and to the work of Sec. 5.4.1 for density $\rho = 0.0133\mu m^{-2}$ with $A = 750000\mu m$ and $\kappa = 1.83\mu m^{-1}$ with $\sigma = 3.0\mu m$.

with $\sigma = 3.0\mu m$ for the density $\rho = 0.0133\mu m^{-2}$. As can be seen, for the present experimental datasets the RDF can't be reproduced with the size-adapted colloidal interaction parameters like from the work of [107].

We have seen, that for the reproduction of the complete RDFs of the higher densities an attractive part of the potential is necessary, which could be identified by the Inverse Boltzmann method. This is likely due to three-body effects beyond Debye-Hueckel theory, which are present at these higher densities. Hence, a Yukawa fit is not possible to match these potentials for reproducing the RDFs. However, the pair interaction is responsible for the first ascent of the RDF of the first peak. At this part, the Yukawa

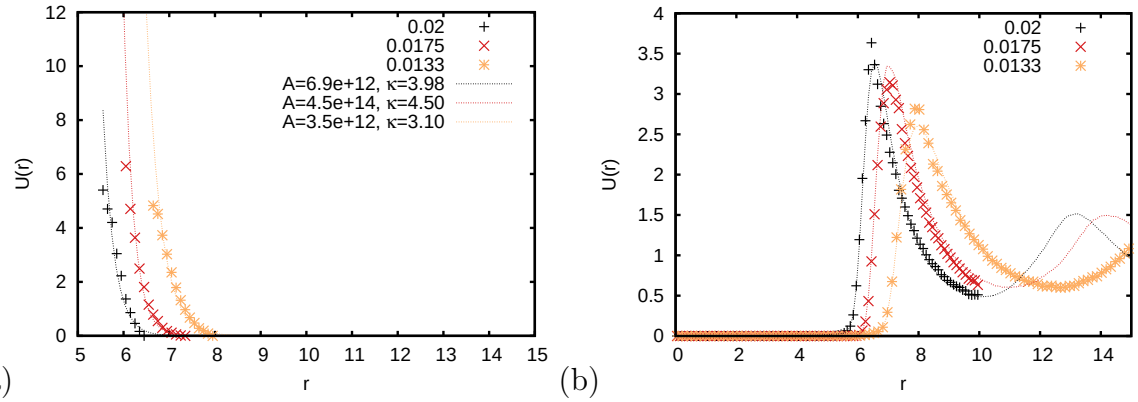

Figure 5.7: (a) Comparison of the potentials obtained by the Inverse Boltzmann method when only taking the first ascent of the RDF into account for the densities $\rho = 0.0133\mu m^{-2}$, $\rho = 0.0175\mu m^{-2}$ and $\rho = 0.02\mu m^{-2}$. (b) The first important part of the particular RDF is also matched using the repulsive potential from (a).

$\rho[\mu m^{-2}]$	$A[\mu m]$	$\kappa[\mu m^{-1}]$
0.0133	6.9×10^{12}	3.98
0.0175	4.5×10^{14}	4.50
0.02	3.5×10^{12}	3.10

Table 5.3: Inverse Boltzmann Yukawa fitting results for the higher densities. The screening length κ^{-1} is in the range of $222nm - 323nm$.

potential decays very fast and many-body interactions don't play a role. In the next section, we try to simulate this part in great detail.

Comparison of the repulsive part at high densities

In this section we use the Inverse Boltzmann method to find potentials, which match the first ascent of the first peaks of the particular RDFs for the densities $\rho = 0.0133\mu m^{-2}$, $\rho = 0.0175\mu m^{-2}$ and $\rho = 0.02\mu m^{-2}$. Therefore, many frames of the experimental system have been used to sample the RDFs in higher detail compared to the previous sections to obtain more datapoints for the first (important) parts of the RDFs.

Fig. 5.7(a) shows the potentials which were obtained when only the ascent of the first peaks of the RDFs were taken into account using the Inverse Boltzmann method. As expected, the potentials do not show an attractive part. In addition, at least the first important peak of the corresponding RDFs could be matched using such an approach for the potential (Fig. 5.7(b)).

Table 5.3 shows the corresponding values for a Yukawa fit to these potentials, which is also shown in Fig. 5.7(a). From these fits and for this case we can conclude, that the

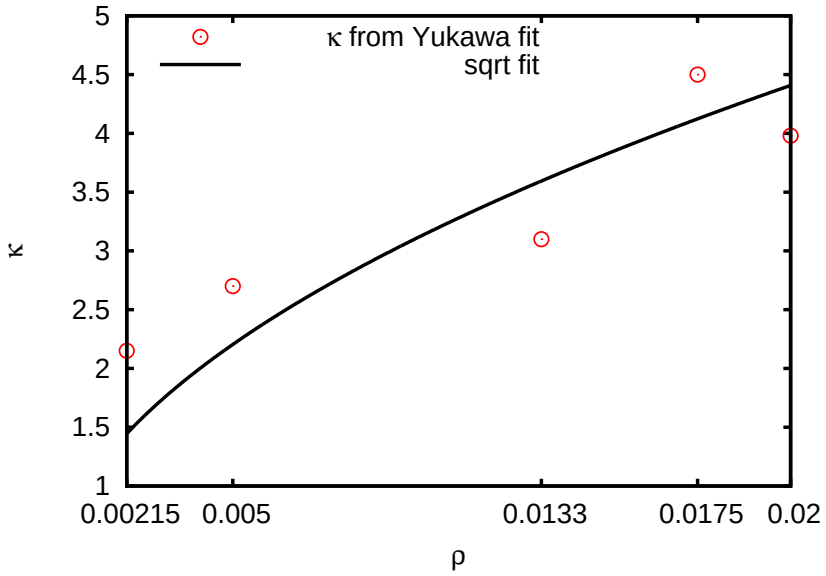

Figure 5.8: Dependence of the inverse screening length κ on the number density. Symbols denote κ values from the Inverse Boltzmann method, the straight line depicts the theory.

screening length for the Yukawa potential is in the range $222nm - 323nm$. We remark, that in the experimental system the screening length changes with density, because ions are dissociated by the colloidal particles whose density changes. Furthermore, the screening length can change during time evolution, because the solution absorbs CO_2 from the environment, for example, and hence the ion concentration changes which influences the screening length. The latter aspect is difficult to investigate, but we are able to address the dependence of κ on the density ρ.

5.4.3 Dependence of the screening length on the density

The inverse screening length in the DLVO theory is given by

$$\kappa = (8\pi l_B c_s)^{1/2} \tag{5.5}$$

with the Bjerrum length of water $l_B = 0.7nm$ and the reservoir salt concentration $c_s = (N_d\rho)/\sigma$ when assuming a layer height of σ, where N_d is the number of dissociated charges, and ρ the density of the colloids. Fig. 5.8 shows the dependency of κ on the density ρ in our measurements. From Eq. (5.5) we expect a behavior which is $\kappa \propto (\rho)^{1/2}$, therefore we fit a trend $\kappa(\rho) = a_f(\rho)^{1/2}$ to the values obtained from the simulations (line in figure 5.8). For the fitting parameter we obtain $a_f = 31.17$, hence this leads to the number of dissociated charges per colloid with diameter $\sigma = 3.9\mu m$

via

$$(8\pi l_B N_d \rho/\sigma)^{1/2} = a_f(\rho)^{1/2} \tag{5.6}$$

$$\Longrightarrow N_d = \frac{\sigma a_f^2}{8\pi l_B} \approx 2 \times 10^5. \tag{5.7}$$

The surface of the spherical particles has an area of $A = 4\pi(\sigma/2)^2 = 47.78\mu m^2 = 4.778 \times 10^7 nm^2$, which means that – speaking in orders – approximately every 100-th charge on the colloidal surface is dissociated in this case. For a different layer height, this number must be scaled accordingly.

5.5 Conclusions

In this section we have shown that MD simulations can be used to determine not easily accessible quantities in experimental systems as well as that the Yukawa interaction model can indeed be used to model the interaction in real colloidal systems except for high densities, where multi-body interactions may play a role.

However, also in these systems the main part of the repulsive interaction could be reproduced by the Inverse Boltzmann method and fitted to a Yukawa shape, which gives reasonable values for the screening length, which was – depending on the density – in the range of $222nm - 500nm$ and could be characterized by the DLVO theory in Sec. 5.4.3.

In conclusion, we can say that a Yukawa pair interaction can be used to model the screened effective forces in colloidal systems which substantiate our intention to simulate the crystallization of charged macromolecules using such a model.

6 Crystallization of charged macromolecules at low supersaturations

During this work, many simulations have been performed to investigate the crystallization of charged macromolecules. However, presenting every simulation would go beyond the scope of this thesis. Therefore, the results with the highest physical significance are presented in great detail. Some of the findings of this chapter are part of the following publications [109, 110]:

- Kai Kratzer, Dominic Roehm, and Axel Arnold — Homogeneous and Heterogeneous Crystallization of Charged Colloidal Particles. High Performance Computing in Science and Engineering '14, Springer International Publishing (2014).

- Kai Kratzer and Axel Arnold — Two-stage crystallization of charged colloids at low supersaturations, arXiv:1410.8695 (2014).

6.1 Introduction

The crystallization of charged macromolecules plays an important role in many fields, such as biology, soft matter physics or materials science. For example, proteins are crystallized for structure determination by scattering [15, 111, 112].

In experiments, macromolecules can be investigated in great detail due to their well known properties and the fact that they can be investigated by microscopy or scattering methods [107, 113], e.g. investigations have been performed concerning nucleation rates for different densities [114, 115, 116]. In computer simulations, the Yukawa interaction introduced in Sec. 2.3 is used to study the dynamics of such systems.

However, despite of recent advances in colloidal crystallization [1, 9, 13, 14], there is still no closed theory of the crystallization of charged macromolecules. For example, the mechanisms and dynamics how crystals nucleate are not yet fully understood. Is there a two-stage crystallization process or not, and how are the different crystallite structures selected, e.g. is an fcc-like core established in a previously nucleated bcc-like structure [9]? What are the crystallization pathways and how are structures converted during this process? Are there precursors like density changes or structural

ordering already at an early stage, which predict the formation of a crystallite in the bulk [13, 14, 117, 118]?

According to Ostwald, the phase which is closest in free energy is nucleated first, which doesn't have to be the stable phase [76]. In addition, Stranski and Totomanow found that the phase with the lowest free energy barrier is nucleated first [119], and according to Alexander and McTague, the nucleation of a bcc-like phase is favored in a liquid [120].

For investigating the crystallization process in full detail, experiments or simulations at low supersaturations close to the phase coexistence lines are necessary, where the attachment rate of growth units is slow and the nucleation process can be studied in great detail. In our simulations, we have to advance to low pressures P where the attachment rate is small, which means that the crystal doesn't grow abruptly including usually many defects. In addition, if the attachment rate would be high, multiple crystals are able to nucleate simultaneously, leading to many domains rather than a large monocrystal.

In this work, we contribute to a clearer picture of crystallization by using our simulation model of charged macromolecules together with rare event sampling simulations (Sec. 2.5 and chapter 3) to access the onset of crystal growth at low supersaturations, where the energy barrier towards nucleation is high (Sec. 2.4). This allows us to investigate the mononuclear nucleation process directly which is triggered only by spontaneous fluctuations of the homogeneous bulk, and to study the mechanisms which are necessary to form a certain crystallite structure.

6.2 Details of the investigations

6.2.1 Simulation

We perform MD simulations in the isothermal-isobaric ensemble, realized by a Langevin thermostat and a barostat [55], in a 3D box with periodic boundaries using ESPResSo [17] together with FRESHS [98] and with the FFS optimizations from Ref. [94]. As a reference for our simulations, we use the well-known phase diagram of the Yukawa model [62, 121].

The simulation is set up in the initial state A, which is the liquid state, by distributing the desired number of particles at random in the 3D simulation box and a following warmup equilibration[1]. This leads to an undercooled metastable liquid state, where the particles are not in a crystal structure because we distributed them randomly and the energy barrier towards crystallization prevents the system to advance spontaneously to the stable crystallized state. The way towards the final state B is then characterized by the order parameter.

[1]For details refer to the ESPResSo user's guide, `http://espressomd.org`

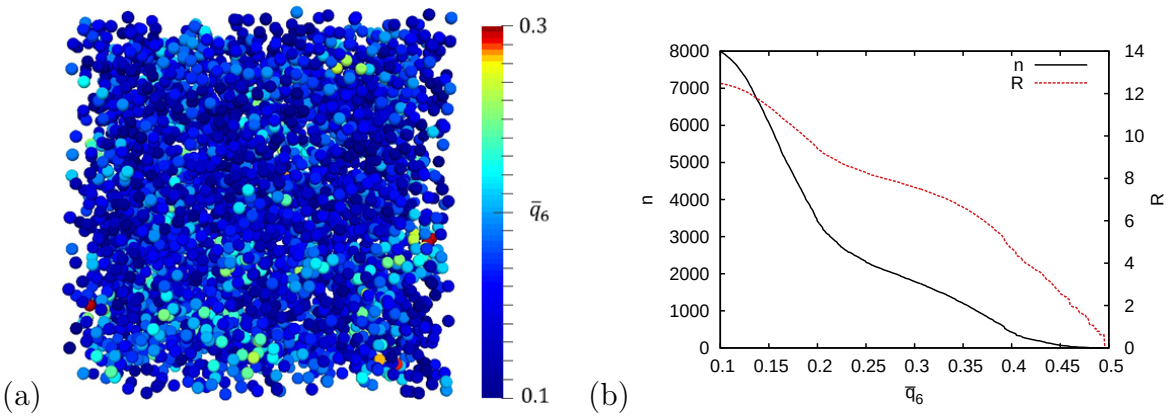

Figure 6.1: (a) Snapshot of the simulated system in the initial undercooled liquid state. The coloring of the particles is according to the fluctuating $\overline{q}_6$ parameter. (b) Dependency of the cluster size n and the effective cluster radius R on the $\overline{q}_6$ threshold for a typical system snapshot during crystallization.

We simulate for different contact energy values ϵ and for different pressures P. All simulations have been performed with an inverse screening length of $\kappa = 5$, which is comparable to the length scales of the experimental colloidal system of chapter 5. Note, that we use reduced units in this chapter. We fix the simulation timestep $\tau = 0.01$, and $k_BT = 1$ in our simulations. The simulation model is the screened Coulomb (Yukawa) model, where we set the cutoff of the interaction potential like described in Sec. 2.3, Eq. (2.34). For FFS, we use the exploring scouts method and set $p_{\mathrm{des}} = 0.5$, $d_{\mathrm{min}} = 1$, and $M_{\mathrm{trial}} = 15$. Further parameters are given in the particular sections.

6.2.2 Order parameter

We use the size of the largest crystal cluster in the system as order parameter (compare Sec. 2.4.3), detected via a cluster analysis of solid neighbor particles identified by $\overline{q}_6 \geq 0.29$. Fig. 6.1(a) shows a typical snapshot of such a system in the initial liquid state A, with the coloring of the particles according to the $\overline{q}_6$ parameter. In this illustration, the fluctuations of the $\overline{q}_6$ parameter in the liquid can be seen. Domains with higher $\overline{q}_6$ values are possible candidate domains for a crystal seed. During time evolution of the conventional simulation run solid particles with $\overline{q}_6 \geq 0.29$ occur and vanish at different locations in the supersaturated liquid.

Note, that in this model a single particle can be identified as solid like, which sounds nonphysical at first sight. But, the solid particle detection depends on the environment of a particle, and if the particle is embedded in a particular crystal lattice, it is considered as solid-like. However, regarding the neighbors, their environment must not be solid-like which leads to the case that they aren't labeled as solid part of the cluster.

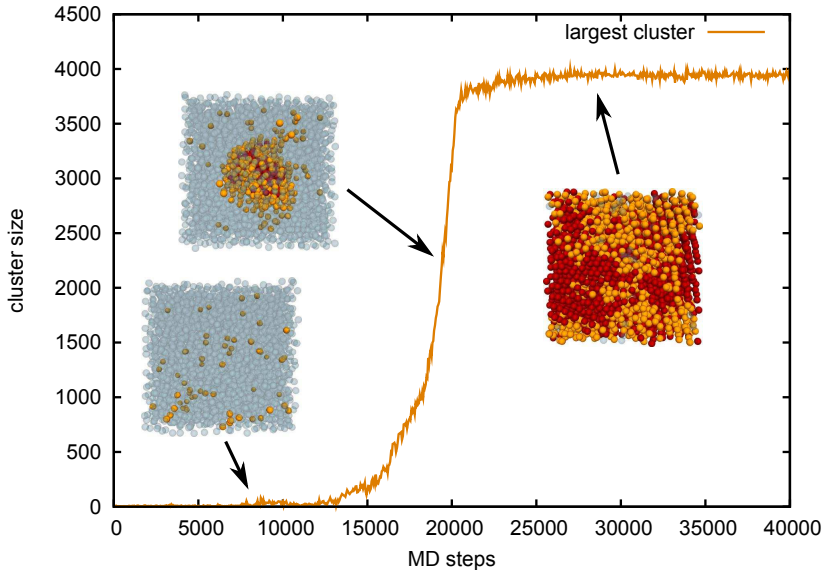

Figure 6.2: Order parameter visualization: Growth of the largest cluster of solid particles in a crystallization simulation, detected via $\overline{q}_6 \geq 0.29$. The images depict snapshots of the largest cluster at the indicated points. Blue colorcoding is for liquid-like particles, red and golden stand for solid-like particles of fcc-like and bcc-like type, respectively.

6.2.3 Crystal cluster analysis

The labeling of solid particles depends on the selection of the threshold of the $\overline{q}_6$ parameter, which influences the absolute nucleus size n as well as the effective cluster radius given by

$$R = \left(\frac{3n}{4\pi\rho}\right)^{1/3}, \tag{6.1}$$

with the number density ρ, assuming spherical cluster growth. Fig. 6.1(b) shows the change of the cluster size n and radius R on the $\overline{q}_6$ threshold for a typical system snapshot of the crystallization pathway. As a consequence, the threshold for the $\overline{q}_6$ parameter should be kept constant in the simulations to obtain comparable results for the cluster sizes. To advance the simulation using FFS, this threshold plays a minor role, because if it is kept constant the cluster size grows monotonously during the advancement towards the crystallized final state B, which fulfills the requirement of the order parameter for FFS. Note, that in our case the threshold of $\overline{q}_6 \geq 0.29$ is not arbitrarily. We choose this value such, that it represents a dividing layer between the liquid domain and a certain solid crystal structure, see also Sec. 2.4.3 and Ref. [82].

Fig. 6.2 visualizes a typical order parameter transformation during the crystallization process for a fixed $\overline{q}_6$ threshold, in this case for a brute-force simulation at a high pressure of $P = 42$, where the system crystallizes spontaneously.

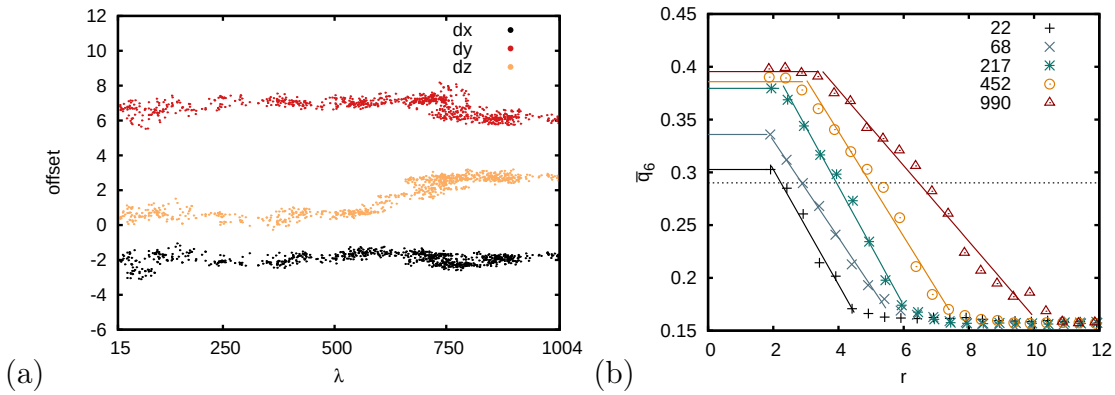

Figure 6.3: Nucleation process: (a) shift of the center of mass of the largest cluster in the system in the directions dx, dy and dz relative to the initial box coordinates. (b) radial analysis of the $\overline{q}_6$ order parameter for different cluster sizes, indicated by the symbols. The dashed line depicts the $\overline{q}_6 = 0.29$ threshold which was used to determine the cluster size.

As can be seen, the crystallization process can be characterized by this order parameter. Here, only solid particles and no certain crystal lattice structure is detected. This is done in the post-processing, where different crystal structures can be distinguished by also taking the $\overline{q}_4$ parameter into account.

For the radial analysis of the cluster, we create spherical shells around its center of mass with

$$V_{\text{shell}} = \frac{4}{3}\pi(r_o - r_i)^3, \tag{6.2}$$

where r_o is the outer radius and r_i the inner radius of a particular shell. To average over the same number of particles, we choose the sphere shell radii such, that the number of particles is constant in every shell, e.g. $N = 5$ particles. We use the mean of these two radii as the value for the histogram bin,

$$r = \frac{1}{2}(r_i + r_o). \tag{6.3}$$

Fig. 6.3(a) shows the position of the center of mass of the largest crystallite cluster during a successful FFS simulation pathway. The center of mass position is nearly constant in this case, which indicates that we're looking at the same cluster during the growth process and no second cluster with competing size is in the system at the same time, which would lead to jumps in the position of the center of mass. Note, that this constant behavior is already an indication for balanced growth in each direction of the 3D system.

Fig. 6.3(b) shows the radial analysis around the center of mass of the $\overline{q}_6$ order parameter for different cluster sizes during the nucleation process, created with the help of the spherical shells around the center of mass according to Eq. (6.2) and

ϵ	P	ΔP_{tail}	ρ_A	ρ_B	η_A	η_B	k_{AB}	λ_B structure
2	25.72	0.72	0.9739	1.0103	0.5099	0.5290	8.5×10^{-35}	hcp/fcc
2	27.75	0.75	0.9943	1.0262	0.5206	0.5373	4.0×10^{-18}	hcp/fcc
2	28.77	0.77	1.0041	1.0358	0.5257	0.5423	1.2×10^{-15}	hcp/fcc
8	35.67	0.67	0.8188	0.8288	0.4287	0.4339	2.6×10^{-26}	bcc/hcp
8	38.71	0.71	0.8408	0.8508	0.4402	0.4455	1.0×10^{-14}	bcc/hcp
8	40.73	0.73	0.8547	0.8648	0.4475	0.4528	2.1×10^{-10}	bcc/hcp
20	23.35	0.35	0.5467	0.5514	0.2863	0.2887	3.2×10^{-24}	bcc
20	25.37	0.37	0.5618	0.5668	0.2942	0.2968	3.7×10^{-16}	bcc
20	28.40	0.40	0.5830	0.5877	0.3053	0.3077	2.3×10^{-10}	bcc/hcp

Table 6.1: Simulation details and crystallization rates k_{AB} for different values of ϵ and P for $\kappa = 5$. The number densities ρ and volume fractions η are determined from the snapshots on λ_A and λ_B, respectively. The number of particles in all systems is $N = 8192$. The last column indicates the crystal structure of the main part of the cluster at the border of B.

Eq. (6.3). The datapoints correspond to an averaged value in time over 20 integration steps, and the cluster size in the legend is the averaged cluster size during these steps. Fig. 6.3(b) demonstrates the dependence of the cluster size on the choice of the $\overline{q}_6$ threshold. In contrast to the Classical Nucleation Theory, where one assumes a sharp transition between old and new phase, there is a continuous transition from the liquid-like to the solid-like particles, which we will consider when we compare our simulations to this theory. The transition region is larger than the cluster at the critical nucleus.

6.3 Results

In this section we present the main results of the Forward Flux Sampling simulations. First, we address the dependency of the crystallization rates on the system parameters pressure P and the pair interaction contact value ϵ. By investigating the crystallization rates, the rare transitions can be quantified.

6.3.1 Crystallization rates

Table 6.1 gives an overview of the simulation details and transition rates k_{AB} in $[\tau^{-1}\sigma^{-3}]$ for different contact values ϵ and pressures P, which leads to a certain number density ρ and volume fraction η for the states A and B. In addition, the values for the pressure tail corrections ΔP_{tail} which are calculated using Eq. (2.38) are given in this table. The error in the rates k_{AB} ranges from ± 3 in the exponent for the lower rates to ± 1 in the exponent for the higher rates. To improve the error, a lot of additional computational effort would have to be spent in simulating more trajectories.

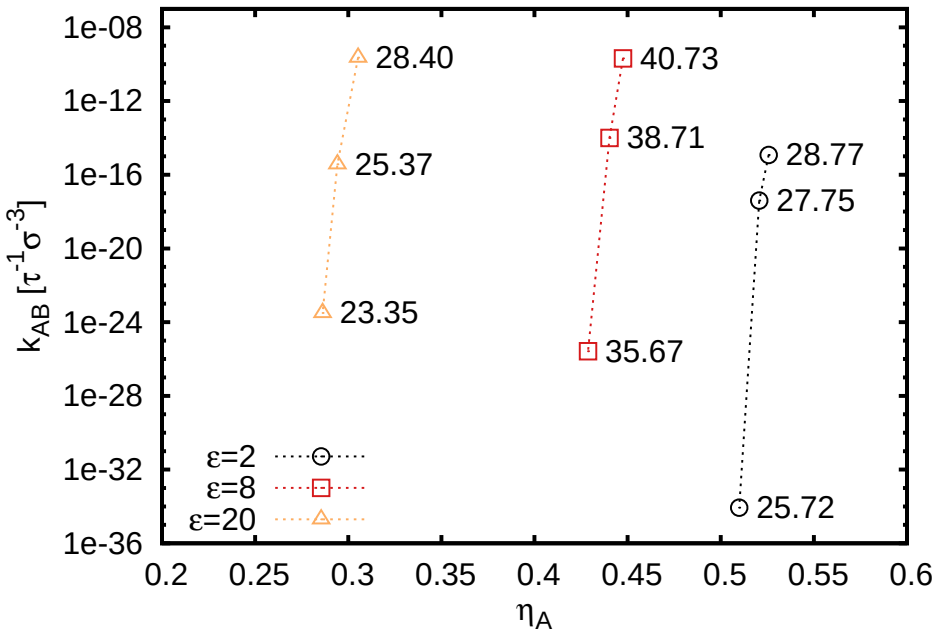

Figure 6.4: Transition rates k_{AB} for different initial volume fractions on the first interface η_A, where the system is liquid-like, to the final state B, where $\geq 90\%$ of the particles are solid-like. The datapoints are grouped by the contact value ϵ, and the number besides a datapoint denotes the pressure in the simulation. With decreasing pressure P, the volume fraction is lower which decreases the transition rate k_{AB} drastically.

All these simulations have been performed with $N = 8192$ particles, with the border of the liquid state A located at $\lambda_A = 15$ and with the border of the final state B located at $\lambda_B = 7300$. Table 6.1 also contains the information about the main composition of different structures of the crystal cluster at the border of state B. This must not be the final structure of the stable state in the phase diagram, but an indication for the spontaneous nucleated structure for the state where 90% of the particles are in the largest cluster. There is a tendency to higher $\overline{q}_4$ values for higher pressures and lower contact values which is caused by hcp-like and fcc-like structures at these conditions.

Fig. 6.4 shows the values for k_{AB} plotted versus the volume fraction η_A of the snapshots at the border of the liquid state λ_A. With decreasing pressure P the volume fraction is lower, the system is closer to the coexistence line which means a lower chemical potential difference $\Delta\mu$ and hence a lower transition rate k_{AB}.

Note, that the transition rates are the outcome from considering all trajectories of a complete FFS simulation, which includes the successful and non-successful pathway branches in the FFS splitting scheme. In the following sections, we analyze the successful pathways to investigate – amongst others – the nucleation mechanisms.

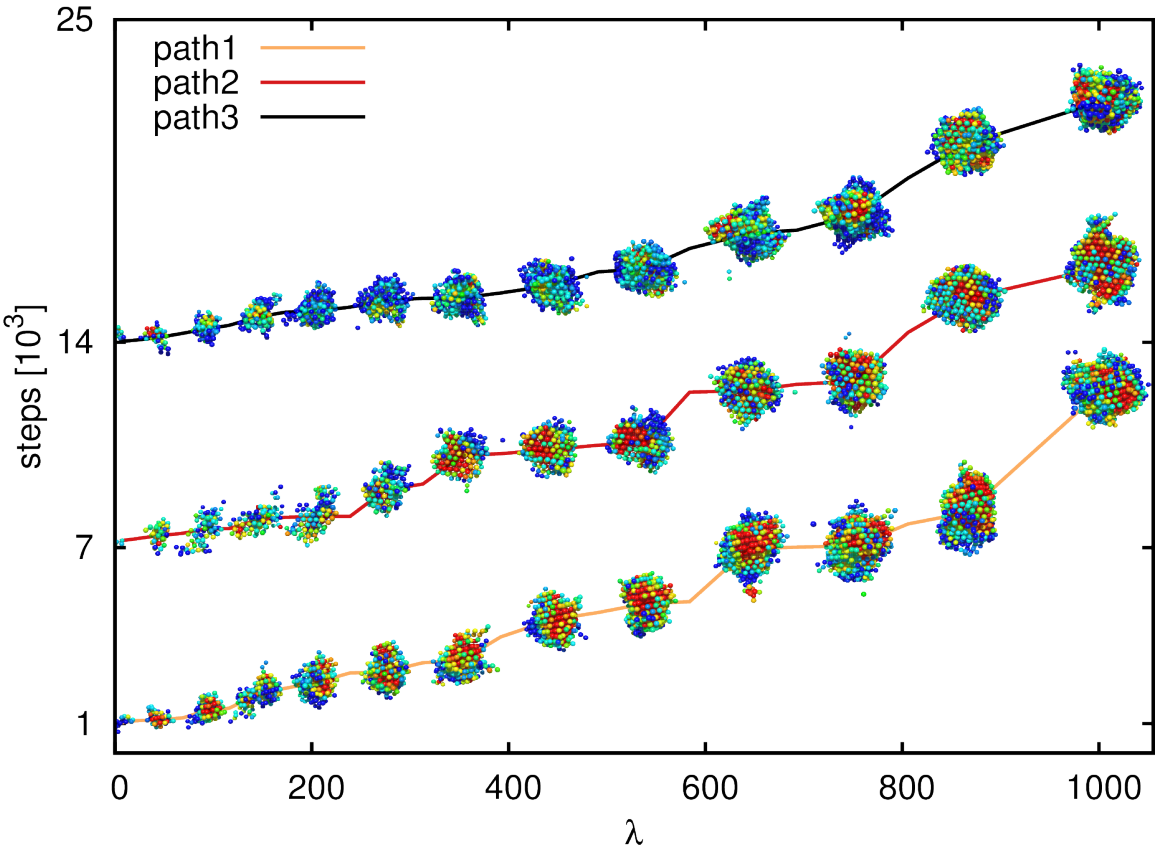

Figure 6.5: Nucleating critical clusters at a low contact energy, $\epsilon = 2$, and a pressure $P = 25.72$. Blue color coding depicts a bcc-like structure with $\overline{q}_4 \leq 0.05$ and red color coding denotes fcc-like particles with $\overline{q}_4 \geq 0.1$. The colors in between stand for hcp-like structures (values according to Ref. [82]). In this case, a bcc, hcp, and fcc signature is visible. Note, that only approximately every second interface snapshot is shown in this image.

6.3.2 Nucleation pathways

For a low contact energy of the potential, $\epsilon = 2$, and a pressure $P = 25.72$, the averaged volume fraction of the liquid state on λ_A is $\eta_A = 0.5099$ and changes to $\eta_B = 0.5290$ during crystallization. Fig. 6.5 shows the nucleation of critical clusters for these settings, which are successful traces of the FFS simulation up to the second to last automatically placed interface λ_{n-1}, which is possible because we set the target transition probability to $p_{\text{des}} = 0.5$. As a result there is no additional interface placed beyond a committor value of 0.5, which corresponds to the committor value at the critical cluster size [94, 24]. During the nucleation process, bcc-like, hcp-like, and fcc-like crystal structures can be observed in the clusters.

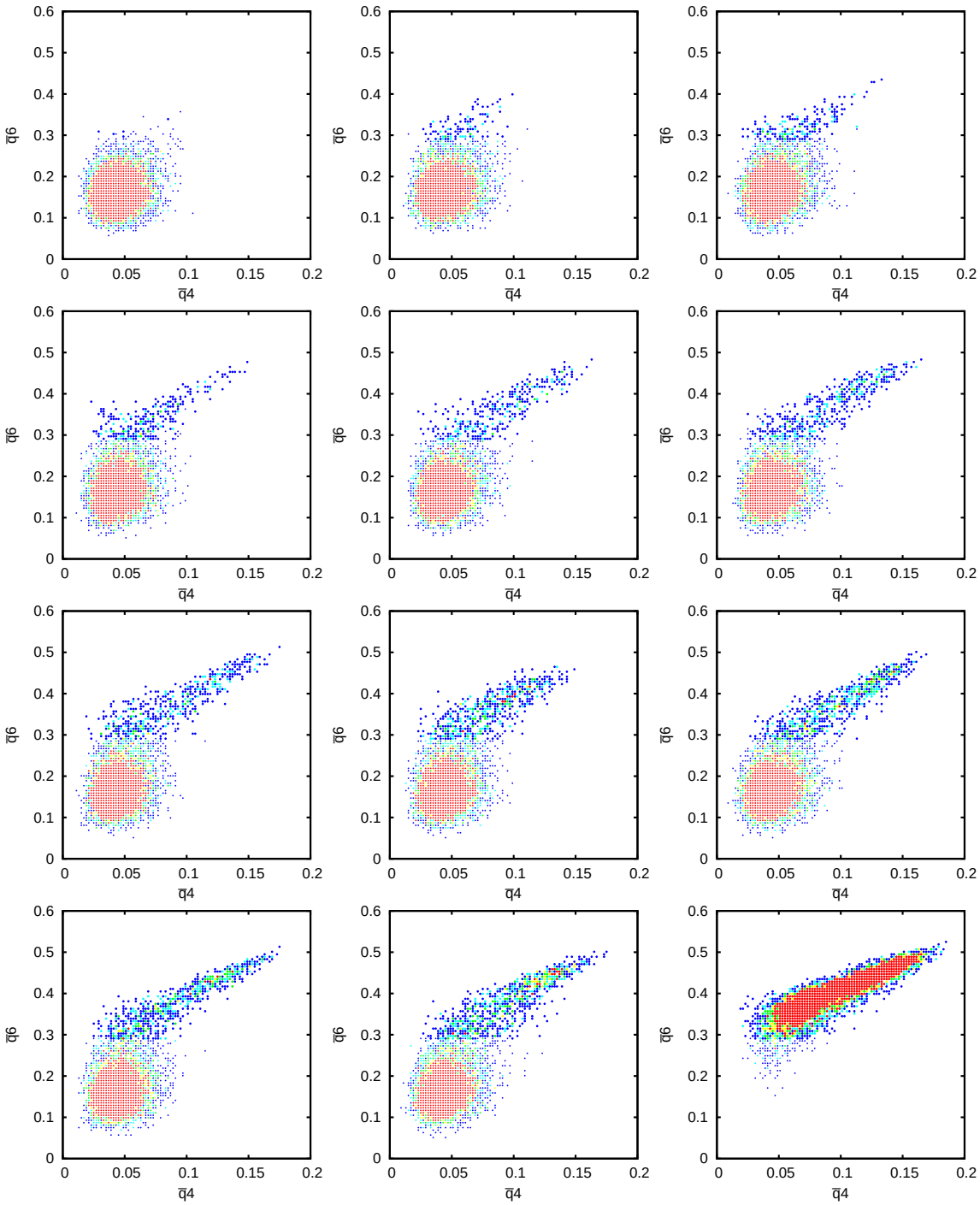

Figure 6.6: Time series of the $\overline{q}_4$ and $\overline{q}_6$ parameters per particle for $\epsilon = 2$. Blue color coding stands for 1 particle, a red point depicts ≥ 5 particles. The larger points denote particles in the largest cluster. The second to last plot corresponds to the system which contains the critical cluster, and the last plot shows the system at the border of state B.

For an improved analysis of the particle structures during a crystallization trace, we consult the distribution of the $\overline{q}_4$ and $\overline{q}_6$ parameters of all particles in the system. This is visualized in Fig. 6.6, which shows a time series of selected snapshots of one crystallization trace of Fig. 6.5 as scatter plots of the $\overline{q}_l$ parameters. In Fig. 6.6 all particles which are solid-like and in the largest cluster are depicted as larger dots. The smaller dots represent particles which are not in the largest cluster. In addition, the shown domain of the order parameters was subdivided into 100 bins for each direction and the color of the dots represents the count of each bin to indicate the domains where many particles have similar order parameter values.

At the beginning of the time series, the system is in the liquid state, where the order parameters of most particles are $\overline{q}_4 \leq 0.1$ and $\overline{q}_6 \leq 0.29$. Only a few particles are beyond these boundaries, and a very small fraction forms the largest solid cluster in the system (approximately 5 larger dots in the first scattering plot). In the following image mainly the $\overline{q}_6$ values have increased, and more particles are in the largest cluster. Then, the values are scattered also to higher $\overline{q}_4$ domains, and the density of the dots with $\overline{q}_4 \geq 0.1$, which is considered as fcc-like, increases during nucleation of the cluster until the critical size, which is until the second to last image. The transition from the second to last image to the last image illustrates the change of the system from the state which contains the critical cluster to the final state on λ_B where 90% of the particles are in the largest solid cluster. The system shows a strong tendency to the hcp/fcc-like domain. Note, that the system is only driven to higher $\overline{q}_6$ values, but it is not driven towards higher $\overline{q}_4$ values. The increase in the $\overline{q}_4$ parameter can be seen as a result of the ordering process under the preset simulation conditions.

For further investigations and to visualize the cluster growth with the relevant order parameters we show slices of the simulation box in x-y, x-z, and y-z direction, with a thickness of plusminus the neighbor cutoff, which was determined by the first minimum of the radial distribution function $g(r)$. For better comparability, all system snapshots have been transformed to the center of mass of the critical cluster at λ_{n-1}. This allows us to study the location in the system where the critical cluster will form at an early stage, e.g. for the identification of precursors to the onset of crystal nucleation. Fig. 6.7 shows this analysis for the critical cluster with a size of approximately 1000 particles.

The images depict slices through the center of mass in every box direction, where images in the same column correspond to the same slice. The brackets $\langle S \rangle$ around a system quantity S denote an average of the particle's value itself with the values of its N nearest neighbors within the cutoff radius. The first row in Fig. 6.7 shows the averaged $\overline{q}_4$ order parameter and the second row the $\overline{q}_6$ order parameter. The last row shows the averaged neighbor distance $\langle r \rangle$ of the particle with its nearest neighbors, which characterizes the local density.

As can be seen in the images, the $\overline{q}_4$ order parameter (first row) has higher values at the positions where the solid particles of the cluster are located, but the distribution

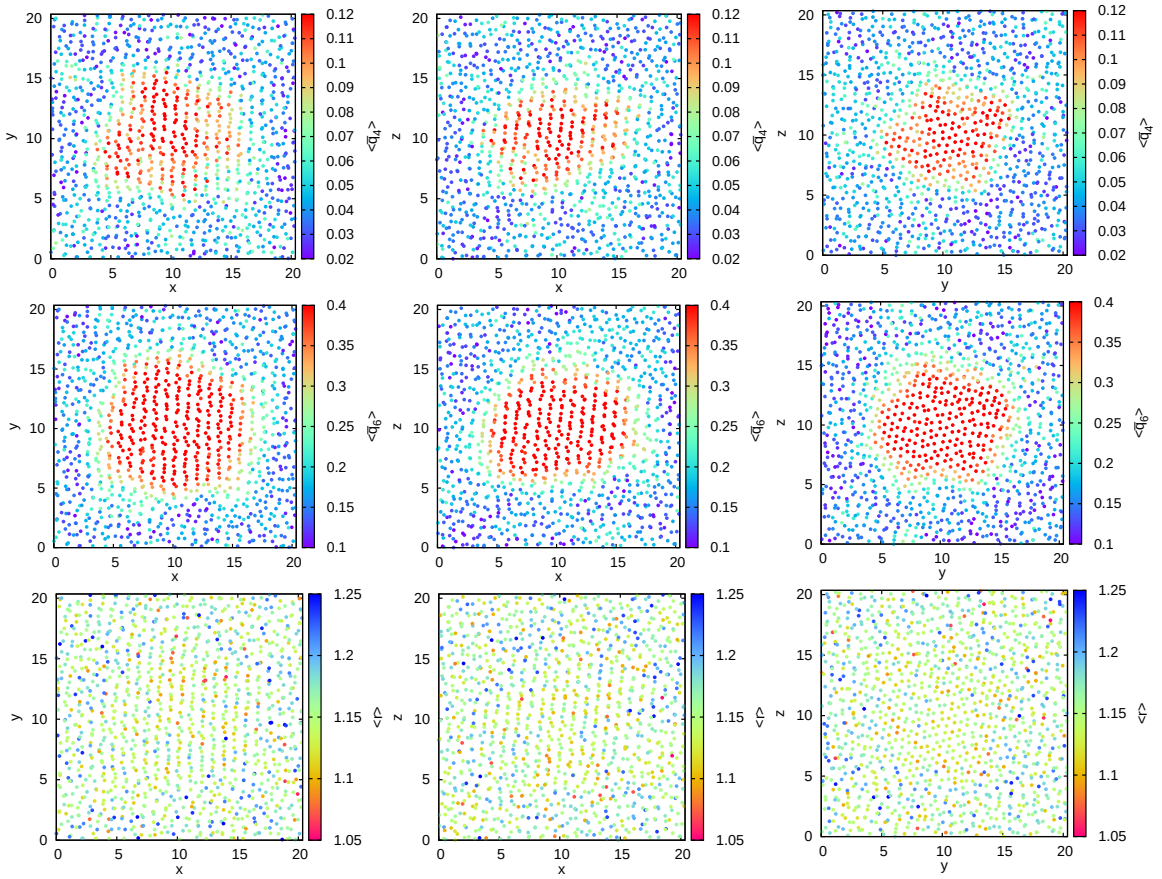

Figure 6.7: Slices of the 3D cluster near the critical size in every direction of the box (x-y, x-z, y-z) for $\epsilon = 2$. The slices have a size of plusminus the neighbor cutoff, determined by the first minimum of the RDF. The first row shows the $\langle \overline{q}_4 \rangle$ distribution, the second row the $\langle \overline{q}_6 \rangle$ distribution, and the last row the neighbor distance $\langle r \rangle$. The cluster shows a high ordering for both parameters, $\overline{q}_4$ and $\overline{q}_6$.

is not uniform across the cluster. This means that the cluster consists of fcc-like particles and also of hcp-like and of bcc-like particles in this case. In contrast, the values of the $\overline{q}_6$ order parameter (second row) correlate nicely with the circular shape of the cluster as expected from CNT, and the profile decreases smoothly at the border of the cluster. The alignment of the particles can also be seen in the panels of the last row. However, the particles corresponding to the cluster in the system do not show a significantly different local density compared to the other particles in the system. The different neighbor distances are distributed over the whole domain in these images.

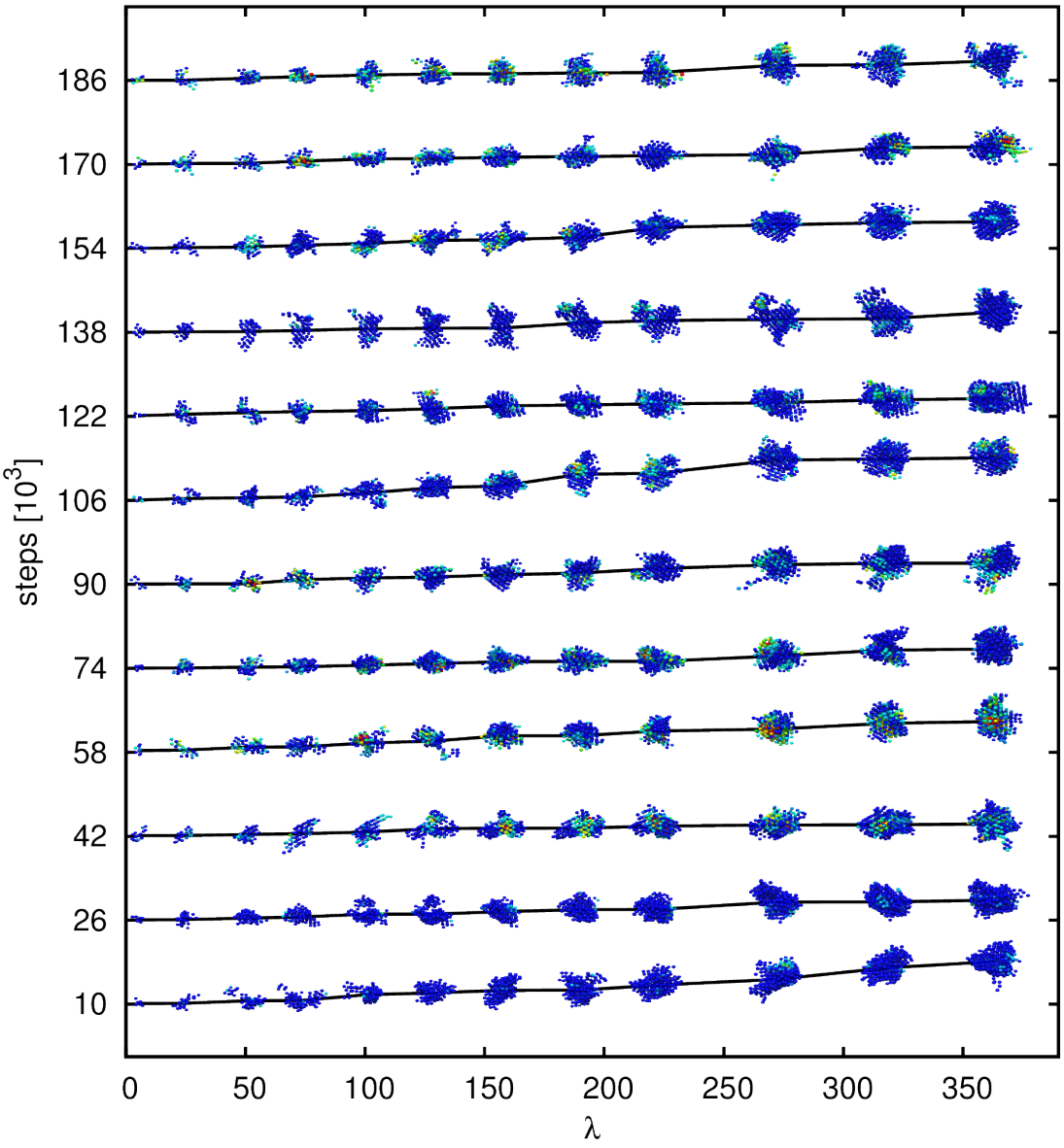

Figure 6.8: Nucleating critical clusters with $\epsilon = 20$ and $P = 25.37$. The color coding is the same as in Fig. 6.5. Despite of the fact that the phase point for these conditions is located in the fcc-like region of the phase diagram, the critical cluster pathways consist of mainly bcc-like particles in all cases, with only small fluctuations of hcp-like and fcc-like structures (compare to Fig. 6.5).

As a next step, we set the Yukawa contact value to $\epsilon = 20$ and simulate at a comparable pressure of $P = 25.37$. Fig. 6.8 shows the corresponding pathways for this higher contact energy. According to the phase diagram (see also Fig. 2.4), the stable phase of this point is fcc-like. As the color coding by $\overline{q}_4$ in the figure depicts, there are also fluctuations of this fcc-like structure, but they are much smaller than in the previous case for $\epsilon = 2$. The main part of the pathways contain clusters which consist of bcc-like particles, with a much smaller number of hcp-like and fcc-like particles compared to the previous case with $\epsilon = 2$. In addition, the critical cluster size is much smaller for $\epsilon = 20$ and consists of approximately 360 particles. From the last snapshots shown in Fig. 6.8, the system crystallizes spontaneously to a state which consists of mainly bcc-like particles. To quantify this observation, Fig. 6.9 shows the time series of the scatter plots of the $\overline{q}_4$ and $\overline{q}_6$ parameters for one representative successful trace:

- At the beginning, most of the particles are in the liquid domain and only a small number of larger dots are present, which stand for the particles in the largest cluster.

- In the following images of the time series, the dots shift mainly to higher $\overline{q}_6$ values without increasing the $\overline{q}_4$ order, but a few fluctuations are present.

- The second to last picture represents the configuration at interface λ_{n-1} near the critical cluster size, and the transition to the last image occurs during crystal growth until the border of state B.

Note, that in our Forward Flux Sampling simulations, we only ratchet the system towards states where more particles are included in the largest cluster, detected via the $\overline{q}_6$ parameter, and do not optimize for a certain structure. In this case, it is more convenient for the system to form a bcc-like structure spontaneously, than the truly stable fcc-like structure, as predicted by McTague [120].

Now, we present the investigations of the critical cluster for $\epsilon = 20$, analyzed in 3D slices through the simulation box as in the lower contact energy case before. Fig. 6.10 shows the $\overline{q}_4$ analysis in the first row and the $\overline{q}_6$ analysis in the second row as well as the averaged neighbor distance images in the last row. Columns correspond to the same slices. The critical cluster is much smaller for $\epsilon = 20$ than for $\epsilon = 2$. Note, that the scale for the $\overline{q}_4$ analysis in the second row is different than for the previous case in Fig. 6.7, the absolute values are much smaller. Here, the overall $\overline{q}_4$ values are much smaller. As we can see in the $\overline{q}_4$ panels, there are fluctuations present, but they are not as large as in the $\epsilon = 2$ case and have a different noncircular distribution. These fluctuations appear only for short periods of time and dissolve again. The $\overline{q}_6$ panels in the second row show nicely the circular shape and the higher $\overline{q}_6$ values in the center with a smooth decrease at the border. Again, no correlations to the solid cluster can be seen in density panels for $\epsilon = 20$.

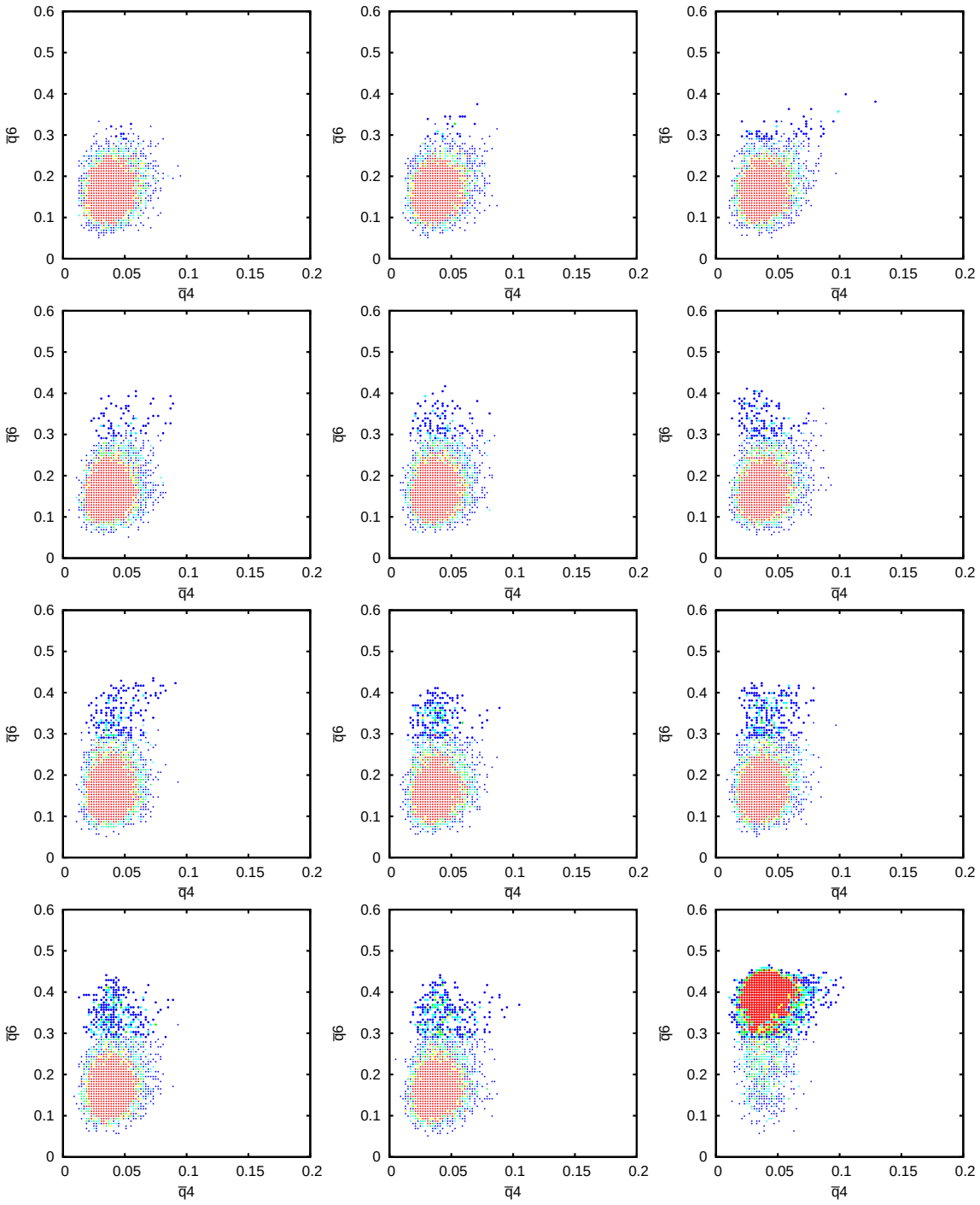

Figure 6.9: Time series of the $\overline{q}_4$ and $\overline{q}_6$ parameters per particle for $\epsilon = 20$ with the same color coding like in Fig. 6.6. The particles arrange mainly in the $l = 6$ order, with a bcc-like structure in the end.

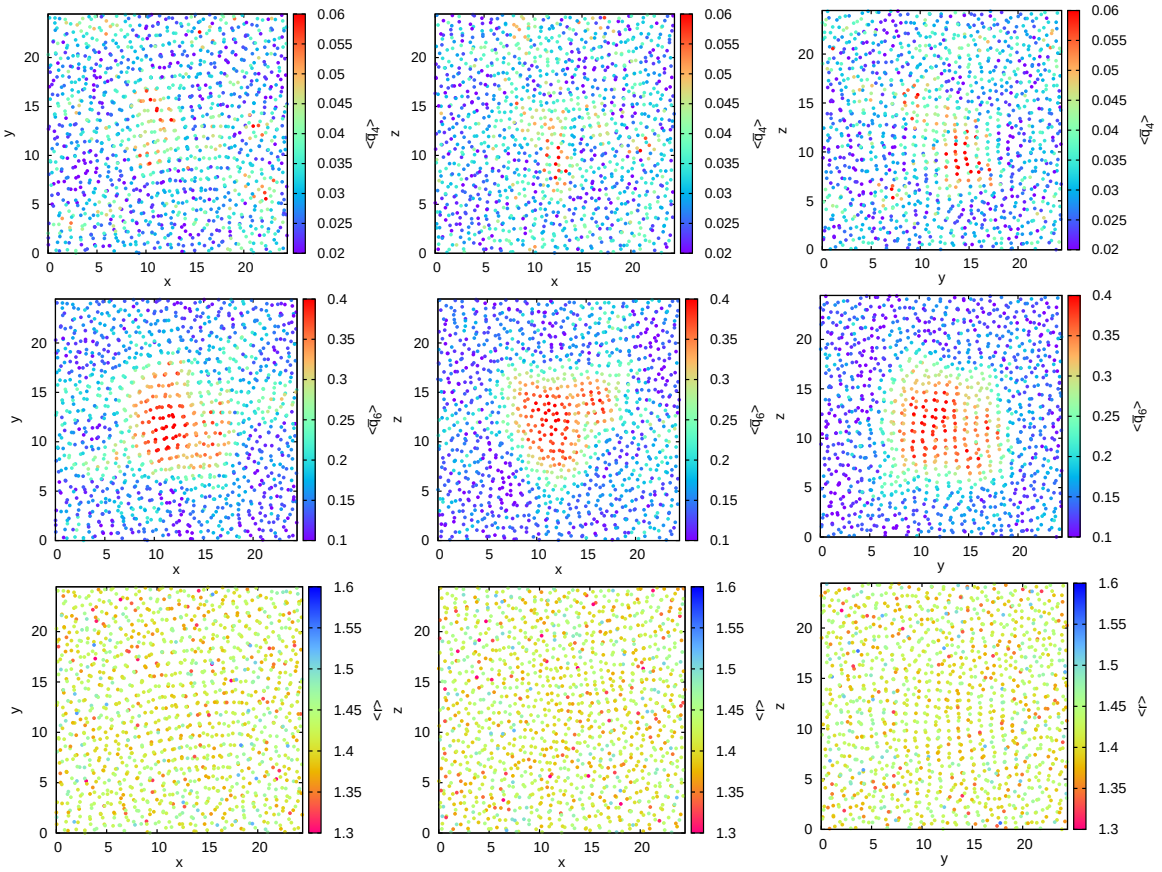

Figure 6.10: Slices of the 3D cluster near the critical size in every direction of the box (x-y, x-z, y-z) for $\epsilon = 20$. The slices have a size of plusminus the neighbor cutoff, determined by the first minimum of the RDF. The first row depicts the $\langle \overline{q}_4 \rangle$ distribution, the second row the $\langle \overline{q}_6 \rangle$ distribution, and the last row the neighbor distance average $\langle r \rangle$.

One question remains still open in the case for $\epsilon = 20$: The phase point of the system is located in the fcc-like domain in the phase diagram. However, our system crystallized to the bcc-like domain. To perform further investigations of this behavior we use the $\overline{q}_4$ order parameter together with an FFS simulation to optimize for clusters of fcc-like particles. We conducted the following approaches:

(a) Use the configuration points of the final state B as starting points for the FFS simulation with the $\overline{q}_4$-based order parameter. Here, the system is a defective bcc-like crystal.

(b) Set up a perfect bcc crystal lattice at the same simulation conditions and try to grow an fcc-like crystal in this system.

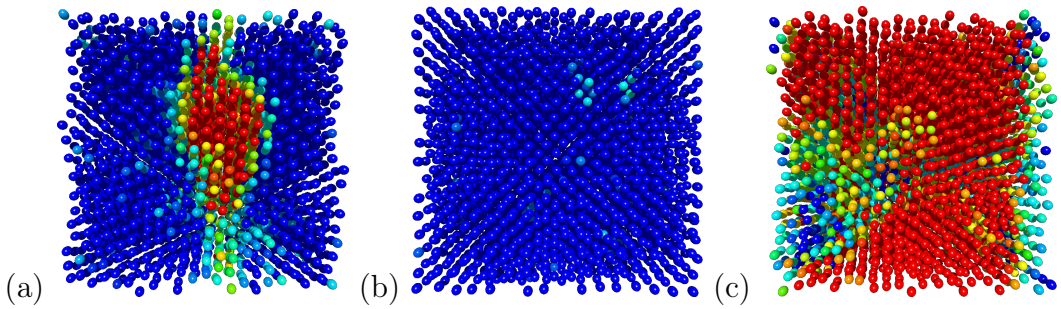

Figure 6.11: Results of different approaches to transform the system with $\epsilon = 20$ to the hcp/fcc-like state, obtained via FFS simulations and the $\overline{q}_4$ order parameter. The color coding is blue for bcc-like particles, red for fcc-like particles and in between for hcp-like particles. We show the states which could be reached if using as starting point (a) the (defective) bcc-like snapshots from the previous $\overline{q}_6$ simulation run, (b) a perfect bcc crystal, and (c) the snapshots of the previous simulation containing the critical clusters. As one can see, approach (c) leads to the desired result.

(c) Use the snapshots from the previous simulation which contain the critical clusters for a new FFS simulation and try if an fcc-like pathway is possible. In our case, these are the snapshots at λ_{n-1}.

Fig. 6.11(a)-(c) gives an overview of the results of these approaches. In approach (a), an hcp/fcc-like domain of a certain size could be grown in the bcc-like phase. However, the FFS simulation was not able to advance the system to higher fcc-like particle numbers in available computation time, the trajectories for reordering the particles are very long in this case. In addition, fluctuations are necessary to reorder the structures, and in this case a larger box size may help to achieve this aim. For the second approach (b), which can be seen as an extreme case of (a), it was not possible to grow a cluster with more than $\lambda = 6$ particles in the largest cluster with hcp-like structure ($\overline{q}_4 \geq 0.06$). Most of the runs immediately fall back to the bcc-like phase, which leads to a very high computational effort, e.g. we launched about a million trial runs starting at the interface with $\lambda = 6$, and none of them was able to reach the next interface. Thus, we can conclude that the defects in system (a) were responsible for allowing some hcp/fcc-like particles to grow, already indicating a free energy barrier. In the third approach (c), where we start from the system snapshots of the previous simulation containing the critical clusters, it was possible to advance to a higher number of hcp/fcc-like particles. Fig. 6.11(c) shows a snapshot where approximately 65% of the particles are fcc-like ($\overline{q}_4 \geq 0.1$). In addition, many hcp-like particles ($\overline{q}_4 \geq 0.05$) can be found in the system. If an fcc-like cluster of size $\lambda \approx 1400$ has been reached, the probability to fall back is less than 25%. Fig. 6.12 shows the time series of the $\overline{q}_4$ and $\overline{q}_6$ parameters for the transition bcc-critical $\rightarrow$ hcp/fcc.

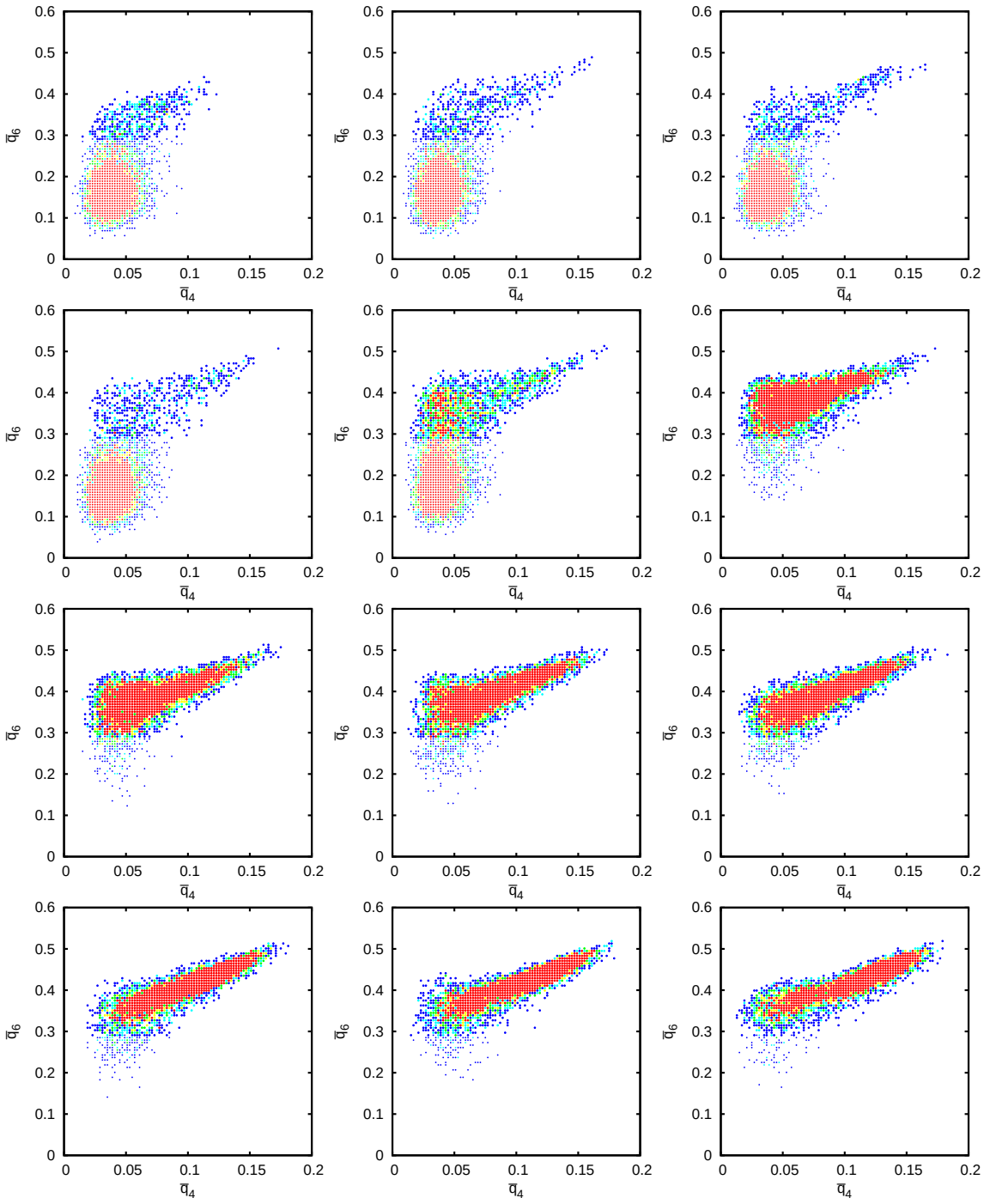

Figure 6.12: Scatter plots of the $\overline{q}_4$ and $\overline{q}_6$ order parameters for the transition from the system states which comprise the critical clusters to an fcc-like state. Thereby, the largest cluster size of particles with $\overline{q}_4 \geq 0.1$ was used as order parameter.

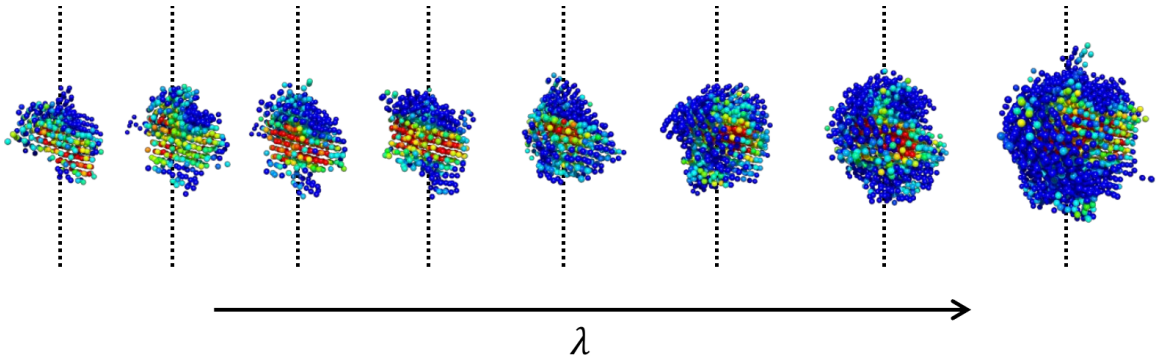

Figure 6.13: Beginning of the transition bcc-critical → hcp/fcc for $\epsilon = 20$ when driving the simulation via $\overline{q}_4$. The first snapshot shown is located on λ_0 of the new simulation. Note, that the core is mainly fcc-structured.

First, most of the particles in the largest cluster are located in the bcc-like and hcp-like domain (compare also with the second to last image of Fig. 6.9, which is from the collection of system states comprising the critical cluster). During the next steps, more particles advance towards the fcc-like structure with $\overline{q}_4 \geq 0.1$, because this transition is driven with our order parameter. In addition, we observe more particles crystallizing to the solid state with $\overline{q}_6 \geq 0.29$. Then, the system transforms further while crystallizing to a similar image like in the case for $\epsilon = 2$ (see also last scattering plot in Fig. 6.6). Fig. 6.13 visualizes the crystal clusters of the beginning of such a trajectory. The first cluster of this trace is located on interface λ_0 of the new simulation. As can be seen, the crystal clusters are optimized for more particles with higher $\overline{q}_4$ values along the nucleation trace.

Before we give an explanation for the observations, we summarize the transitions as follows: The observed pathway for $\epsilon = 2$ is

$$\text{liquid}(\rightarrow \text{bcc}) \rightarrow \text{hcp/fcc}.$$

We put the bcc-like domain in brackets, because this domain is only crossed, and there is no retardation or stay. For $\epsilon = 20$, the spontaneous transition reads

$$\text{liquid} \rightarrow \text{bcc},$$

and the following transitions are not observed:

$$\begin{aligned} \text{liquid} &\nrightarrow \text{hcp/fcc}, \text{ or} \\ \text{bcc} &\nrightarrow \text{hcp/fcc}. \end{aligned}$$

However, using the $\overline{q}_4$ parameter in FFS, it is possible to perform the transition

$$\text{liquid} \rightarrow \text{bcc-critical} \rightarrow \text{hcp/fcc}.$$

Videos of all the transitions are available on the supplementary materials webpage [122].

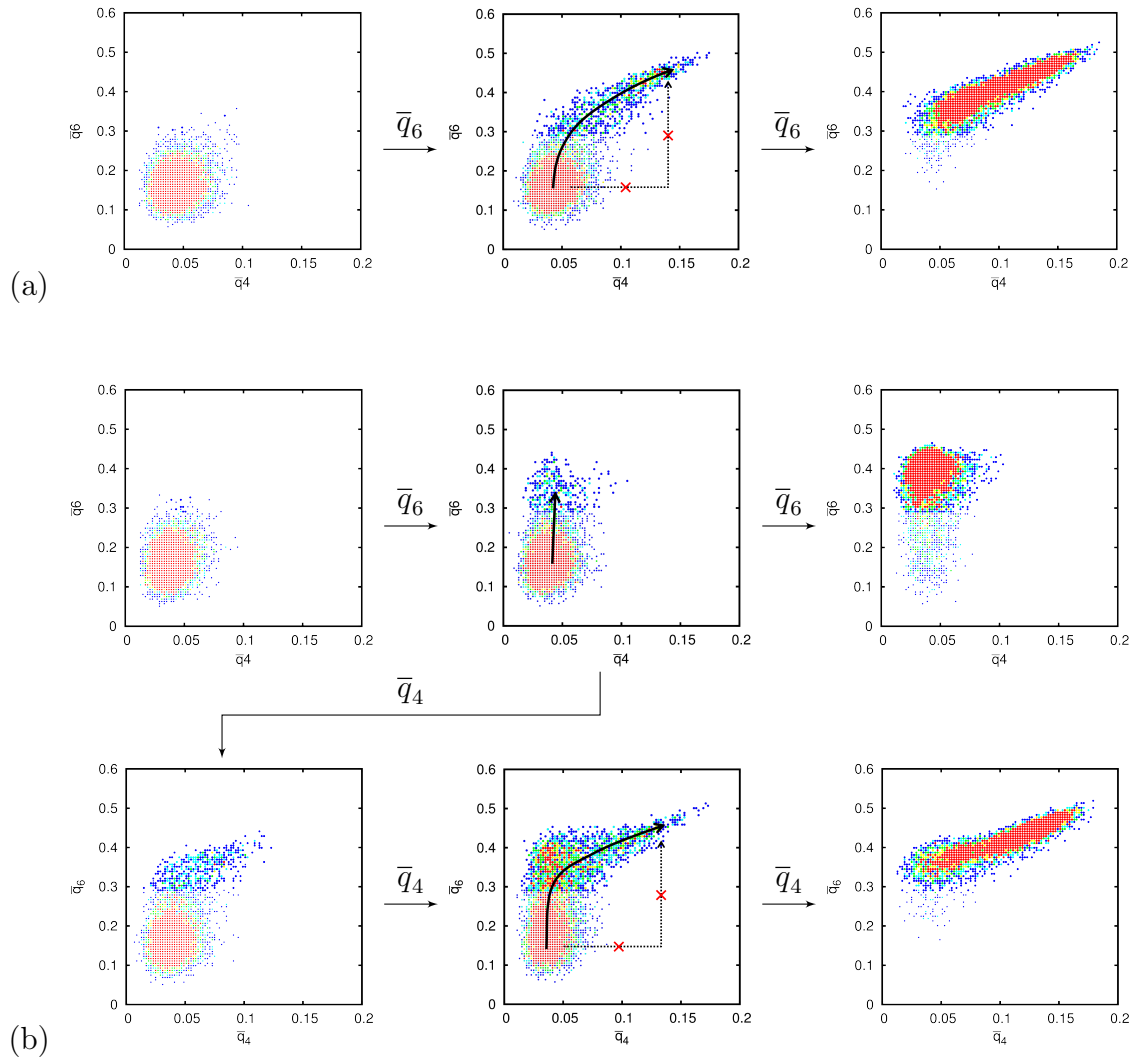

Figure 6.14: Two-stage nucleation: (a) $\epsilon = 2$ with snapshots on λ_A, critical, and λ_B. (b) $\epsilon = 20$ with snapshots on λ_A, critical, λ_B, and the branch of the FFS simulation driven by $\overline{q}_4$. The solid arrows in the figures indicate transition pathways.

6.3.3 Two-stage nucleation

Two-stage nucleation takes place, if the direct transition to the truly stable phase in the parent phase is not possible [70]. Fig. 6.14 summarizes the transitions which we investigated in the previous chapters for $\epsilon = 2$ and $\epsilon = 20$ with the help of the corresponding scattering plots. Fig. 6.14(a) shows selected snapshots of the transition for the $\epsilon = 2$ case, where we use only the $\overline{q}_6$ parameter to drive the transition. Fig. 6.14(b) shows the corresponding transition for the $\epsilon = 20$ case, where the transition is performed

spontaneously to the bcc-like state when only using the $\overline{q}_6$ parameter. In addition, the branch where we use the $\overline{q}_4$ parameter to drive the simulation is shown. In all our simulations we observe, that there is no direct transition liquid $\rightarrow$ fcc (crossed arrows). It was also not possible to drive the simulation directly from the liquid to higher $\overline{q}_4$ order parameters. Hence, there is a *two-stage* nucleation process in all cases:

$$\text{liquid} \;\rightarrow\; \text{bcc} \;\rightarrow\; \text{hcp/fcc}.$$

This behavior is explained naturally with the presence of two energy barriers in Yukawa systems: One for the liquid $\rightarrow$ bcc transition and one for the bcc $\rightarrow$ hcp/fcc transition. The latter one is smaller for lower values of ϵ and can be overcome spontaneously for $\epsilon = 2$. For $\epsilon = 20$ we have to drive the two-stage mechanism in the simulation with the additional help of the $\overline{q}_4$ parameter due to the larger second barrier.

Since there is no direct transition liquid $\rightarrow$ fcc, the liquid-bcc surface in the system is a requirement to overcome the second barrier and for triggering the second transition to the hcp/fcc-like structure, which is a *heterogeneous* nucleation mechanism in the homogeneous system.

6.3.4 Nucleation mechanism

In Sec. 6.3.2 we have seen that the $\overline{q}_6$ parameter is distributed uniformly across the slices of the clusters, with a smooth decay at the border of the cluster (see also Fig. 6.7 and Fig. 6.10). However, the $\overline{q}_4$ parameter shows another behavior. Fig. 6.15 depicts a selection of characteristic distributions of the $\overline{q}_4$ parameter in the system during nucleation of the clusters. The highest values of the $\overline{q}_4$ distributions aren't located in the center of the slices, but fluctuate at the liquid-bcc interface of the cluster, which shows the *heterogeneous* nucleation at the border.

For $\epsilon = 20$ we observe that the fluctuations are smaller, which explains why the system crystallizes to the bcc-like phase, namely because the second energy barrier can't be overcome spontaneously.

To quantify the $\overline{q}_4$ transition bcc-critical $\rightarrow$ hcp/fcc-like, we calculated the transition rate with respect to the surface area of the critical-bcc cluster, and obtain a transition rate of

$$k_{\text{critical-bcc,fcc}} \;=\; 9.52 \times 10^{-08} \tau^{-1} \sigma^{-2}.$$

In summary, two mechanisms play a role for the full transition liquid $\rightarrow$ hcp/fcc: The presence of an already established bcc-like interface, and the necessary fluctuations at the border of the crystal cluster to nucleate a certain structure, and to overcome the second free energy barrier.

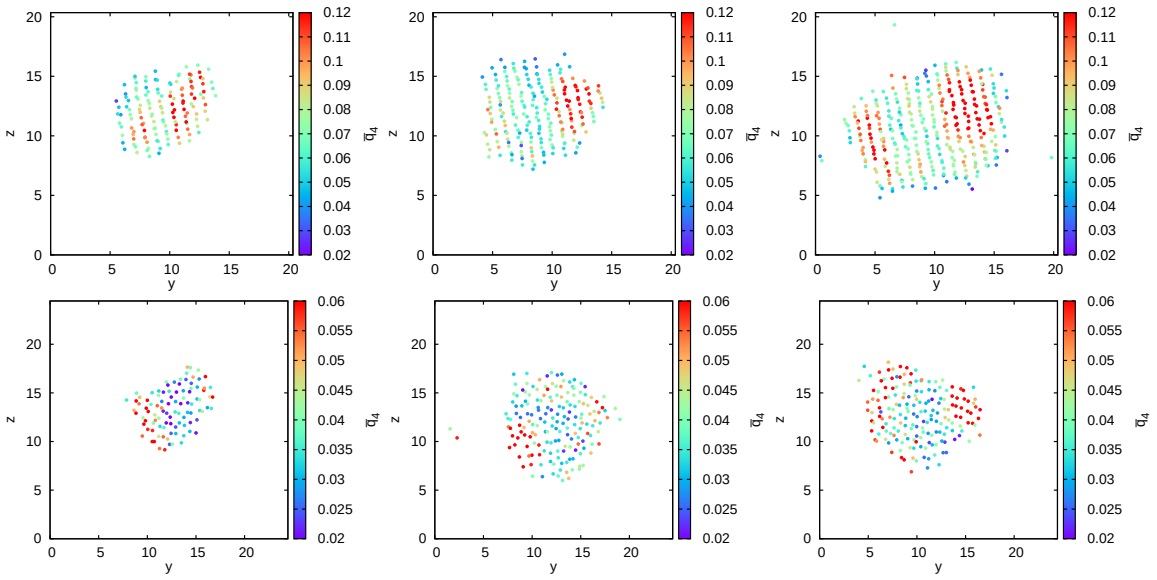

Figure 6.15: Heterogeneous nucleation at the border: $\overline{q}_4$ distributions during nucleation. First row for $\epsilon = 2$ and second row for $\epsilon = 20$. Only solid particles with $\overline{q}_6 \geq 0.29$ are shown in these images. The highest values of the $\overline{q}_4$ parameters occur at the border of the clusters and oscillate during nucleation. Note, that the color scale is different for each row, the fluctuations for $\epsilon = 20$ are much smaller.

6.3.5 Precursors

An interesting question is, if precursors in the liquid phase at an early stage can be detected, which indicate the local onset of crystal nucleation. Fig. 6.16 shows a characteristic set of snapshots at λ_A, obtained by backtracking the successful FFS trajectories. Note, that these snapshots are an outcome of the initial brute-force simulation run, which was started at a random configuration of state A, and was performed without driving the system using an order parameter.

As can be seen, there are only a few solid particles present in the examined slices, and no seeds apart from the one leading to the critical cluster. The pressure at which we simulate is very low, therefore we observe only sporadic formations of solid clusters due to fluctuations. The $\overline{q}_4$ order parameter shows a few smaller regions at random positions where the values are slightly elevated. However, the $\overline{q}_6$ order parameter shows a domain of higher values already in the center of the box, where the critical cluster will nucleate, in all slices. The density maps in the last row do not show any correlations. In contrast to previous work, where the crystallization was studied experimentally at higher supersaturations [13], the density panels in the last row do not show any indications where the critical cluster could form. The distances of the particles are distributed randomly over the whole domain.

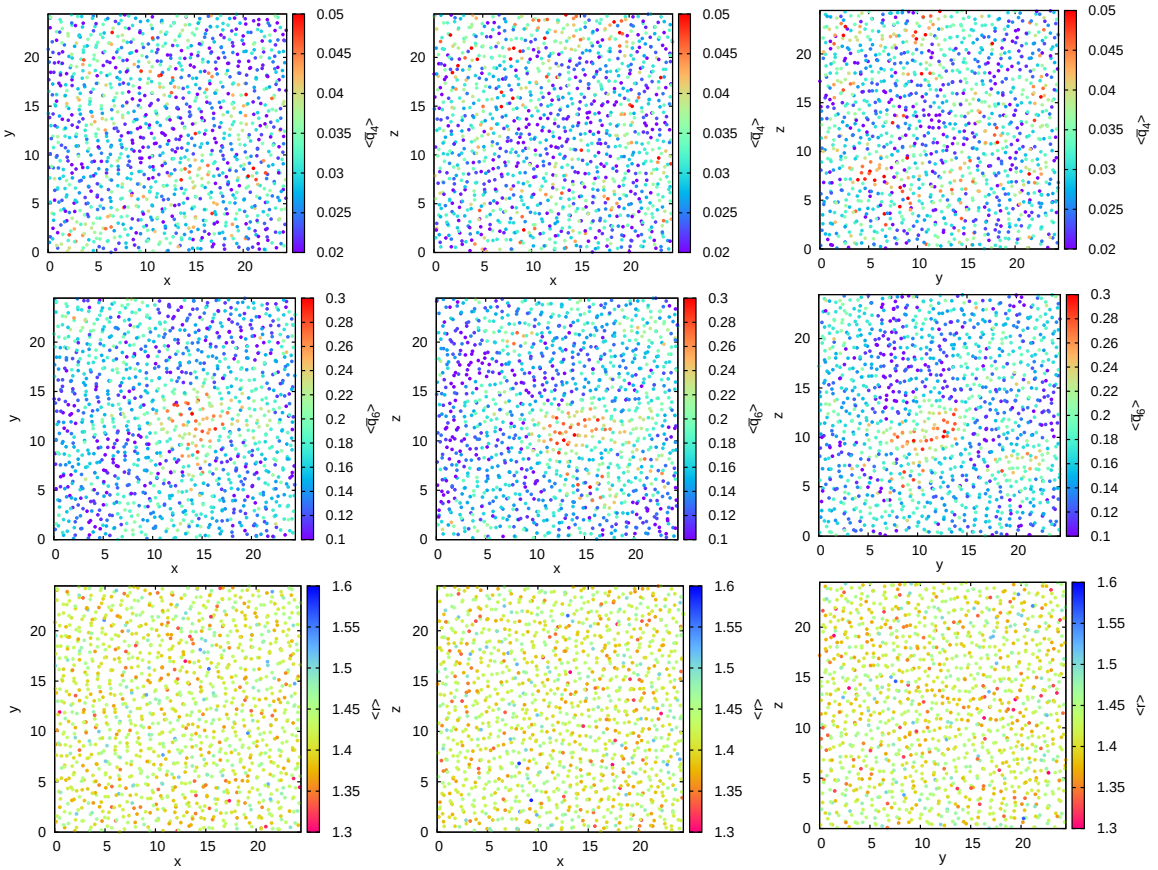

Figure 6.16: Slices of the seed from which the critical cluster of Fig. 6.10 has been grown. The snapshots are transformed according to the center of mass of the critical cluster. Note, that the scales of the $\overline{q}_4$ and $\overline{q}_6$ plots are adjusted to the lower values. The $\overline{q}_6$ distribution (row in the middle) is the quantity of the system which shows a distinct indication where the crystal nucleates.

Thus, the $\overline{q}_6$ distribution is the quantity of the system which shows a distinct indication at this early stage where the critical cluster will form and can be seen as a precursor in this case. As the λ_A border is from the initial brute-force simulation run, this is not an artifact from the FFS simulation being driven by the $\overline{q}_6$ and shows a posteriori that this was a good choice.

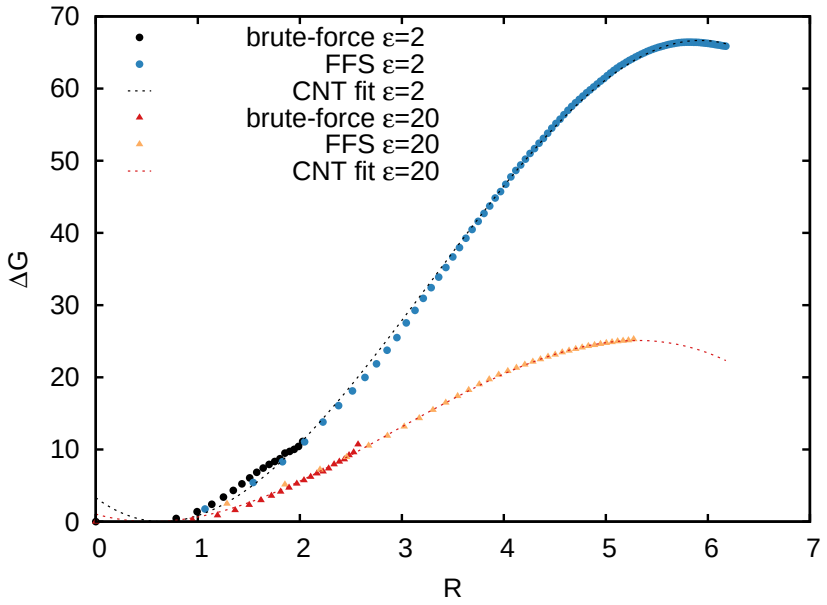

Figure 6.17: ΔG for $\epsilon = 2$ and $\epsilon = 20$, obtained from FFS simulations with $\overline{q}_6 \geq 0.29$ for the identification of solid particles. We allow a shift of the theory curve in R direction because of the unknown real cluster size, which is dependent on the $\overline{q}_6$ threshold.

6.3.6 Comparison to Classical Nucleation Theory

For the comparison with CNT, we calculated the free energy difference $\Delta G(\lambda)$ for the crystallization simulations using the method of calculating stationary distributions with FFS (see also Sec. 2.5.3). The first part of the distribution was obtained using brute-force simulations which has been combined with the distribution of the FFS simulation via a least-square fit like described in Ref. [43].

The resulting energy landscapes $\Delta G(R)$ for $\epsilon = 2$ and $\epsilon = 20$ are shown in Fig. 6.17. Note, that for $\epsilon = 20$ the energy landscape corresponds to the first transition liquid $\rightarrow$ bcc. The simulations have been performed with a threshold of $\overline{q}_6 \geq 0.29$ for the identification of solid particles. Due to the unknown real cluster size we allow a shift in R direction of the theory curve with an offset ΔR_f. The shifts for $\epsilon = 2$ and $\epsilon = 20$ are 0.65 and 0.54, respectively, which is less than a particle diameter. As we can see, it is possible by applying such a shift to fit the free energy profile ΔG of the Classical Nucleation Theory to the free energy profile obtained from the simulations via stationary distributions.

Table 6.2 gives an overview of the FFS simulation and fitting results. From the fitting, we obtain for $\epsilon = 2$ an effective critical cluster size of $R^* = 5.92$, a chemical potential difference of $\Delta\mu = -0.2148$ and a surface tension of $\gamma = 0.5723$, and for $\epsilon = 20$ an effective critical cluster size of $R^* = 5.44$, a chemical potential difference of $\Delta\mu =$

ϵ	λ^*	λ_{n-2}	λ_{n-1}	R^*	ΔG^*	ΔR_f	$\Delta\mu$	γ
2	846	865	1004	5.92	67	0.65	−0.2148	0.5723
20	379	318	363	5.44	25	0.54	−0.1953	0.2637

Table 6.2: Stationary distribution FFS simulations: CNT fitting results.

-0.1953 and a surface tension of $\gamma = 0.2637$. The values of the chemical potential are comparable to the values of Ref. [9], however, deviations can occur because we simulate at slightly different volume fractions.

The automatically placed interfaces λ_{n-2} and λ_{n-1} coincide with the critical cluster size λ^*, which confirms that the FFS interface placement methods with $p_{\text{des}} = 0.5$, which corresponds to the committor value at λ^*, can be used to estimate the size of the critical cluster λ^*.

7 Conclusions

In this thesis we used advanced rare event sampling techniques to investigate the crystallization of charged macromolecules close to phase equilibrium.

To this aim, we implemented a powerful local bond order parameter and a cluster analysis algorithm to the simulation software ESPResSo (chapter 2). Using this combination, the progress of crystallization can be characterized by mapping the system state in phase space to a one-dimensional order parameter, namely the size of the largest crystalline cluster in the system.

We used the Yukawa model as simulation model for the charged macromolecules, which consists of a screened Coulomb potential for the pair interaction of the macromolecules. We tested the applicability of this model against experimental colloidal systems of Polystyrene spheres at different densities in collaboration with the Bechinger group (PI2) at the University of Stuttgart (chapter 5) and find with the help of the Inverse Boltzmann method, that the Yukawa model can in general be used to explain the interactions of the charged colloidal particles. Difficulties only occurred when advancing to high densities of the experimental system, because the radial distribution functions were sharply peaked for these densities, which can be caused by many-body interactions in the system. In this case, a potential which has also an attractive part is better suited than the Yukawa potential to model the sharp peaks of the RDF. However, the important shape of the RDF at smaller radii could be reproduced by the Yukawa potential to a certain extent. For the screening lengths of the colloids we found a dependency on the density of the system, which occurs because of the different number of dissociated charges for a different number of colloids per volume, which we compared to DLVO calculations.

To investigate the nucleation process at an early stage, we performed crystal growth simulations directly from the homogeneous liquid phase, to our knowledge closer to phase equilibrium than it has been done so far for a system of charged macromolecules. Until now, studies at these conditions haven't been possible with conventional brute-force simulations due to the high energy barrier towards nucleation and hence the long waiting time for the assembly of the crystal cluster from the bulk without an artificial seed or impurity. We used rare event sampling techniques, more precisely the Forward Flux Sampling technique, to overcome the energy barrier and to grow the single crystal cluster in the system only with local fluctuations.

The charged macromolecule simulation problem is computationally more expensive than e.g. a hard sphere system or a Lennard-Jones system, therefore we had to put effort in not only parallelizing, but also enhancing the efficiency of the method, where

the latter one depends sensitively on the interface locations of FFS. Based on our analytical preliminary considerations, we introduced two algorithms for placing the interfaces at their optimal positions, automatically and on-the-fly during the simulation. Therefore, the interface locations are estimated by a small number of trial runs.

This improvement increases the efficiency and the set-up of the simulation tremendously, because now, only initial and final state must be specified in terms of the order parameter and in the following the simulated system performs the transition through phase space automatically and optimized, saving a lot of simulation time.

In principle, these placement methods can also be applied to other rare event sampling methods which are based on advancing the system in terms of an interface-based order parameter, as well as be extended to systems with more than one order parameter. Beyond that, the placement methods could also be used to adjust the reaction coordinate on-the-fly, as the simulation progresses.

We tested both, the parallelization of FFS and the enhancement of the efficiency by fundamental simulations and also with the computationally more expensive simulation of the crystallization of charged macromolecules. We could confirm the analytical prediction of the efficiency of the FFS simulation with the help of the fundamental simulations. In the case of the crystallization simulation we could show that with the optimized FFS method the efficiency was significantly larger than without the optimizations, even if we already used pre-knowledge for the simulation setup. Without pre-knowledge, simulating the crystallization of charged macromolecules and especially the placement of the interfaces was a 'hit and miss' task and turned out to be impossible at certain conditions.

We developed the Flexible Rare Event Sampling Harness System which contains all the improvements of the methods and is able to steer many simulation clients in a highly parallel and asynchronous manner (chapter 4). To bridge waiting times which can occur due to the different length of the trajectories, we introduced so-called 'ghost' runs, which are performed if computation resources are available but the system would have to wait for a previously started run to finish. The result of these pre-calculations are stored in a different location and transferred in the next calculation stage, if the random configuration point in FFS is drawn.

In addition, FRESHS is based on a server-client principle and was designed with a module-based structure, which allows the flexible implementation of further sampling algorithms as well as the attachment of different simulation packages and user-defined simulation code with the help of a plug-in system and an appropriate harness script. This creates an objective basis for comparing e.g. the performance of calculating trajectories with different simulation tools, or the applicability of different sampling algorithms. The further development of FRESHS was performed in close collaboration with the group of T. Schilling and J.T. Berryman at the University of Luxembourg. At present, FRESHS is suited for simulating quasi-static and dynamic rare event systems in equilibrium and non-equilibrium and can be used with GROMACS,

LAMMPS, ESPResSo and self-written code. FRESHS uses standard networking for the communication between the sampling method and the simulation of the physics, which makes it possible to use even heterogeneous hardware resources at different high performance computing locations. In the post-processing, powerful analysis tools can be used to analyze the rare event sampling simulation. FRESHS is open-source and publicly available, making the investigation of different rare events accessible for many research groups.

Having the powerful, optimized and parallel tools to simulate rare events, we addressed the crystallization of charged macromolecules at low supersaturations in great detail (chapter 6). For our simulations we used the efficient implementation of FFS with the automatic, optimized interface placement and ghost runs in FRESHS together with the software package ESPResSo on the client side for the calculation of trajectory fragments. We simulated the macromolecules in an NPT ensemble with Langevin dynamics and used a 3D simulation box with periodic boundary conditions. The initial state was set up by randomly placing and equilibrating the particles under the particular conditions which lead to the fluid phase, and the final state was in most cases defined in terms of the order parameter such that approximately more than 90% of the particles are member of the largest cluster of solid particles in the system. Thereby, a particle was defined as solid-like depending on the spatial orientation of the neighboring particles, quantified via the local bond order parameters. The crystal lattice was mainly analyzed in the post-processing.

By using such a system setup, we were able to simulate at low supersaturations and to grow a single cluster in the system. Thereby, the interfaces positions in FFS were determined by the placement methods, which leads to the situation, that the last interface was placed at a location near the critical cluster size. From the critical cluster size on, no further interface was necessary for the last step of the transition to the final state, because growth then happens spontaneously.

We were able to investigate transition rates under different conditions which wouldn't have been possible with conventional brute-force simulations. Thereby we find, that the transition rate decreases drastically with decreasing pressure due to lower volume fractions and a lower supersaturation of the system, which leads to a higher free energy barrier towards nucleation. For lower contact energies and higher pressures we obtained more likely an hcp/fcc-like structure than for higher contact energies and lower pressures. To analyze this behavior, we investigated also the crystallization pathways under these conditions. For low contact energies and a certain pressure we obtained a spontaneous transition to the hcp/fcc-like state of the system. We analyzed the structure formation and the crystal clusters for this transition in great detail and find crystal clusters of nearly spherical shape during the nucleation process.

For high contact energies at a comparable pressure we obtained a direct transition to a bcc-like metastable state of the system. As the stable state at this phase point is also the fcc-like domain, we used multiple approaches to drive the system towards this stable state. Thereby, it was not possible to drive the system from a fully converted

bcc-like system towards the hcp/fcc-like state. However, it was possible by starting from the system states comprising the critical clusters to optimize for a larger number of fcc-like particles using our FFS simulations. This could be explained by the presence of a liquid-bcc interface, which supports the heterogeneous nucleation of the hcp/fcc-like structure at the surface of the crystal. These investigations lead to the question, if the crystallization is a two-stage mechanism or not.

In all our simulations, we didn't see a direct transition from liquid to the fcc-like domain without crossing the bcc-like and hcp-like domains, and it wasn't possible to crystallize the system, at least in available computation time, by just optimizing for an fcc-like cluster in the system. Using the structure analysis, the crystallization pathways could be analyzed, which always took the transition liquid $\rightarrow$ bcc $\rightarrow$ hcp/fcc in our investigations, which is a two-stage process. This means, that there are two energy barriers towards the stable state, and the second one for the transition bcc $\rightarrow$ hcp/fcc is smaller for lower contact energies than for higher values, which leads to the result, that for the low contact value the barrier can be overcome spontaneously, and for the high contact value we have to drive the system over the second barrier using an extra FFS simulation. In addition, the already existing bcc-like interface is a necessary condition for the further nucleation of hcp-like and fcc-like particles at the crystal surface. To quantify this process, we calculated the transition rate of the second transition with respect to the surface area of the critical cluster. We further analyzed the growth mechanisms and found, that fluctuations of the hcp/fcc-like particles take mainly place at the surface of the crystal cluster, and that these fluctuations are much stronger for lower contact energies, which also helps to overcome the second energy barrier. The final crystal structure is then nucleated starting from these fluctuations at the border.

To identify precursors, we investigated the nucleation at an early stage, namely at the border of the initial state A, where the crystal seed is born from the parent phase. Therefore, we analyzed the configuration of the liquid at the local positions where the critical cluster will form at a later stage. For both, low and high contact energies, we find correlations of the $\overline{q}_6$ structure at the appropriate positions. Hence, the formation of a crystal cluster is indicated by local sixfold symmetry at this early stage. We do not observe correlations with the local density or fourfold symmetry in our simulations. The nucleation is started by spontaneous local fluctuations which lead to a bcc-like ordering, as predicted by Alexander and McTague.

For comparing the simulations to the Classical Nucleation Theory, we computed the stationary distributions and the free energy landscape ΔG with a forward and backward FFS simulation, where in the latter case the crystal cluster was dissolved again. The curve of the Classical Nucleation Theory was fitted to the obtained free energy landscape, assuming an effective spherical cluster shape and allowing a shift in radial direction due to the unknown real cluster size, which is influenced by the order parameter threshold. The radial offset was found to be less than a particle diameter, which confirms that the choice of the $\overline{q}_6$ threshold of 0.29 was suitable to describe

the nucleation process of solid particles. From the fitting results the surface tension and chemical potential difference for the particular simulation could be extracted. The chemical potential values are comparable to already known values from different calculations of previous work. Thus, CNT seems to be a surprisingly good model for crystal nucleation. However, the relevant phase for nucleation is not the thermodynamically most stable one, but rather the first formed metastable phase.

In conclusion, the findings of this work on colloidal crystallization contribute directly to a closed theory of colloidal crystallization: Unlike previously argued, the free energy landscapes are well-described by classical nucleation theory, where the relevant transition is the one to the first metastable phase. This phase is always a bcc crystal, even if the thermodynamically stable phase is an fcc crystal. In this case, nucleation is a two-stage process, which however does not influence nucleation rates or the structure of the critical cluster. The nuclei are almost spherical, so that edges of the crystal play a minor role. Also, the crystal surface is fairly diffuse, which however is taken into account by the surface tension. Nucleation is mainly the formation of a sixfold symmetry, which can already be seen at the onset of crystallization in the supersaturated liquid.

Beyond that, the advanced rare event sampling techniques developed during this thesis help other researchers to investigate rare events in diverse fields like biology, chemistry, medicine and physics.

Acknowledgements

I would like to thank my advisor Axel Arnold for the possibility to work in the field of soft matter physics and rare event sampling and for the great supervision during this time. Without his scientific knowledge and advice, this thesis would not have been possible. I also thank Hans-Rainer Trebin and Frank Noé for being the co-referees of this thesis.

I am very grateful to Rosalind Allen for hosting me as a guest researcher in her group at the University of Edinburgh. Some of the results of this thesis are based on the scientific outcome of this stay. Special thanks go to Josh Berryman and Tanja Schilling for the cooperation and development concerning the Flexible Rare Event Sampling Harness System (FRESHS). For providing us experimental data which could be compared to our simulation model and for the cooperation, I thank Lamiss Zaidouny and Clemens Bechinger from the PI2, University of Stuttgart. I would like to thank Yevgen Dorozhko, Yuriy Yudin, Colin W. Glass and Michael Resch from the HLRS for the cooperation concerning the Science Experimental Grid Laboratory.

Many thanks to Aaron Taudt and Johannes Zeman who contributed with very helpful work during their theses to FRESHS. I would like to thank the colleagues from the Institute for Computational Physics for the help and advice during my enjoyable time at the institute. In addition, special thanks to the members of the advanced rare event sampling group for the helpful discussions.

I would like to thank the SimTech staff for the organization as well as the SimTech PhD colleagues for having a great time, not only during the PhD weekends but also in the Graduate School seminars.

Thanks go to the HLRS, BW-Grid and the SimTech cluster for providing computational resources. For financial support of the project within the Cluster of Excellence in Simulation Technology (EXC 310/1) at the University of Stuttgart I would like to thank the German Research Foundation (DFG).

Last but not least, I would like to thank my friends and the Tuesday regulars' lunch table, the longest praline of the world, and my family for the great encouragement and support in countless ways during this time.

List of Figures

List of Tables

Bibliography

[1] Denis Gebauer. Wie bilden sich Kristalle? *Nachrichten aus der Chemie*, 61(11):1097–1100, 2013.

[2] G. Subramania, K. Constant, R. Biswas, M. M. Sigalas, and K.-M. Ho. Optical photonic crystals fabricated from colloidal systems. *Applied Physics Letters*, 74(26):3933–3935, 1999.

[3] Vicki L. Colvin. From opals to optics: Colloidal photonic crystals. *MRS Bulletin*, 26:637–641, 8 2001.

[4] Robert K. Scopes. *Protein purification: principles and practice*. Springer, New York, 3. ed. edition, 1994.

[5] S. D. Durbin and G. Feher. Protein crystallization. *Annual Review of Physical Chemistry*, 47(1):171–204, 1996.

[6] Naomi E Chayen and Emmanuel Saridakis. Protein crystallization: from purified protein to diffraction-quality crystal. *Nature Methods*, 5:147 – 153, 2008.

[7] Alexander McPherson. Review current approaches to macromolecular crystallization. In P. Christen and E. Hofmann, editors, *EJB Reviews 1990*, volume 1990 of *European Journal of Biochemistry*, pages 49–71. Springer Berlin Heidelberg, 1991.

[8] Juan Manuel Garcia-Ruiz. Nucleation of protein crystals. *Journal of Structural Biology*, 142(1):22 – 31, 2003. Macromolecular crystallization in the structural genomics era.

[9] Stefan Auer and Daan Frenkel. Crystallization of weakly charged colloidal spheres: a numerical study. *Journal of Physics: Condensed Matter*, 14(33):7667, 2002.

[10] J. R. Savage and A. D. Dinsmore. Experimental evidence for two-step nucleation in colloidal crystallization. *Phys. Rev. Lett.*, 102:198302, May 2009.

[11] Takeshi Kawasaki and Hajime Tanaka. Formation of a crystal nucleus from liquid. *Proceedings of the National Academy of Sciences*, 107(32):14036–14041, 2010.

[12] Kipton Barros and W. Klein. Liquid to solid nucleation via onion structure droplets. *J. Chem. Phys.*, 139(17):–, 2013.

[13] Peng Tan, Ning Xu, and Lei Xu. Visualizing kinetic pathways of homogeneous nucleation in colloidal crystallization. *Nature Physics*, 10:73 – 79, 2014.

[14] Laszlo Granasy and Gyula I. Toth. Crystallization: Colloidal suspense. *Nature Physics*, 10:12 – 14, 2014.

[15] A. McPherson. Introduction to protein crystallization. *Methods*, 34(3):254–265, 2004.

[16] Richard P Sear. Nucleation: theory and applications to protein solutions and colloidal suspensions. *Journal of Physics: Condensed Matter*, 19(3):033101 (28pp), 2007.

[17] Hans-Jörg Limbach, A. Arnold, Bernward A. Mann, and Christian Holm. Espresso – an extensible simulation package for research on soft matter systems. *Comput. Phys. Commun. 174*, 9:704–727, 2006.

[18] G. M. Torrie and J. P. Valleau. Monte-Carlo free-energy estimates using non-Boltzmann sampling - application to subcritical Lennard-Jones fluid. *Chem. Phys. Lett.*, 28:578, 1974.

[19] D. Frenkel and B. Smit. *Understanding Molecular Simulation. From Algorithms to Applications.* Academic Press, Boston, second edition, 2002.

[20] D. Chandler. Statistical mechanics of isomerization dynamics in liquids and transition state approximation. *J. Chem. Phys.*, 68:2959–2970, 1978.

[21] C. H. Bennett. Algorithms for chemical computations. In R.Christofferson, editor, *ACS Symposium, Series No.46*, Washington, D.C., 1977. American Chemical Society.

[22] C. Dellago, P. G. Bolhuis, F. S. Csajka, and D. Chandler. Transition path sampling and the calculation of rate constants. *J. Chem. Phys.*, 108:1964–1977, 1998.

[23] C. Dellago, P. G. Bolhuis, and P. L. Geissler. Transition path sampling. *Adv. Chem. Phys.*, 123:1–78, 2002.

[24] P. G. Bolhuis, D. Chandler, C. Dellago, and P. L. Geissler. Transition path sampling: Throwing ropes over rough mountain passes, in the dark. *Annu. Rev. Phys. Chem.*, 53:291–318, 2002.

[25] P. G. Bolhuis. Transition path sampling on diffusive barriers. *J. Phys.: Condens. Matter*, 15:S113–S120, 2003.

[26] J. Rogal and P. G. Bolhuis. Multiple state transition path sampling. *J. Chem. Phys.*, 129:224107, 2008.

[27] T. S. van Erp, D. Moroni, and P. G. Bolhuis. A novel path sampling method for the calculation of rate constants. *J. Chem. Phys.*, 118:7762, 2003.

[28] T. S. van Erp and P. G. Bolhuis. Elaborating Transition Interface Sampling methods. *J. Comp. Phys.*, 205:157–181, 2005.

[29] T. S. van Erp. Efficient path sampling on multiple reaction channels. *Comp. Phys. Commun.*, 179:34–40, 2008.

[30] A. K. Faradjian and R. Elber. Computing time scales from reaction coordinates by milestoning. *J. Chem. Phys.*, 120:10880–10889, 2004.

[31] A. M. A. West, R. Elber, and D. Shalloway. Extending molecular dynamics timescales with milestoning: Example of complex kinetics in a solvated peptide. *J. Chem. Phys.*, 126:145104, 2007.

[32] E. Vanden-Eijnden, M. Venturoli, G. Ciccotti, and R. Elber. On the assumptions underlying milestoning. *J. Chem. Phys.*, 129:174102, 2008.

[33] G. Henkelman, B. P. Uberuaga, and H. Jonsson. A climbing image nudged elastic band method for finding saddle points and minimum energy paths. *J. Chem. Phys.*, 113:9901–9904, 2000.

[34] G. Henkelman and H. Jonsson. Improved tangent estimate in the nudged elastic band method for finding minimum energy paths and saddle points. *J. Chem. Phys.*, 113:9978–9985, 2000.

[35] W. E, W. Ren, and E. Vanden-Eijnden. String method for the study of rare events. *Phys. Rev. B*, 66:052301, 2002.

[36] W. E, W. Ren, and E. Vanden-Eijnden. Finite temperature string method for the study of rare events. *J. Phys. Chem. B*, 109:6688, 2005.

[37] G. A. Huber and S. Kim. Weighted-ensemble Brownian dynamics simulations for protein association reactions. *Biophys. J.*, 70:97–110, 1996.

[38] H. Kahn and T.E. Harris. Estimation of particle transmission by random sampling. *National Bureau of Standards Applied Mathematics Series*, 12:27–30, 1951.

[39] R. J. Allen, P. B. Warren, and P. R. ten Wolde. Sampling rare switching events in biochemical networks. *Phys. Rev. Lett.*, 94:018104, 2005.

[40] R. J. Allen, D. Frenkel, and P. R. ten Wolde. Simulating rare events in equilibrium or nonequilibrium stochastic systems. *J. Chem. Phys.*, 124:024102, 2006.

[41] Rosalind J. Allen, Daan Frenkel, and Pieter Rein ten Wolde. Forward flux sampling-type schemes for simulating rare events: Efficiency analysis. *J. Chem. Phys.*, 124(19):194111, 2006.

[42] Rosalind J Allen, Chantal Valeriani, and Pieter Rein ten Wolde. Forward flux sampling for rare event simulations. *Journal of Physics: Condensed Matter*, 21(46):463102, 2009.

[43] C. Valeriani, R. J. Allen, M. J. Morelli, D. Frenkel, and P. R. ten Wolde. Computing stationary distributions in equilibrium and nonequilibrium systems with forward flux sampling. *J. Chem. Phys.*, 127:114109, 2007.

[44] E. E. Borrero and F. A. Escobedo. Simulating the kinetics and thermodynamics of transitions via forward flux / umbrella sampling. *J. Phys. Chem. B*, 113:6434–6445, 2009.

[45] E. E. Borrero and F. A. Escobedo. Reaction coordinates and transition pathways of rare events via forward flux sampling. *J. Chem. Phys.*, 127:164101, 2007.

[46] A. Dickson and A. R. Dinner. Enhanced sampling of nonequilibrium steady states. *Annu. Rev. Phys. Chem.*, 61:441–459, 2010.

[47] A. Warmflash, P. Bhimalapuram, and A.R. Dinner. Umbrella sampling for nonequilibrium processes. *J. Chem. Phys.*, 127:154112, 2007.

[48] F. A. Escobedo, E. E. Borrero, and J. C. Araque. Transition path sampling and forward flux sampling. applications to biological systems. *Journal of Physics: Condensed Matter*, 21(33):333101, 2009.

[49] Manuel Villén-Altamirano and José Villén-Altamirano. Restart: a straightforward method for fast simulation of rare events. In *Proceedings of the 26th conference on Winter simulation*, WSC '94, pages 282–289, San Diego, CA, USA, 1994. Society for Computer Simulation International.

[50] Frédéric Cérou and Arnaud Guyader. Adaptive multilevel splitting for rare event analysis. Rapport de recherche RR-5710, INRIA, 2005.

[51] F. Kuypers. *Klassische Mechanik: mit über 300 Beispielen und Aufgaben mit Lösungen*. Lehrbuch Physik. John Wiley & Sons, Limited, 2008.

[52] Hans C Andersen. Molecular dynamics simulations at constant pressure and/or temperature. *J. Chem. Phys.*, 72(4):2384–2393, 1980.

[53] Glenn J. Martyna, Douglas J. Tobias, and Michael L. Klein. Constant pressure molecular dynamics algorithms. *The Journal of Chemical Physics*, 101(5):4177–4189, 1994.

[54] Scott E. Feller, Yuhong Zhang, Richard W. Pastor, and Bernard R. Brooks. Constant pressure molecular dynamics simulation: The langevin piston method. *The Journal of Chemical Physics*, 103(11):4613–4621, 1995.

[55] A Kolb and B Dünweg. Optimized constant pressure stochastic dynamics. *J. Chem. Phys.*, 111(10):4453–4459, 1999.

[56] Alexander P. Lyubartsev and Aatto Laaksonen. Calculation of effective interaction potentials from radial distribution functions: A reverse monte carlo approach. *Phys. Rev. E*, 52:3730–3737, Oct 1995.

[57] Dirk Reith, Mathias Pütz, and Florian Müller-Plathe. Deriving effective mesoscale potentials from atomistic simulations. *Journal of Computational Chemistry*, 24(13):1624–1636, 2003.

[58] Jacob N. Israelachvili. *Intermolecular and surface forces*. Academic Press, London [u.a.], 2. ed. edition, 1992.

[59] William Bailey Russel, Dudley Albert Saville, and William Raymond Schowalter. *Colloidal dispersions*. Cambridge university press, 1992.

[60] Phil Attard. Electrolytes and the electric double layer. *Advances in Chemical Physics*, 92:1–160, 1996.

[61] G. Dupont, S. Moulinasse, J.P. Ryckaert, and M. Baus. The b.c.c.-f.c.c.-fluid triple point as obtained from monte carlo simulations of the yukawa model for monodisperse colloidal suspensions. *Molecular Physics*, 79(2):453–456, 1993.

[62] Fouad El Azhar, Marc Baus, Jean-Paul Ryckaert, and Evert Jan Meijer. Line of triple points for the hard-core yukawa model: A computer simulation study. *J. Chem. Phys.*, 112(11):5121–5126, 2000.

[63] Gerhard Nägele. On the dynamics and structure of charge-stabilized suspensions. *Physics Reports*, 272(5-6):215 – 372, 1996.

[64] J. D. Weeks, D. Chandler, and H. C. Andersen. Role of repulsive forces in determining the equilibrium structure of simple liquids. *J. Chem. Phys.*, 54:5237, 1971.

[65] Gordon E Moore et al. Cramming more components onto integrated circuits, 1965.

[66] Evert Johannes Willem Verwey, J Th G Overbeek, and Jan Theodoor Gerard Overbeek. *Theory of the stability of lyophobic colloids.* Courier Dover Publications, 1999.

[67] S Alexander, PM Chaikin, P Grant, GJ Morales, P Pincus, and D Hone. Charge renormalization, osmotic pressure, and bulk modulus of colloidal crystals: Theory. *J. Chem. Phys.*, 80(11):5776–5781, 1984.

[68] Evert Jan Meijer and Daan Frenkel. Melting line of yukawa system by computer simulation. *J. Chem. Phys.*, 94(3):2269–2271, 1991.

[69] F. Bitzer, T. Palberg, H. Löwen, R. Simon, and P. Leiderer. Dynamical test of interaction potentials for colloidal suspensions. *Phys. Rev. E*, 50:2821–2826, Oct 1994.

[70] Dimo Kashchiev. *Nucleation: basic theory with applications.* Butterworth-Heinemann, Oxford [u.a.], 1. publ. edition, 2000.

[71] Ivan V. Markov. *Crystal Growth for Beginners - Fundamentals of Nucleation, Crystal Growth and Epitaxy.* World Scientific Publishing Co. Pte. Ltd., 2003.

[72] J Willard Gibbs. *The Collected Works of J. Willard Gibbs, Volume I: Thermodynamics.* Yale University Press, 1928.

[73] M. Volmer and A. Weber. Keimbildung in übersättigten gebilden. *Z. Phys. Chem*, 119:227, 1926.

[74] L Farkas. Keimbildungsgeschwindigkeit in übersättigten dämpfen. *Z. phys. Chem*, 125:236–242, 1927.

[75] R. Becker and W. Döring. Kinetische behandlung der keimbildung in übersättigten dämpfen. *Annalen der Physik*, 416(8):719–752, 1935.

[76] Ostwald. Studien über die bildung und umwandlung fester körper. *Z. Phys. Chem.*, 22:289, 1897.

[77] Jakov I. Frenkel'. *Kinetic theory of liquids.* Clarendon Press, Oxford, 1947.

[78] Chantal Valeriani. *Numerical studies of nucleation pathways of ordered and disordered phases.* PhD thesis, FOM AMOLF, 2007.

[79] N. G. van Kampen. *Stochastic processes in chemistry and physics.* Elsevier, 2007.

[80] Baron Peters and Bernhardt L Trout. Obtaining reaction coordinates by likelihood maximization. *J. Chem. Phys.*, 125(5):054108, 2006.

[81] Paul J Steinhardt, David R Nelson, and Marco Ronchetti. Bond-orientational order in liquids and glasses. *Physical Review B*, 28(2):784, 1983.

[82] Wolfgang Lechner and Christoph Dellago. Accurate determination of crystal structures based on averaged local bond order parameters. *J. Chem. Phys.*, 129(11):114707, 2008.

[83] Richard Phillips Feynman, Robert B Leighton, and Matthew Sands. *[Lectures on physics]; The Feynman lectures on physics. 2 (1969). Mainly electromagnetism and matter*. Addison-Wesley, 1969.

[84] Richard P Sear. Nucleation in the presence of slow microscopic dynamics. *The Journal of chemical physics*, 128(21):214513, 2008.

[85] Titus S. van Erp, Daniele Moroni, and Peter G. Bolhuis. A novel path sampling method for the calculation of rate constants. *J. Chem. Phys.*, 118:7762, 2003.

[86] Jarosław Juraszek. *Proteins in action: simulations of conformational changes in small proteins*. PhD thesis, Universiteit van Amsterdam, 2008.

[87] 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. *The Journal of Chemical Physics*, 133(24):244115, 2010.

[88] C. Valeriani, E. Sanz, and D. Frenkel. Rate of homogeneous crystal nucleation in molten nacl. *The Journal of Chemical Physics*, 122(19):194501, 2005.

[89] C. Velez-Vega, E. E. Borrero, and F. A. Escobedo. Kinetics and reaction coordinate for the isomerization of alanine dipeptide by a forward flux sampling protocol. *J. Chem. Phys.*, 130:225101, 2009.

[90] Ernesto E Borrero and Fernando A Escobedo. Folding kinetics of a lattice protein via a forward flux sampling approach. *The Journal of chemical physics*, 125(16):164904, 2006.

[91] Ernesto E. Borrero and Fernando A. Escobedo. Optimizing the sampling and staging for simulations of rare events via forward flux sampling schemes. *J. Chem. Phys.*, 129(2):024115, 2008.

[92] E.E. Borrero, M. Weinwurm, and C. Dellago. Optimizing transition interface sampling simulations. *J. Chem. Phys.*, 134:244118, 2011.

[93] Yevgen Dorozhko, Kai Kratzer, Yuriy Yudin, Axel Arnold, Colin W. Glass, and Michael Resch. Rare event sampling using the science experimental grid laboratory. In B.H.V. Topping and P. Iványi, editors, *Proceedings of the Fourteenth*

International Conference on Civil, Structural and Environmental Engineering Computing, page Paper 207, Stirlingshire, UK, 2013. Civil-Comp Press.

[94] Kai Kratzer, Axel Arnold, and Rosalind J. Allen. Automatic, optimized interface placement in forward flux sampling simulations. *J. Chem. Phys.*, 138(16):164112, 2013.

[95] Natalia Currle-Linde, Uwe Küster, Michael Resch, and B. Risio. Science experimental grid laboratory (SEGL) dynamic parameter study in distributed systems. In *PARCO*, pages 49–56, 2005.

[96] Yuriy Yudin, Tatjana Krasikova, Yevgen Dorozhko, Natalia Currle-Linde, and Michael Resch. An efficient workflow system in real HPC organization. In *International Workshop on Science Gateways (IWSG 2010)*, pages 20–22, 2010.

[97] Yuriy Yudin, Tatjana Krasikova, Yevgen Dorozhko, and Natalia Currle-Linde. Modular workflow system for HPC applications. In *International Conference on High Performance Computing (ICHPC 2013)*, 2013.

[98] Kai Kratzer, Joshua T Berryman, Aaron Taudt, Johannes Zeman, and Axel Arnold. The flexible rare event sampling harness system (freshs). *Computer Physics Communications*, 185(7):1875–1885, 2014.

[99] Matthias U Bohner, Johannes Zeman, Jens Smiatek, Axel Arnold, and Johannes Kästner. Nudged-elastic band used to find reaction coordinates based on the free energy. *J. Chem. Phys.*, 140(7):074109, 2014.

[100] Joshua T. Berryman and Tanja Schilling. Sampling rare events in nonequilibrium and nonstationary systems. *J. Chem. Phys.*, 133:244101, 2010.

[101] Nils B. Becker, Rosalind J. Allen, and Pieter Rein ten Wolde. Non-stationary forward flux sampling. *J. Chem. Phys.*, 136(17):174118, 2012.

[102] Kai Kratzer and Joshua T. Berryman. Website of the Flexible Rare Event Sampling Harness System (FRESHS): www.freshs.org.

[103] H.J.C. Berendsen, D. van der Spoel, and R. van Drunen. Gromacs: A message-passing parallel molecular dynamics implementation. *Computer Physics Communications*, 91(1–3):43 – 56, 1995.

[104] S. Plimpton. Fast parallel algorithms for short-range molecular dynamics. *J. Comp. Phys.*, 117:1–19, 1995.

[105] C. Dekker. Solid-state nanopores. *Nature Nanotech*, 2:209–215, 2007.

[106] W. Sung and P. J. Park. Polymer translocation through a pore in a membrane. *Phys. Rev. Lett.*, 77:783–786, Jul 1996.

[107] M. Brunner, C. Bechinger, W. Strepp, V. Lobaskin, and H. H. von Grünberg. Density-dependent pair interactions in 2d. *EPL (Europhysics Letters)*, 58(6):926, 2002.

[108] Thijs JH Vlugt, JPJM Van der Eerden, Marjolein Dijkstra, Berend Smit, and Daan Frenkel. Introduction to molecular simulation and statistical thermodynamics. *Delft, The Netherlands*, 2008.

[109] Kai Kratzer and Axel Arnold. Two-stage crystallization of charged colloids at low supersaturations. *arXiv:1410.8695*, 2014.

[110] Kai Kratzer, Dominic Roehm, and Axel Arnold. *High Performance Computing in Science and Engineering '14*. Springer, 2014.

[111] Peter G Vekilov. Phase transitions of folded proteins. *Soft Matter*, 6(21):5254–5272, 2010.

[112] A Basak Kayitmazer, Daniel Seeman, Burcu Baykal Minsky, Paul L Dubin, and Yisheng Xu. Protein–polyelectrolyte interactions. *Soft Matter*, 9(9):2553–2583, 2013.

[113] Thomas Palberg. Crystallization kinetics of repulsive colloidal spheres. *Journal of Physics: Condensed Matter*, 11(28):R323, 1999.

[114] H-J Schöpe and T Palberg. Crystal nucleation versus vitrification in charged colloidal suspensions. In *Trends in Colloid and Interface Science XV*, pages 82–86. Springer, 2001.

[115] Hans Joachim Schöpe and Thomas Palberg. A study on the homogeneous nucleation kinetics of model charged sphere suspensions. *Journal of Physics: Condensed Matter*, 14(45):11573, 2002.

[116] Patrick Wette and Hans Joachim Schöpe. Nucleation kinetics in deionized charged colloidal model systems: A quantitative study by means of classical nucleation theory. *Physical Review E*, 75(5):051405, 2007.

[117] T Schilling, HJ Schöpe, M Oettel, G Opletal, and I Snook. Precursor-mediated crystallization process in suspensions of hard spheres. *Phys. Rev. Lett.*, 105(2):025701, 2010.

[118] John Russo and Hajime Tanaka. The microscopic pathway to crystallization in supercooled liquids. *Scientific Reports*, 2, 2012.

[119] IN Stranski and D Totomanow. Rate of formation of (crystal) nuclei and the ostwald step rule. *Z. Phys. Chem*, 163:399–408, 1933.

[120] Sh Alexander and J McTague. Should all crystals be bcc? landau theory of solidification and crystal nucleation. *Physical Review Letters*, 41(10):702, 1978.

[121] Antti-Pekka Hynninen and Marjolein Dijkstra. Phase diagrams of hard-core repulsive yukawa particles. *Phys. Rev. E*, 68:021407, Aug 2003.

[122] Kai Kratzer. Crystallization at low supersaturations – supplementary materials webpage: http://www.icp.uni-stuttgart.de/~kratzer/.