Unterschrift Betreuer

TECHNISCHE
UNIVERSITÄT
WIEN
Vienna University of Technology

Dissertation

Self-Assembly of Shape- and Interaction Anisotropic Particles on a Surface

Ausgeführt zum Zwecke der Erlangung des akademischen Grades
eines Doktors der Naturwissenschaften

unter der Anleitung von
Ao.Univ.Prof. DI Dr. Gerhard Kahl
Institut für Theoretische Physik (E136)

durch

DI Susanne Wagner
Matrikelnummer: 1125888

Wien, 30.Juni 2022

Unterschrift Studentin

Abstract

The self-assembly behaviour of colloidal and patchy particles, i.e., colloids with localized surface functionalizations, is central to a rapidly growing field of research in chemistry and physics. Over the last three decades, the surge of attention towards syntheses of these particles is explained by the manifold existing and future technological applications of materials with nano- and micrometre-sized structures.
In computational and theoretical models, hard-core particles have a long history, dating back to the 1950s, when computers started to become an indispensable tool in statistical physics. The patchy particle model was invented in the 1980s for a theoretical description of associated liquids. Nowadays, the explanation of colloidal and patchy particles' self-assembly using modelling and simulations to compute phase behaviour can serve as a predecessor for directing efforts in real-life implementations.

In this PhD-Thesis, I present two research projects, both focusing on ensembles of elliptic particles using Monte Carlo simulations.
In the first project, which is presented in Part II of this thesis, different lattice configurations formed by hard-core elliptic particles are investigated. There are infinitely many possible lattice configurations for these types of particles. My investigations provide an answer to the question of which one of these lattices is the entropically most favourable one. I present an in-depth analysis of a possible lattice parameterization as well as the search for two lattice candidates, namely the parallel and diagonal lattices, for the following free energy computations. By the use of finite-size corrections, I show that the parallel lattice is entropically more favourable than the diagonal lattice state.
In the second project, presented in Part III, the self-assembly scenarios of hard-core elliptic particles decorated with attractive short-range patches are investigated. By imitating the adsorption process of particles into assembled structures on a surface using grand-canonical Monte Carlo simulations, I characterize various such assemblies, ranging from porous networks to chains. Structural properties, as well as some dynamic effects, are described.
Part I of this thesis collects the description of all methods used throughout the research projects presented in Part II and Part III.

Kurzfassung

Das Selbstorganisationsverhalten von kolloidalen Teilchen, die über sogenannte Patches - Interaktionsflächen auf den Oberflächen der Kolloide - gegenseitige Bindungen eingehen können, ist ein sich rasant entwickelnden Forschungsgebiet in der Chemie und Physik. Die Vielfältigkeit der bestehenden und zukünftigen technologischen Anwendungen von Materialien mit Strukturen auf Nano- und Mikrometerskalen, begründet das starke Interesse an der Erforschung von Synthese und Phasenverhalten dieser Teilchen.
In simulationsbasierten und theoretischen Modellen haben Kolloide mit hartem Interaktionspotential eine lange Geschichte, beginnend in den 1950er Jahren als Rechenmaschinen zu einem unverzichtbaren Werkzeug in der statistischen Physik wurden. Das Modell von harten Teilchen mit Patches wurde in den 1980er Jahren für die theoretische Beschreibung von assoziierten Flüssigkeiten entwickelt. Heutzutage sind Teilchenmodellierung und Simulationsuntersuchungen zur Berechnung des Phasenverhaltens und der Selbstorganisation dieser Teilchen ein wichtiger erster Schritt in Richtung experimenteller Realisierungen.

In dieser Dissertation stelle ich zwei Forschungsprojekte vor. Mittels Monte-Carlo-Simulationen werden Ensembles von elliptischen Teilchen untersucht. Im ersten Projekt, vorgestellt in Teil II, untersuche ich Gitterkonfigurationen welche von harten elliptischen Teilchen gebildetet werden. Ellipsen haben eine unendliche Zahl an Möglichkeiten, sich in Gitter zu konfigurieren. Meine Untersuchungen zeigen, welches dieser Gitter das entropisch günstigste ist. Ich präsentiere eine Gitterparametrisierung sowie die Bestimmung von zwei Gitterkandidaten, namentlich dem parallelen und dem diagonalen Gitter. Darauf folgend werden die freien Energien dieser Gitterkandidaten berechnet. Unter Anwendung von Fehlerkorrekturen, welche von Effekten der endlich großen Simulationsbox stammen, wird gezeigt, dass das parallele Gitter das entropisch günstigste ist.
Im zweiten Projekt, vorgestellt in Teil III, werden die Selbstorganisations Szenarien von mit Patches dekorierten harten elliptischen Teilchen, untersucht. In den Simulationen wird der Adsorptionsprozess der Teilchen auf eine Oberfläche mit Hilfe des großkanonischen Ensembles nachgeahmt. Die enstehenden Teilchenanordnungen werden charakterisiert. Diese reichen von porösen Netzwerken bis hin zu Kettenkonfigurationen. Strukturelle Eigenschaften, sowie einige dynamische Effekte werden beschrieben.
Teil I dieser Arbeit umfasst die Beschreibung aller Methoden zusammen, welche in den Forschungsprojekten aus Teil II und Teil III verwendet werden.

Acknowledgements

I express my thankfulness to the members of the Soft Matter research group, who I have shared so much time every day, *David Toneian*, *Clemens Jochum*, *Benedikt Hartl*, *Andreas Zöttl* and *Gaurav Shrivastav* as well as to *Florian Benedetti*, *Carina Karner* and *Emanuela Bianchi*. Thank you for making office time so much more enjoyable. Thank you for your advice, support and friendship.
A special thank you and acknowledgement of his work goes to *Franciszek Bartnik*, who I was supervising on his Master Thesis and whose code also contributed to the following work.
To all members of the Biointerface Doctoral College: It was such a joyous and educational experience getting to know all of you! I am happy to say that the time spent with you greatly served my personal development equally as much as it created valuable scientific experiences in the interdisciplinary field.
To my supervisor *Gerhard Kahl*, thank you for giving me the chance to begin my PhD within the Soft Matter Group and the Biointerface Doctoral College. Thank you for creating opportunities, so I could meet and learn from different people. Also thank you for your careful proofreading of this thesis.
Very importantly, I want to thank all external scientific advisors who have helped me immensely with the advancement of my projects and from whom I have learned so much, *Eva G. Noya*, *Yura Kalyuzhnyi* and *Andrij Baumketner*.

On a more personal level, I want to thank you, Gaurav, who has supported me while writing this thesis with your always kind words. I will always remember our coffee break and your wise words, "Everything good needs time, even time needs time.", which became my motto during the writing process of this thesis. Thank you, Florian, for your immense support during the series of technical crises. My deep gratitude goes to Carina, who has never failed to believe in me successfully finalizing my PhD journey. Without you, I would not have accomplished what is now finally achieved. Your scientific experience, along with your interest in sharing it with me, was so valuable. Thank you for your support, friendship, spark, and curiosity!
Finally, I would not be finishing this thesis today without the special friendship and support of two persons - Raphaela, I admire your fearlessness and strong will! Clemens, you helped me so much by listening and simply understanding.

With the end of my academic education path, I want to express my gratitude to all lecturers who sparked my interest and curiosity and the tutors who have helped me countless times understand complex problems. Most importantly, thank you to all my lecture hall and office comrades. Thank you for your companionship and your friendship throughout the years!

All jenen die mich aufzogen, nährten, lehrten und bildeten -
meinen Eltern, Großeltern, LehrerInnen, FreundInnen und Geschwistern.

* * *

To everyone who raised, nurtured, taught and educated me -
my parents, grandparents, teachers, friends and siblings.

Contents

Part I.

Methods

1. Simulation Program

For the purpose of conducting research projects within my PhD, I developed a Monte-Carlo (MC) program for the simulation of various particle models on a two-dimensional plane. The types of simulation comprise thermodynamic ensembles, such as the canonical, grand-canonical and isobaric-isothermal ensemble, as well as advanced methods such as the Gibbs ensemble, Replica Exchange Monte Carlo and free energy methods, such as Einstein Molecule method and thermodynamic integration methods.
In this chapter, I want to roughly sketch out the structure and functionality of this program. I will also clarify the naming conventions of MC-parameters and variables that will be used throughout the thesis.
I will refrain from explaining the basics of Monte-Carlo simulations in Soft Matter systems. Instead, I will refer to standard literature such as *Understanding Molecular Simulations* by Frenkel and Smit [1], *Computer Simulation of Liquids* by Allen and Tildesley [2] or *A Guide to Monte Carlo Simulations in Statistical Physics* by Landau and Binder [3].

While developing the MC-program, the following design goals stood out:

1. **Modularity** - creating a code that is modularized in logical entities entering the simulation of the system. Details on all modules are discussed in section 1.2.

2. **Configurability** - simulation parameters can be configured upon execution of the code, either via terminal input or via configuration files. Still, there exists an option of using default parameters, meaning that the user can use hard-coded, predefined simulation parameters.

3. Designing the code for **multiple use cases** such as different thermodynamic ensembles, free-energy methods or extended ensembles. Combining multiple options for simulation methods was achieved by using a Fortran preprocessor, specifically the Intel® preprocessor `fpp`. The preprocessor interprets directives within the code aimed at the preprocessor specifically. By defining macros and using `if-else` conditions, the preprocessor selects pieces of code that put together give the source code for the defined simulation method. Further explanation of available macros (indicated as MACRO) follows in section 1.1.

4. **Usability** - simulations may extend over several days of computing time. Because of time limits on supercomputers, the simulation program needed to be designed such that after initial equilibration, a simulation can be paused at any moment. After this pause, the simulation can be safely resumed.

1.1. Functionality

The functionality of simulation methods, such as simulation of different thermodynamic ensembles and other advanced methods within the MC-program are briefly explained in this section. Parameters that reference a parameter within the MC-program are written as `parameter`. Also, with each method, the macros that must be defined are specified as MACRO.

Canonical Ensemble - NVT

In each cycle a total number of `N` MC-moves are proposed. A MC-move is either a translation or rotation move of a single particle and picked with probabilities `probT` and `probR`(→ `module_mc.F90`). Usually with `probT=0.5` and `probR=0.5`.

Isothermal-Isobaric Ensemble - NPT, NPTA, NPTL

In the isothermal-isobaric ensemble, volume varies, and pressure is kept constant. Therefore attempts to change the volume (using NPT) and also shape (using NPTL and/or NPTA) of the simulation box are introduced. The additional MC-move changing size and/or shape of the simulation box is attempted with an overall probability of `probV`, such that `probV+probT+probR=1`.
For simulations of liquids or gases, the shape of the simulation box remains constant, and only its size is scaled (NPT). The respective MC-move proposes a volume change to $\log V$ [1, p. 119].
If a solid or any other system with internal stresses and strains is simulated, MC-moves acting on the shape of the simulation box to release these stresses must be integrated. Therefore by defining an additional macro NPTL the two sides `Lx` and `Ly` of the box are attempted to be changed independently of one another. The respective probabilities for a MC-move acting on the two sides are `probLx` and `probLy`, such that `probLx+probLy=1`.
For full flexibility of the simulation box shape, another macro NPTA is defined that introduces a MC-move attempting to change the angle `theta` enclosed by `Lx` and `Ly` with probability `probTh`, such that `probLx+probLy+probTh=1`.

Varying Lattice Conformation - NPTW, NPTT

For the problem of lattice conformations of hard ellipses (see specifically in part II) the lattice parameter space spanned by two lattice parameters, ω and

τ, must be sampled. A high probability of a specific lattice parameter will then indicate a stable state point. For this problem, two lattice state MC-moves were introduced - a ω-move (NPTW) and a τ-move (NPTT) which attempt changes of the two lattice parameters. The two methods work independently from one another, meaning that the method NPTW attempts changes in ω at constant τ, and the method NPTT attempts changes in τ at constant ω. An in-depth discussion of the underlying algorithm is presented in section 2.2.
Lattice parameter moves are attempted with probability `probOm` and `probTa`. The probability `probOm` is chosen such that `probR+probT+probOm=1` and probability `probTa` is chosen equivalently. Both methods use a maximum displacement `dOmax` and `dTmax` that are adjusted to achieve an acceptance ratio between `OmRatio_lower` and `OmRatio_upper` or `TaRatio_lower` and `TaRatio_upper`.

Grand Canonical Ensemble - MVT

For simulation of the grand canonical ensemble, in which, in addition to the volume and temperature, the chemical potential μ is kept constant, particle insertion and deletion moves are introduced. These moves occur with an overall probability `probN`. Within this implementation, the probability of choosing an insertion over a deletion move is always 50%. Other than that, different schemes for choosing between insertion and deletion move exist in the literature, see [1, p. 130] and might be designed to achieve better efficiency of the algorithm as it adjusts insertion and deletion probability to the current particle density. However, in the framework of this program, this scheme could not be realized. Alternatives to adjusting to the present adsorption or desorption behaviour of the system could not be designed while guaranteeing detailed balance conditions.
If a particle deletion is picked for an empty simulation box, the MC-move is immediately rejected. On the opposite side, the simulation box can be filled up with particles up to the closest packing fraction. The allocated memory for the arrays holding positions and orientations was chosen according to the maximum number of particles `N` in the simulation box.

Extended Hamiltonians - Einstein Molecule Method - EIN, POINT

For more elaborate simulation methods such as the *Einstein Molecule Method* (see section 4.4), the Hamiltonian with the kinetic and particle interaction term can be extended by an Einstein Crystal potential, i.e. quadratic restoring forces of particles to their equilibrium lattice positions and orientations. In the code, the Einstein crystal Hamiltonian is switched on using EIN. Another step of the *Einstein Molecule Method*(section 4.4) involves umbrella sampling (section 4.2) where particles are treated as point particles. For this step, the macro POINT is defined.

Replica Exchange MC - REX

In Replica Exchange Monte-Carlo simulations, a number of simulation boxes, so-called replicas, are simulated in parallel with each box at a different temperature, covering a predefined temperature range. Between the replicas, swap moves are proposed with probability `probS` attempting to swap configurations between two boxes. In total, `probT+probR+probS=1`.
Maximum displacements for translation and rotation are adjusted to achieve an acceptance rate of MC-moves within every box independently.

Gibbs Ensemble - GIBBS

The Gibbs Ensemble is mostly used to simulated phase coexistence between liquid and gaseous states [1]. The overall thermodynamic parameters `N`,`V` and `T` are given. The two boxes have volumes `V1`, `V2` and particles `N1`, `N2` but share the overall constant volume `V=V1+V2` and particle number `N=N1+N2`. With a probability of `probV` a volume exchange between the two boxes is proposed. Particles are exchanged between two simulation boxes with probability `probX`. In total `probT+probR+probV+probX=1`.
Like for Replica Exchange simulations, maximum displacements for translation and rotation are adjusted in each box independently, resulting in a larger maximum displacement for the gas phase and a smaller one for the liquid phase.

Simulation of Direct Coexistence - COEX1, COEX2

In contrast to Gibbs Ensemble, where the simulation design tries to imitate liquid and gas states in bulk, direct coexistence simulations explicitly simulate the interface between two phases (usually liquid and solid phase) within the simulation box. Using this method, first, a solid phase must be equilibrated within a simulation run in the canonical ensemble (NVT). The equilibrated configuration is taken as the initial configuration of the direct coexistence simulation. Therefore, the simulation box size is doubled. The program takes an empty copy of the original simulation box and merges the two boxes at one edge. The second simulation box is initialized with a random configuration of particles at a chosen density. Within the first step (defining **COEX1**), the solid particles are omitted from all MC-moves while the particles in the liquid phase are relaxed while in contact with the solid interface. The next step (defining **COEX2**) considers all particles, solid and liquid particles, within a canonical ensemble.

1.2. Modularity and Configurable Parameters

In the following, all modules within the MC-program are briefly discussed. The module structure was chosen such that they represent logical and practical en-

tities. The most important parameters of each module are listed in table 1.1-table 1.6, explained, and options for program configuration are given.

Global Parameters - `module_parameters.F90`

The module contains all global parameters such as precision `prec` of real numbers, the integer kind `intk`, the path to the project's folder location, internal paths, as well as file names and formats. Finally, this module also contains the number of particles `N` [1] which is a static parameter.

Initial Configuration - `module_ini.F90`

Depending on particle type and on the type of problem, different initial configurations can be created. Generally the type of initial configuration is selected by two parameters `or_order` and `pos_order`, which indicate the particles' orientational and positional order. The options for the parameter indicating the positional ordering, `pos_order` are given in table 1.1.

parameter	value	description
`pos_order`	`rand`	random positions
	`equ`	positions from an equilibrated conf.
	`exp` (`exp-hex`)	expanded hexagonal lattice conf.
	`exp-squ`	expanded square lattice conf.
	`ldg`	low density groundstate
	`cpd`	positions according to closest packed density
	`tau`	positions according to ellipse lattice parameter τ
`or_order`	`rand`	random orientations
	`equ`	orientations from an equilibrated conf.
	`dimer`, `trimer`, `zigzag`, `line`	orientations for patchy disk particle lattice phases
	`kagome`	orientations for the patchy disk particle kagome lattice
	`ome`	orientations according to ellipse lattice parameter ω

Table 1.1.: Parameters of `module_ini.F90` and description of values for configuration.

[1] In the grand-canonical ensemble `N` refers to the maximum number of particles in the simulation box. The number of particles is then defined as `NPart`.

Monte Carlo Parameters - `module_mc.F90`

The module `module_mc.F90` contains all parameters building the framework of all different types of MC simulation. Some of these are the MC simulation's total length in cycles, length of discarded cycles and sampling step size.

The main structural component of the MC simulation is the cycle. In each cycle a number of `N` (→ `module_parameters.F90`) MC-moves are proposed. A typical MC-move is proposed to a randomly chosen particle or cluster of particles (see section 2.3). Different types of MC-moves depend on the chosen statistical ensemble. On average `N*probT` translation MC-moves are proposed every cycle, with the probability parameter `probT`. Probability parameters for each type of MC-move are set within `module_mc.F90`.
Every MC-simulation consists of a number of cycles (`no_cycles`) of which a number of cycles (`disc_cycles`) are used for an initial equilibration as well as for adjusting MC-parameters such as the maximum displacements. This adjustment is made such that acceptance ratios of MC-moves within target intervals are achieved. These cycles (`disc_cycles`) are not used for sampling; configurations are discarded. For the remaining number of cycles (`samp_cycles`), the sampling process takes place. For the duration of the sampling cycles, configurations and system parameters are saved every `samp_step` cycle.
As an overview of the most important parameters from `module_mc.F90`, see table 1.2.

parameter	description
`no_cycles`	total number of cycles
`disc_prop`	proportion of `no_cycles` used for adjusting maximum displacements that then will be discarded
`equ_prop`	proportion of `no_cycles` used for an initial equilibration of the configuration in the canonical ensemble for subsequent simulation using NPT or REX
`samp_step`	number of cycles between sampling two configurations
`par_step`	number of cycles between sampling system parameters, such as total energy, number of particles N for MVT or ω, τ for NPTW and NPTT
`adj_step`	number of cycles between computing acceptance ratios and adjusting maximum displacements accordingly
`probT`, `probR`, ...	probabilities for proposing an MC-move according to the ensemble

Table 1.2.: Configurable parameters of module `module_mc.F90` used to define the framework of the MC-simulation.

Acceptance Statistics - `module_acc.F90`

This module contains all parameters that are used to adjust the acceptance ratios of proposed MC-moves; see table 1.3. This adjustment takes place only within the first equilibration cycles `disc_cycles` and every `adj_step` cycle (→ `module_mc.F90`). When `disc_cycles` are completed, the maximum displacements are fixed at an average value from the last `tr_len` adjusted maximum displacements. This feature is implemented in order to average out fluctuations that might occur during the adjustment of the maximum displacement.

parameter	description
`tr_len`	tracking length for maximum displacement `dUmax`
`dUmax_init`	initial value for maximum displacement `dUmax` (can be adjusted to system's initial density)
`dUmaxUP`	maximum value for `dUmax`
`dUmaxLOW`	minimum value for `dUmax`
`TRatio_upper`	upper bound for targeted acceptance ratio
`TRatio_lower`	lower bound for targeted acceptance ratio

Table 1.3.: Configurable parameters of `module_acc.F90` and descriptions for adjusting acceptance ratios for translation MC-moves. Other types of MC-moves use the same naming convention.

Particle Parameters - `module_particle.F90`, `module_numpart.F90`

In this module particle parameters determining the particle geometry and interaction sites as well as subroutines for determining the interaction potential between two particles are contained.
Further, the definition of the simulation unit of length is related to a particle's geometric property, such as the diameter $\sigma := 1$ for disk particles. All other geometric properties are given in this length unit defined corresponding to the particle type. The particle types include single patchy disk particles (SPP), double patchy disk particles (DPP), hard disk particles (HDP), hard elliptic particle (HEP), elliptic patchy particles (EPP) as well as super-elliptic hard particles (SEP). The interaction model, overlap criteria and length units are presented for each research project dealing with a specific particle type independently; see part II and part III.

Thermodynamic Parameters - `module_thd.F90`

The module `module_thd.F90` includes the thermodynamic parameters, such as temperature, pressure, chemical potential, etc. for the selected ensemble type. The simulation unit for the Boltzmann factor is chosen as $k_B = 1$.

parameter	description
`eps`	energy per particle in a bond of two particles, i.e. 50% of the pair-potential depth, such as ϵ for square-well potentials[2])
`max_no_bonds`	maximum number of bonds per particle
`maxNN`	maximum number of neighbours
`ia_diameter`	interaction diameter, i.e. longest dimension
`rho_0`	density for closest packing configuration
`A_part`	area of the particle
`SIGSQ`	maximum extent squared of the hard core body
`acc_rule`	acceptance rule is 0 for point particles (see section 4.4) and 1 else

Table 1.4.: Parameters in `module_particle.F90` that are not specific to one particle type.

parameter	unit	description
`rho_star`	–	reduced particle density, $\rho^* = \rho/\rho_0$ with density of closest packing ρ_0
`T`, `beta`	$[\epsilon]$, $[\epsilon^{-1}]$	temperature and inverse temperature in units of the patch's square-well depth[3], ϵ
`P_star`	$[A_{\text{part}}^{-1}]$	reduced pressure in units of the inverse particle area
`mu_star`	$[T]$	chemical potential in thermal units
`Tmax`,`Tmax`	$[\epsilon]$	min. and max. temperatures used in replica exchange MC (REX)
`beta_f`	$[\beta]$	ficticous inverse temperature for cluster-move MC (CMMC)

Table 1.5.: Configurable parameters of `module_thd.F90`, units and descriptions.

Loop Parameters - `module_lcl.F90`, `module_cpc.F90`

Instead of performing loops over all particle pairs when computing the total energy of the system, the simulation time can be drastically reduced by using linked cell lists [1, p. 550] or the nearest neighbours for closed packed conformations. The loop-options LCL and CPC refer to linked cell lists and closest packed conformations, respectively.
Within the MC-program, the cells must be configured by providing the number of

[2]Square-well potentials are used as part of the Kern-Frenkel type of patch-potential in part III

[2]Patches in part III are designed using a square-well potential of depth ϵ.

cells `no_cells` and additionally a two-dimensional array `no_cells_1d` providing the number of cells along the x- and y axis.

Parameters for Einstein Molecule Method - `module_emm.F90`, `module_field.F90`

The module `module_field.F90` offers the option of extending the Hamiltonian of the simulated system. Apart from the particles' interaction potential, an additional field term can be added to the Hamiltonian. To be more specific, this module is accessed when simulating states for the Einstein Molecule method (see section 4.4).
The module `module_emm.F90` is used for processing the simulation data and computing the free energy contributions.

parameter	value or unit	description
`tf_method`	`non`, `log`, `mob`	selects method of transformation for numeric integration of ΔA_2 within the Einstein Molecule Method (see section 4.4).
`Lambda`	$[\beta\sigma^{-2}]$, $[\beta rad^{-2}]$	value for the re-scaled Einstein spring constant, with σ as the respective unit of length
`no_lambdas`	-	number of sampling points in the numeric integration ΔA_2.
`tf_param`	-	the transformation parameter for the selected transformation method.
`trunc_length`	$0.0 - 1.0$	truncation length for MC-trajectories

Table 1.6.: Configurable parameters in `module_field.F90` for simulation and configurable parameters of `module_emm.F90` for computation of free energies within the Einstein Molecule method (see section 4.4)

LOG module - `module_log.F90`

The objective of `module_log.F90` is to read and write to PAR-files that are stored with simulation output files and store names and values of configured parameters from all modules. In this way, the module ties together all modules containing configurable parameters.
Three types of PAR-files exist: `PAR_ini`, `PAR_sim` and `PAR_equ`, where the first one includes configuration parameters for the initial configuration while the latter two include simulation parameters.

Testing MC-moves - `module_testmoves.F90`

The module `module_testmoves.F90` entails all methods for testing whether a proposed MC-move is accepted or not. Subroutines for each MC-move include

the energy calculation of the new and old configuration as well as the Metropolis algorithm [4] resulting in an acceptance or rejection of the proposed move.

Cluster-MC functions - `module_cluster.F90`

This module includes all functions used for the cluster moving algorithm (see section 2.3). Also cluster analysis tools are contained in this module.

Modules for post-processing - `module_analysis.F90`, `module_ana.F90`, `module_thdint.F90`

The modules contain post-processing functions and respective parameters. These modules are used not within the simulation program but within post-processing programs such as `Analysis.F90` or `FreeEnergy.F90`.

However, most of the post-processing is preformed using a newly developed python3 [5] package, customized for data analysis of the simulation output by the Fortran-95 simulation program. This package mostly relies on the python-modules *pandas* [6], *numpy* [7] and *networkx* [8].

2. Algorithmic Details

Within this chapter, the parts of the algorithms used for simulation and analysis are discussed in detail. In section 2.1 the representation of particle positions within the simulation box and, in particular, this representation for a lattice of ellipses is presented. Following, section 2.2 presents two types of MC-moves to sample a lattice-configurations of hard ellipses with different lattice parameters. Another simulation method, the cluster move algorithm, as proposed by Whitelam [9] and an adaption to chain-movement, is presented in section 2.3. Finally, section 2.4 presents a newly developed method for the analysis of particle networks. The so-called loop-detection algorithm identifies particle loops within a network, based solely on its abstract graph properties.

2.1. Simulation Box Matrix & Unit Box Coordinates

Within the MC-simulation program the N particle positions are stored as a vector $\vec{u}_i$ (with $i = 1...N$) in unit-box coordinates. These are Cartesian coordinates within the unit square with the center at the point of origin for the most efficient implementation of periodic boundary conditions [1, p. 32]. The particles' real positions, represented as vectors $\vec{r}_i$ $(i = 1, ..., N)$ are obtained by transformation of $\vec{u}_i$ via the simulation-box matrix, $\mathbf{B}$:

$$\vec{r}_i = \mathbf{B} \cdot \vec{u}_i \tag{2.1}$$

The form of the simulation-box matrix determines the shape and size of the physical simulation box.

2.1.1. Representation of the hexagonal lattice of disks

A particular lattice state is always represented as N particle positions $\vec{u}_i$ in the unit box and a simulation-box matrix. For the hexagonal lattice I chose to split the lattice points into two rectangular sub-lattices $\vec{u}_{lm}$ and $\vec{u}_{l'm'}$. The indices (l, m) as well as (l', m') giving the lattice positions such that each particle i has a unique lattice position, $i \equiv (l, m), i \in [1, \frac{N}{2}]$ and $i' \equiv (l', m'), i' \in (\frac{N}{2}, N]$. The sub-lattices both carry N_x particles along the x-axis and $\frac{N_\mathrm{y}}{2}$ particles along the

y-axis, such that $N = N_\mathrm{x} \cdot N_\mathrm{y}$. The lattice positions are given by:

$$\vec{u}_{lm} = \begin{pmatrix} u_{\mathrm{x},l} \\ u_{\mathrm{y},m} \end{pmatrix} = \begin{pmatrix} -0.5 + a_\mathrm{x}(l-1) \\ -0.5 + a_\mathrm{y}(m-1) \end{pmatrix}, l \in [1, N_\mathrm{x}] \text{ and } m \in [1, \frac{N_\mathrm{y}}{2}] \tag{2.2a}$$

$$\vec{u}_{l'm'} = \begin{pmatrix} u_{\mathrm{x},l'} \\ u_{\mathrm{y},m'} \end{pmatrix} = \begin{pmatrix} -0.5 + a_\mathrm{x}(l'-0.5) \\ -0.5 + a_\mathrm{y}(m'-0.5) \end{pmatrix}, l' \in [1, N_\mathrm{x}] \text{ and } m' \in [1, \frac{N_\mathrm{y}}{2}] \tag{2.2b}$$

with lattice constants:

$$a_\mathrm{x} = N_\mathrm{x}^{-1} \tag{2.3}$$

$$a_\mathrm{y} = 2N_\mathrm{y}^{-1} \tag{2.4}$$

Given the unit-box coordinates, the simulation-box matrix for a hexagonal lattice of disks with diameter $\sigma = 1$ at a reduced density $\rho^* = \rho/\rho_\mathrm{cpd}$, with ρ_cpd referring to the density at closest packing, will then take the form:

$$\mathbf{B}^\mathrm{disk}(N, \rho^*) = \sqrt{\frac{N}{\rho^*}} \begin{bmatrix} 1 & 0 \\ 0 & \frac{\sqrt{3}}{2} \end{bmatrix} \tag{2.5}$$

2.1.2. Lattices of Elliptic Particles

In the following we will consider a special class of lattices of elliptic particles. These lattices are based on the hexagonal lattice of disks, which means that they can be related to the hexagonal lattice by a number of affine transformations. When creating these lattices for ellipses with a certain aspect ratio κ at reduced density $\rho^* = \rho/\rho_\mathrm{cpd}$, we follow the steps below, each step performing a transformation on the hexagonal lattice of disks:

proposition: A simulation-box matrix $\mathbf{B}^\mathrm{ell}$ shall be formulated such that $\vec{r}_i$ gives the position of each particle i in the lattice:

$$\vec{r}_i = \mathbf{B}^\mathrm{ell} \cdot \vec{u}_i \tag{2.6}$$

step 1: Starting with the the hexagonal lattice of disks, the unit-box coordinates are given by $\vec{u}_i$ as specified for sub-lattice coordinates (l, m) and (l', m') in eq. (2.2). The real positions of disk particles $\vec{r}_i{}'$ are then given by the matrix $\mathbf{B}^\mathrm{disk}$ as in eq. (2.5):

$$\vec{r}_i{}' = \mathbf{B}^\mathrm{disk}(N, \rho^*) \cdot \vec{u}_i \tag{2.7}$$

step 2: To create ellipses from disks we transform the present configuration via elongation along the x-axis and compression along the y-axis. The aspect ratio of the ellipses is defined as $\kappa = \frac{a}{b}$, with semi-major axis a and semi-minor axis b. By applying a transformation $\mathbf{H_1}(\kappa)$ such that the total area is conserved, i.e. $\det(\mathbf{H_1}) = 1$, the unit of length for disks, given by the diameter $\sigma = 1$, simply transforms into the equivalent unit of length for ellipses, $2\sqrt{ab} = 1$. The respective unit area defining the simulation units is given by $4ab = 1$. The transformation matrix is given as:

$$\mathbf{H_1}(\kappa) = \begin{bmatrix} \sqrt{\kappa} & 0 \\ 0 & \frac{1}{\sqrt{\kappa}} \end{bmatrix} \tag{2.8}$$

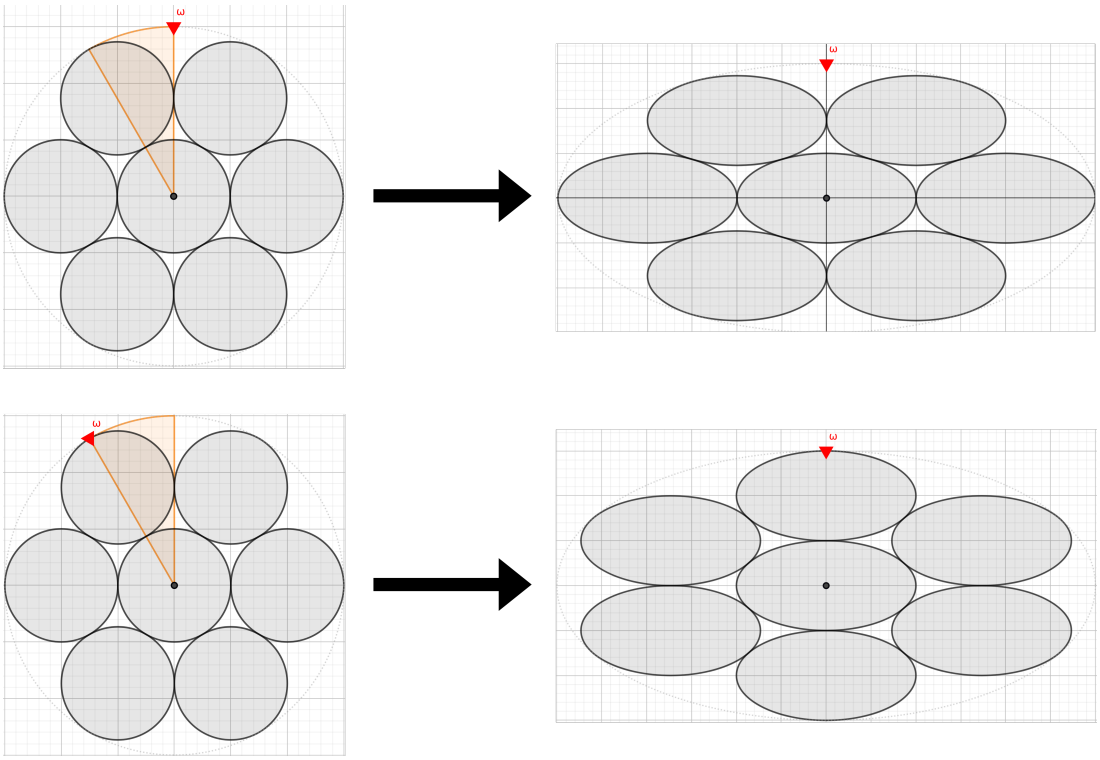


Figure 2.1.: Origin of closest packings of ellipses at different values of ω. The orange wedge indicates possible values for ω that all result in different ellipse lattice configurations. Top configuration at $\omega = 0$ is named parallel, bottom configuration at $\omega = \frac{\pi}{6}$ is named diagonal configuration.

step 3: Due to the six-fold rotational symmetry of the original hexagonal lattice of disks, an additional rotation, $\mathbf{R}$ is introduced before applying $\mathbf{H_1}$ in

step 2. As can be seen in fig. 2.1, each angle of rotation ω within $[0, \frac{\pi}{6})$ results in a different configuration:

$$\vec{r} = \mathbf{R}(\omega')\mathbf{H_1}(\kappa)\mathbf{R}(\omega)\mathbf{B}^{\text{disk}}(N, \rho^*) \cdot \vec{u} \tag{2.9}$$

A second rotation by an angle ω' is introduced for the sole reason to align the lower lattice axis with the x-axis. For deriving ω' we take a vector $\vec{c}$ which is a unit vector aligned with the horizontal lattice axis:

$$\vec{c} = \begin{pmatrix} \cos\omega \\ \sin\omega \end{pmatrix} \tag{2.10}$$

After performing the transformation ($\vec{c} \to \vec{c}\,' = \mathbf{H_1}(\kappa)\vec{c}$), the rotation angle ω' for horizontal alignment of $\vec{c}$ is therefore given by:

$$\cos\omega' = \frac{\hat{e}_x \cdot \vec{c}\,'}{|\vec{c}\,'|} = \cos\omega' = \frac{\cos\omega}{\sqrt{\cos^2\omega + \kappa^{-2}\sin^2\omega}} \tag{2.11}$$

step 4: The larger the angle ω the stronger the distortions of the rectangular box by $\mathbf{H_1}$. The simulation box will therefore be a parallelogram with short distances between periodic boundaries. For simulating particle conformations in the bulk, such a geometric shape of a simulation box is not desirable due to increased effects of the periodic boundaries in the bulk. To avoid this distortion, for $\omega > \frac{\pi}{12}$ another lattice representation is used by introducing another transformation $\mathbf{T}$ with the effect of mirroring the unit-box positions along the identity-line (at 45°). The matrix $\mathbf{H_1}$ is then applied onto the mirrored coordinates:

$$\mathbf{H_1}' \cdot \vec{u}\,' = \mathbf{T}\mathbf{H_1}\mathbf{T}^{-1}\mathbf{T} \cdot \vec{u}. \tag{2.12}$$

Effectively the replacement done is $\mathbf{H_1} \to \mathbf{T}\mathbf{H_1}$, with $\mathbf{T}$ as:

$$\mathbf{T} = \begin{cases} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} & \text{, if } \omega \leq \dfrac{\pi}{15} \\ \\ \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} & \text{, otherwise.} \end{cases} \tag{2.13}$$

Note that using this effective replacement, the unit-box coordinates remain untouched as $\mathbf{T}^{-1}\mathbf{T} = \mathbb{1}$. However, the mirror-transformation will transform the particles' orientations ω as $\omega \to \frac{\pi}{6} - \omega$.

step 5: In addition to the lattice parameter ω, introduced in **step 3** that represents the relative orientation of the ellipses to the lattice axis, another lattice parameter, that will be named τ, is needed to take into account an additional

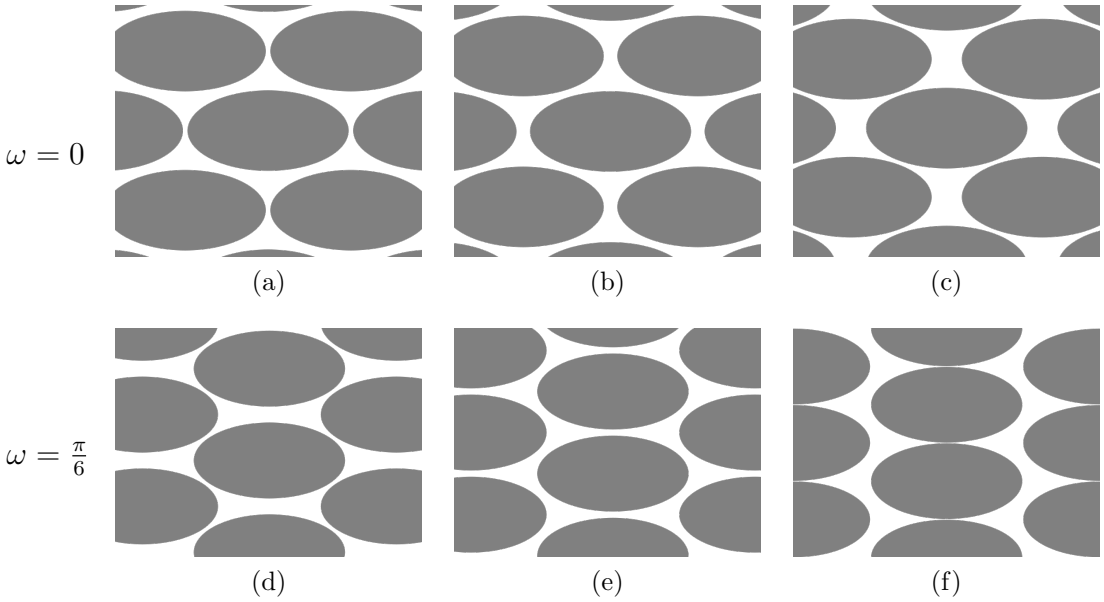


Figure 2.2.: Lattice configurations of ellipses demonstrating parameter τ at two lattice parameters ω: (a) $\tau = 0.9$, (b) $\tau = 1.0$, (c) $\tau = 1.2$, (d) $\tau = 0.8$, (e) $\tau = 1.0$ and (f) $\tau = 1.14$.

freedom of the lattice to arrange the particles positions in horizontal and vertical direction. This parameter therefore determines the relative positions of the ellipses to one another, as can be seen in fig. 2.2. Another affine transformation, $\mathbf{H_2}$, is introduced that acts on the particles' position only and not on their shape:

$$\mathbf{H_2}(\tau) = \begin{bmatrix} \sqrt{\tau} & 0 \\ 0 & \frac{1}{\sqrt{\tau}} \end{bmatrix}. \tag{2.14}$$

In the case of a rectangular simulation box τ is interpreted as a parameter that represents the aspect ratio of the box, given by $\kappa_{\text{box}} = \frac{L_x}{L_y}$ relatively to the aspect ratio $\kappa_{\text{box},0}$ which is the box's aspect ratio obtained from $\mathbf{H_1}$ alone - $\tau = \frac{\kappa_{\text{box}}}{\kappa_{\text{box},0}}$. This condition of a rectangular simulation box is fulfilled for values $\hat{\omega}_1 = 0$ and $\hat{\omega}_2 = \frac{\pi}{3}$, which are named the diagonal and parallel configuration respectively in part II.

summarizing all steps: Following all these steps, the total simulation-box matrix, $\mathbf{B}^{\text{ell}}$ factorizes into the following sub-matrices:

$$\mathbf{B}^{\text{ell}}(\omega, \tau, \kappa, N, \rho^*) = \mathbf{R}(-\omega') \cdot \mathbf{H_1}(\kappa) \cdot \mathbf{R}(\omega) \cdot \mathbf{H_2}(\tau) \cdot \mathbf{T}(\omega) \cdot \mathbf{B}^{\text{disk}}(N, \rho^*), \tag{2.15}$$

where the matrices $\mathbf{H_1}(\kappa)$,$\mathbf{H_2}(\tau)$ and $\mathbf{R}(\omega)$ define the lattice state while $\mathbf{T}(\omega)$ and $\mathbf{R}(-\omega')$ define it's representation in the simulation box and hence are optional.

For the case of $\omega < \frac{\pi}{12}$ the simulation-box matrix gives:

$$\mathbf{B}^{\text{ell}} = \begin{bmatrix} \frac{1}{\sqrt{\kappa}}\cos\omega\sin\omega' - \sqrt{\kappa}\sin\omega\cos\omega' & \frac{1}{\sqrt{\kappa}}\sin\omega'\sin\omega + \sqrt{\kappa}\cos\omega\cos\omega' \\ \frac{1}{\sqrt{\kappa}}\cos\omega\cos\omega' + \sqrt{\kappa}\sin\omega'\sin\omega & \frac{1}{\sqrt{\kappa}}\sin\omega\cos\omega' - \sqrt{\kappa}\cos\omega\sin\omega' \end{bmatrix} \cdot \tag{2.16}$$

$$\cdot \begin{bmatrix} \frac{1}{\sqrt{\tau}} & 0 \\ 0 & \sqrt{\tau} \end{bmatrix} \sqrt{\frac{N}{\rho^*}} \begin{bmatrix} 1 & 0 \\ 0 & \frac{\sqrt{3}}{2} \end{bmatrix}$$

Implementation For the implementation within the Fortran-95 Simulation Program an array `SB` of dimension $(2, 3)$ was chosen, such that:

$$\texttt{SB(1:2,1:3)} = \tag{2.17}$$

$$\begin{bmatrix} \frac{1}{\sqrt{\kappa}}\cos\omega\sin\omega' - \sqrt{\kappa}\sin\omega\cos\omega' & \frac{1}{\sqrt{\kappa}}\sin\omega'\sin\omega + \sqrt{\kappa}\cos\omega\cos\omega' & \sqrt{\frac{N}{\rho^*}}\cdot\sqrt{\tau} \\ \frac{1}{\sqrt{\kappa}}\cos\omega\cos\omega' + \sqrt{\kappa}\sin\omega'\sin\omega & \frac{1}{\sqrt{\kappa}}\sin\omega\cos\omega' - \sqrt{\kappa}\cos\omega\sin\omega' & \frac{\sqrt{3}}{2}\cdot\sqrt{\frac{N}{\rho^*}}\cdot\frac{1}{\sqrt{\tau}} \end{bmatrix}$$

For the case of $\omega > \frac{\pi}{12}$ (see **step 5**), `SB(1,3)` is exchanged with `SB(2,3)` and $\omega \to \frac{\pi}{6} - \omega$.

With this form of implementation, the following is true:

$$\det(\texttt{SB(1:2,1:2)}) = 1 \tag{2.18}$$

$$\texttt{SB(1,3)} \cdot \texttt{SB(2,3)} = \mathrm{A}_{\text{box}} \tag{2.19}$$

The implementation is motivated by partly separating box shape and box size, given by its area A_{box} into distinct elements in the array.

2.2. Lattice MC-moves

The obstacle of finding the candidate lattice structures from a lattice parameterization for lattices of hard ellipses, see section 2.1.2 was tackled by designing special Monte-Carlo moves that attempt to change the lattice parameters ω and τ.

Studying the lattices of hard ellipses, I designed the lattice MC-moves for each lattice parameter independently. For lattice moves, I chose to represent the particle positions and orientations as a combination of the lattice sites and orientation

and the respective deviation:

$$\vec{r}_i = \mathbf{B}^{\text{ell}}(\omega,\tau)(\vec{u}_i{}^0 + \Delta\vec{u}_i) \tag{2.20a}$$
$$\phi_i = \phi_i^0 + \Delta\phi_i \tag{2.20b}$$

with the simulation-box matrix $\mathbf{B}^{\text{ell}}(\omega,\tau)$ representing the transformation of the unit-box coordinates $\vec{u}$ into simulation box coordinates $\vec{r}$ and simultaneously defining the lattice state (ω,τ) (see for more details section 2.1.2). The coordinates $\vec{u}_i{}^0$ and ϕ_i^0 define the lattice-positions in unit-box coordinates and the particle's ideal orientation in the reference lattice state. The simulation-box matrix was defined such that the unit box coordinates $\vec{u}_i{}^0$, as given in eq. (2.2) are independent of the lattice parameters ω and τ. The particles' lattice orientations ϕ_i^0 however depend on the lattice state ω via $\phi_i^0 = \omega'(\omega)$, as derived in eq. (2.11).

Lattice MC ω-move

When attempting a lattice MC ω-move, a random displacement $\Delta\omega$ is chosen within an interval $(-\Delta\Omega_{\text{max}}, \Delta\Omega_{\text{max}})$. The bounding maximum displacement $\Delta\Omega_{\text{max}}$ is adjusted during the course of the discarded cycles (see section 1.2). The adjustment is made according to the move's acceptance ratio, equal to the adjustment of the maximum displacements for single-particle translation and rotation moves, with the goal of reaching an acceptance ratio between $20-50\%$.

During the move from the old ω^o to the new state ω^n by a random step $\omega^n = \omega^o + \Delta\omega$, the particles' positions and orientations change accordingly. Considering the positions, eq. (2.20a), the unit-box coordinates $\vec{u}_i{}^0$ stay unchanged as the ω-state affects the matrices $\mathbf{R}(\omega)$, $\mathbf{R}(-\omega')$ and $\mathbf{T}(\omega)$ as part of the simulation-box matrix:

$$\mathbf{B}^{\text{ell}}(\omega^o) \rightarrow \mathbf{B}^{\text{ell}}(\omega^n = \omega^o + \Delta\omega) \tag{2.21}$$

The transformation of particle orientations ϕ_i, in eq. (2.20b), are given by:

$$\phi_i \rightarrow \phi_i - \omega'(\omega^o) + \omega'(\omega^n), \tag{2.22}$$

as by design the particles lattice orientations $\phi_i^0 = \omega'$. The update of the orientations is done corresponding to the new and old lattice parameter $\omega^{o/n}$ via:

$$\omega'(\omega^{o/n}) = \arccos\left(\frac{\cos\omega^{o/n}}{\sqrt{\cos^2\omega^{o/n} + \frac{1}{\kappa^2}\sin^2\omega^{o/n}}}\right). \tag{2.23}$$

Notice, that if the updated angle, ω^n falls below or above $\omega = \frac{\pi}{12}$ compared to the old angle, ω^o, the simulation-box matrix will undergo the mirroring transformation $\mathbf{T}$ (see section 2.1.2). Also the particles' positions are mirrored by

interchanging their x- and y-coordinates.

Lattice MC τ-move

A lattice MC τ-move, $\tau^n = \tau^o + \Delta\tau$, is attempted via choosing a random displacement of the lattice parameter τ $\Delta\tau$ within the interval $(-\Delta\tau_{\max}, \Delta\tau_{\max})$. The interval's borders are adjusted to achieve an imposed acceptance ratio between $10 - 60\%$.

The move will attempt a change within the simulation-box matrix,

$$\mathbf{B}^{\text{ell}}(\tau^o) \rightarrow \mathbf{B}^{\text{ell}}(\tau^n = \tau^o + \Delta\tau) \tag{2.24}$$

Within the chosen implementation of the simulation-box matrix, see eq. (2.17), the τ-move aims only at two simulation-box entries according to:

$$\texttt{SB}(1,3) \rightarrow \texttt{SB}(1,3) \cdot \frac{\sqrt{\tau^n}}{\sqrt{\tau^o}} \tag{2.25}$$

$$\texttt{SB}(2,3) \rightarrow \texttt{SB}(2,3) \cdot \frac{\sqrt{\tau^o}}{\sqrt{\tau^n}} \tag{2.26}$$

Within the τ-move, the overall area of the simulation box is not changed. Overlapping particles will be the result of τ-moves that extend over the minimum or maximum value for τ. Because these moves are rejected anyhow, a proper definition of the minimum and maximum value for τ is not necessary.

Relations to existing algorithms

The above-suggested algorithm for lattices of hard ellipses relates closely to a MC-algorithm called the *lattice switch* algorithm introduced by Bruce, Wilding, and Ackland [10] and applied further in [11] [12]. In the initial contribution, [10], the authors study the problem of hard-sphere packings in three dimensions in order to compare hcp and fcc packings and to find the entropically more favourable lattice type. By splitting the particles' positions into lattice sites and deviations (similarly to eq. (2.20)), the authors propose the additional MC-move that acts on the lattice coordinates only, called the lattice-switch. Compared to my method proposed for hard ellipses in two dimensions, within the study by Bruce, Wilding, and Ackland of the hard-sphere lattices, the authors have to make the additional effort of biasing pathways from the hcp to the fcc lattice-type using a suitable order parameter. Without going into detail about the defined order parameter and biased configuration distribution, I want to point out here that within the hard ellipse lattices, this biasing procedure was not necessary due to the fact that the competing candidate lattice types are infinitely degenerate and lattice parameters (ω, τ) are continuous variables along the corresponding axes. Due to

this continuity, the transition through all lattice candidate structures is smooth and can be easily sampled without the need for biasing pathways.

2.3. Cluster Moving Algorithm

In MC simulations, the single-particle MC-moving scheme, in which one particle is translated or rotated at a time, is the basis for studying Molecular or Soft-Matter systems. In addition to single-particle moves, cluster moves can be introduced that are designed to move several particles as a cluster in concert. Whitelam and Geissler [9] have given a collection of criteria when it is advisable to use cluster moves instead of single-particle moves.

Within my MC-program, the cluster MC-move algorithm using the static linking scheme by Whitelam and Geissler [9] was implemented. This algorithm will shortly be summarized in the next section. Further, a small modification of this cluster MC-move algorithm is introduced for chain-forming clusters and briefly described. Finally, intricacies when applying cluster MC-moves are discussed, and solutions to potential problems are proposed.

2.3.1. Cluster MC-move Algorithm with Static Linking Scheme

The cluster moving algorithm used in this MC-program uses a recursive cluster formation algorithm with a static linking scheme, closely following the same cluster moving algorithm as proposed by Whitelam described in detail in [13].

Within the pseudocluster formation, a random particle is picked as the initial particle, and following a static linking scheme in a recursive manner, other particles can join the so-called pseudocluster. In a static linking scheme, the linking probability solely depends on the initial state, named μ, before the cluster is moved. The linking probability $p_{ij}(\mu)$ for hard core particles, i and j with an attractive patch potential, ϵ_{ij} is given by [13]

$$p_{ij}(\mu) = \Theta(n_c - n_{\mathcal{C}})\mathcal{I}_{ij}(\mu)(1 - \exp[\tilde{\beta}_f \beta \epsilon_{ij}]), \tag{2.27}$$

with $n_{\mathcal{C}}$ as the maximum cluster size, $\mathcal{I}_{ij}$ a function that is unity when particles i and j interact and zero otherwise. The maximum cluster size, $\mathcal{C}$ is drawn from an inverse probability distribution $p(\xi) \sim \xi^{-1}$ for every new cluster move. Additionally to this kind of upper bound for the cluster size, other restrictions had to be installed further explained in section 2.3.3. Finally a fictitious reciprocal temperature parameter $\tilde{\beta}_f \in [0, 1]$ is introduced. Within the linking probability eq. (2.27), this parameter interpolates between the physical reciprocal temperature β and infinite temperature. As an effect in the limiting case of an infinite reciprocal temperature $\tilde{\beta}_f = 0$, the linking probability will be zero, and single-particle instead of cluster moves will be performed. For simulations done within

this thesis, $\tilde{\beta}_f = 0.5$.

Once the pseudocluster is formed either a cluster-translation or a cluster-rotation are proposed and the new state, ν is then accepted with an acceptance probability W_{acc}, depending on the total energy E_ν and E_μ in the old and new state:

$$W_{\text{acc}}(\mu \to \nu) = \min\big(1, \exp[-\beta(1-\tilde{\beta}_f)(E_\nu - E_\mu)]\big). \tag{2.28}$$

Cluster-Moves embedded in MC-program

Within the MC-program, I embedded cluster moves such that either a cluster-translation or cluster-rotation (with `probCT` and `probCR`)is proposed every cycle with a probability `probCM`. With the complementary probability either a single-particle translation or rotation MC-move (with `probT` and `probR`) is proposed.

Maximum displacements in translation `dCUmax` and `dCPhimax` for clusters are treated independently from respective maximum displacements for single-particle moves. Here, a cluster-move with one particle in the pseudocluster is still considered as a cluster move.

Statistics for ratios of accepted versus proposed cluster moves for rotations and translations (`AcCTM` and `AcCRM`) as well as for the average cluster size of all formed pseudoclusters (`CS`) and of all moved pseudoclusters (`AcCS`) are recorded.

2.3.2. Cluster MC-moves for Chains of Convex Particles with Patches

Specifically for particle models that assemble into chains, a variant of the cluster rotation MC-moves was implemented to facilitate equilibration using cluster motions reminiscent of a more realistic dynamic of chains. The type of particle for which this kind of cluster rotation MC-move might be helpful is a particle with an attractive patch located at a vertex of positive curvature. Usually, these types of particles have the ability to form chains with high flexibility - where the proposed cluster MC-move is designed to explore this flexibility.
For a particle of this chain-forming type (see part III), the use of Whitelam's cluster move algorithm led to the observation that the chains, once formed, remained very stiff. Within Whitelam's cluster rotation MC-move, clusters of particles are rotated with respect to the cluster's center of mass. Particularly in the case of very strong interactions when chains do not break apart anymore once they are formed, with a very high probability, the apparent problem is that the possible flexibility of these chains cannot be explored by the present cluster MC-move algorithm. When a pseudocluster that is a fraction of a chain is undergoing a rotation with reference to its center of mass, this rotation move will most likely break the bonds connecting the pseudocluster to the chain. Hence, when the interaction is very strong, this cluster rotation MC-move will be rejected with a very high probability.

In an attempt to remedy this issue, a new reference point for the cluster rotation was chosen. For the chains, we take either of the two end-points of the chain as the reference point. The end-points of the chain are the two exterior vertices of the particles at the ends of the chain. When a pseudocluster is formed, the reference point is randomly chosen between the two end-points.
This new reference point was chosen such that detailed balance is guaranteed - meaning that the location of the reference point with respect to the pseudocluster must not change during the rotation of the pseudocluster. Any reference point that does not fulfil this requirement breaks detailed balance.

2.3.3. Intricacies when using Cluster MC-moves

This section lists a few intricacies that should be considered when implementing cluster MC-moves. I present solutions to problems I faced when studying the patchy ellipses particle model (part III). Particles of the studied type assemble into chain structures, connected networks and open lattices.

Center of Mass and Periodic Boundary Conditions

The calculation of the center of mass (CM) of a cluster within a simulation box with periodic boundary conditions must be done with special care. When a pseudocluster extends over the simulation box boundary, a straightforward calculation of the CM, e.g. via $\vec{r}_{\mathrm{cm}} = \frac{1}{n_{\mathrm{pc}}} \sum_{i=1}^{n_{\mathrm{pc}}} \vec{r}_i$ (with n_{pc} as the number of particles in the pseudocluster) will lead to a wrong result, due to the minimum image convention. Therefore by using a transformation proposed by Bai and Breen [14] will be used to calculate the correct CM. The algorithm uses two independent transformations, one for the particles' x-coordinates, which are projected onto a one-dimensional ring and a second and equivalent transformation for the y-coordinates. The center of mass of each coordinate is calculated in the transformed ring coordinates, and finally, the true CM can be attained by back transformation into the planar simulation box [14].

Rotation with Reference to End-Points and Periodic Boundary Conditions

With the newly proposed chain-cluster rotation MC-move, a special intricacy upon rotating the particles with reference to one of the end-points of the chain may occur. Due to periodic boundary conditions, the distance vector between the reference point and the particle that is rotated could be chosen as the wrong distance-vector when using the minimum image convention, as can be seen in fig. 2.3. When viewing the reference point solely and the particle that is rotated, there is no way of deciding which distance vector - either the distance with or without minimum image convention - is the appropriate one. To determine this appropriate vector, information on the location of the connecting particles is

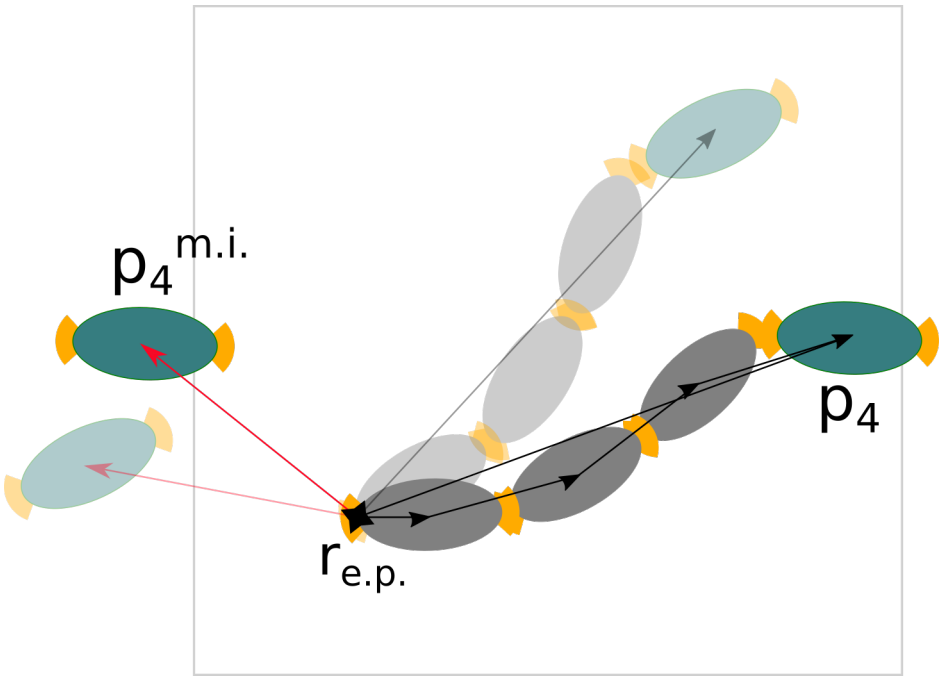


Figure 2.3.: Rotating chain particle p_4 with respect to the chain's end point ($\vec{r}_{\text{e.p.}}$) with (red) and without (black) minimum image convention. Further, particle positions in black with respect to the former chain particle, building up the chain and the correct distance vector for each particle iteratively.

needed. The proposed algorithm to perform the rotation of all pseudocluster particles correctly is the following:

1. start by picking the position of the particle whose vertex serves as the reference point (particle $p_{i=1}$, position $\vec{r}_i$) and setting end-point vector $\vec{r}_{\text{e.p.}}$ as the distance between exterior vertex and $\vec{r}_i$

2. rotate particle p_i by using the distance vector $\vec{r}_{\text{e.p.}}$

3. determine the bonding neighbors of particle p_i

4. pick the bonding neighbor that is part of the pseudocluster $p_{i=i+1}$

5. add the distance vector, $\vec{r}_i - \vec{r}_{i+1}$ to the end-point vector $\vec{r}_{\text{e.p.}} \rightarrow \vec{r}_{\text{e.p.}} + (\vec{r}_i - \vec{r}_{i+1})$

6. rotate particle p_{i+1} using the updated vector $\vec{r}_{\text{e.p.}}$

7. iterate step 2. - 6. until the last particle in the chain (when there is no left bonding neighbor as part of pseudocluster in step 4.)

The described algorithm only works for chains without branching points, which restricts the number of bonds to one per patch. This restriction on the particle model must be considered when applying this type of cluster MC-move.
For other types of chains, e.g. chains with branching points, a slightly extended algorithm could be used, in which the algorithm proceeds from step 5 for each branch independently.

Maximum Cluster Size due to Finite Box Size

When dealing with chain-like clusters and chains extending over the size of the simulation box, two problems can occur: (1) a chain that is enclosed in itself via the periodic boundary conditions will be broken apart by a cluster-rotation, and (2) a cluster-rotation on a chain that is not enclosed and is longer than the extent of the simulation box, can result in an internal overlap of particles. Remember here that interactions within the pseudocluster are considered to be conserved within a cluster MC-move.
For these two reasons, an upper limit for the chain length was determined for which cluster rotations can be performed. This upper limit is defined by the minimum number of particles necessary to form a chain that is enclosed in itself via the periodic boundary conditions. Cluster rotation MC-moves are immediately rejected if this maximum number of particles is exceeded (see subroutine `set_pseudocluster_parameters()`).

Cluster Rotations and Linked Cell Lists

When rotating large clusters (e.g. chains), the positional displacement of particles far from the reference point of rotation (either center of mass or end point of the chain) can easily be larger than the maximum displacement allowed for either single particle or cluster translations. Using linked cell lists, where particle pair interactions are checked only within a cell and its eight neighbouring cells, this new position might not be located within one of these nine cells, determined by the particle's initial position. For this reason, for cluster rotation MC-moves specifically, a computationally more expensive function `virtual_cell_index(i)` is used to determine the cell index of the particle's new location. With this cell as the central cell, new overlaps and interactions with particles of the central and surrounding cells are computed.

As a result, a re-calculation of the old and the new interface energy of the pseudocluster, with the respective cell indices of the old and new positions, is mandatory.

2.4. Algorithms for Particle Network Analysis

In a system of patchy particles (see part III), we observe particles forming network structures. The presented analysis of these structures relies on the representation of the particle network as abstract graphs. First, I discuss the network representation as different types of abstract graphs and then will describe the analysis methods, specifically the loop-counting and artefact-cleaning algorithm, based on these graphs in detail.

2.4.1. Particle networks as abstract graphs

Dealing with the particle network in a more abstract way, mathematical graphs, specifically simple and undirected graphs [15], consisting solely of nodes and edges, are used to represent the physical particle network. Three different representations are identified.

The three types of graph-representations are the *particle-based*, the *patch-based* and the *full* graph-representation (see fig. 2.4). While in the particle-based graph, we represent each particle by one node and its bonds to other particles by an edge, in the patch-based network, each patch is represented by a node and patches on the same core particle are connected by an edge of a different type. The full network consists of two different types of nodes, one representing particle cores and the other patches. An edge again represents either a bond or connects a patch to its core particle node. In all graph types, the particle nodes are named p_i and the patch nodes of the i-th particle are named $p_{i\mathrm{A}}$, $p_{i\mathrm{B}}$, and so forth.

The use of these different types of networks becomes more evident when looking at the calculation of structural characteristics, such as distributions for quantities like bonds per patch, loop size or cluster size. For the calculation of each characteristic, a specific representation of the particle network might turn out to be most useful.

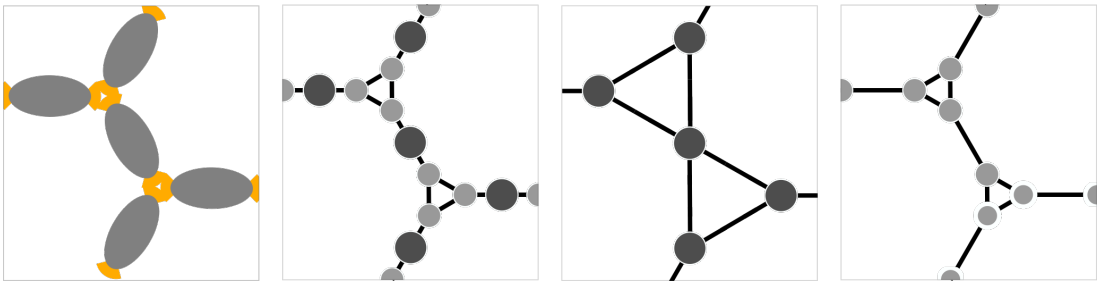


Figure 2.4.: Types of graphs based on patchy particle network, explained by using the example of elliptic patchy particles. From left to right: (a) particle network of patchy ellipses, (b) the *full* network with core and patch - nodes, (c) the *particle-based* network with nodes representing particle cores and (d) the *patch-based* network with nodes representing patches.

2.4.2. Loop-detection algorithm

The here proposed algorithm was designed to detect loops in networks formed by a specific patchy particle model, which are patchy ellipses (introduced in part III) with two oppositely located patches.

Specifically, we characterize the boundary to a pore in a network as a loop. This boundary is a chain of particles enclosing the pore. The loop, as we define it, is not a proper characteristic of the abstract graph, as the abstract graph does not include any spatial information. To explain in more detail, a self-enclosed chain of particles is represented as a circuit in the abstract graph, i.e. a path that starts and ends at the same node and does not visit any node more than once. Determining whether the circuit is a loop that in its particle network counterpart does or does not enclose another loop cannot be determined directly as a well-defined property of the abstract graph.

However, the proposed algorithm identifies loops solely using the abstract graph of the network and no further information on the particles' positions. This is achieved by the algorithm attempting to exploit further characteristics of a loop and its neighbourhood reflected in the graph.

The loop-detection algorithm consists of two parts, where the first part, named the loop-counting algorithm, attempts to count loops for each particle in the net-

work and the second part, the so-called artefact-cleaning algorithm, is responsible for finding artefacts, i.e. falsely detected loops within the loop-counting algorithm:

Loop-Counting Algorithm. For analyzing the loop property, we use the full-network representation (introduced in section 2.4.1) of the particle network. The graph is built using the python package *networkx* [8].
The loop-counting algorithm surveys the environment of each particle p_i to find the loops that include this particle. In this algorithm, it is assumed that each particle can be part of at most two loops. This number could change for a different particle model, and the algorithm might be adapted. Trying to find at most two loops per particle, the algorithm follows these steps:

1. pick a particle p_i, which will be named the base particle
2. the base particle's node, as well as edges to the two patch nodes in the graph is removed
3. the shortest path[1] between the two patch nodes of p_i is searched
4. if a path is found, then save it as the first loop L_1; if no such path is found reiterate from step 1., otherwise continue:
5. find the first branching points along each of the two paths of L_1 starting from $p_{i\mathrm{A}}$ and $p_{i\mathrm{B}}$ (i.e. patch nodes to particle node p_i)
6. all particle nodes between $p_{i\mathrm{A}}$ and $p_{i\mathrm{B}}$ and the respective branching points are deleted.
7. apply the shortest path method once more to possibly find a second path between the two patch nodes $p_{i\mathrm{A}}$ and $p_{i\mathrm{B}}$.
8. if a path is found, then save it as the second loop L_2; if no path is found, exit and start from step 1.

The loop-counting algorithm produces a collection of all detected loops $L_{1,i}$ and $L_{2,i}$ named the loop collection L.

Remarks. A branching point is characterized by three connected patch-nodes, e.g. $p_{1\mathrm{A}}$, $p_{2\mathrm{A}}$ and $p_{3\mathrm{A}}$ representing the patches of three different particles. These three patch-nodes can be connected differently - as three fully connected patch-nodes or as three patch-nodes with only two bonds, as can be seen in fig. 2.5.

[1] using a standard method within the *networkx*-package

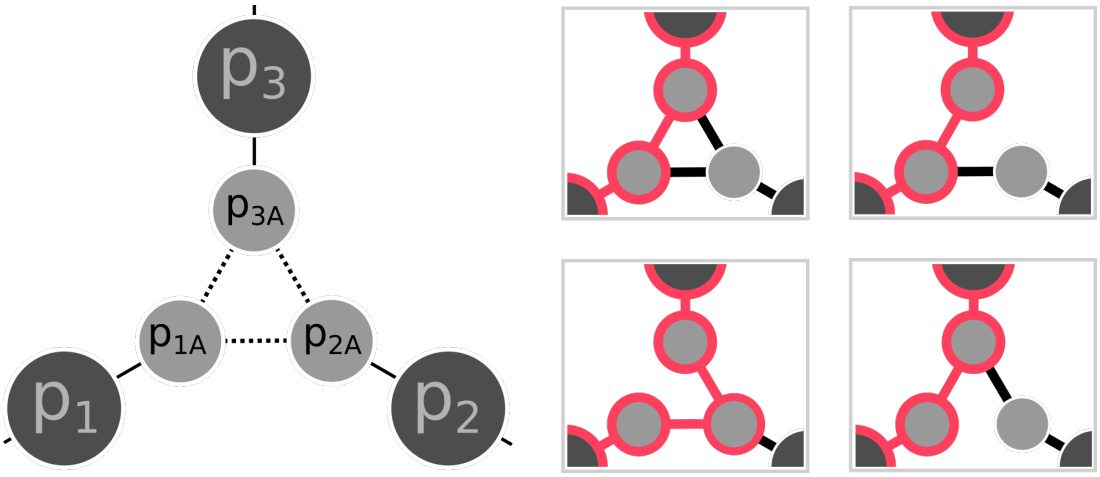


Figure 2.5.: Different scenarios of branching points within the *full* graph. The *full* graph consists of particle nodes (here p_1, p_2 and p_3) as well as patch nodes (here p_{1A}, p_{2A} and p_{3A}). The branching points are searched from the path describing the first loop L_1 marked in red. All four scenarios must be detected as a branching point.

Limitations. Networks in which each particle belongs to exactly two loops can be called pure loop networks. Loop networks consisting of loops with a particular size, e.g. a network of loops with size six is a honeycomb-lattice, is called a pure and monodisperse[2] loop-network. In networks other than pure, monodisperse or nearly monodisperse networks where defects and polydispersity[3] play a big role, the limitations of the algorithm stumbling across those network properties will become evident.
One type of error in the loop-counting algorithm can occur in a not fully condensed network, i.e. when many particles are present that are not forming loops. Here, a branching point that is encountered can, with a higher possibility, lead to a dead end. As a consequence, a second loop following the branched path from this branching point might not be successful, as demonstrated in fig. 2.6. If a second loop is determined, it could be an artefact. As an effect, in specific configurations, some legit loops might not be detected or artefacts generated.

[2]In the sense that loops of only one size exist.
[3]In the sense that loops have dissimilar sizes.

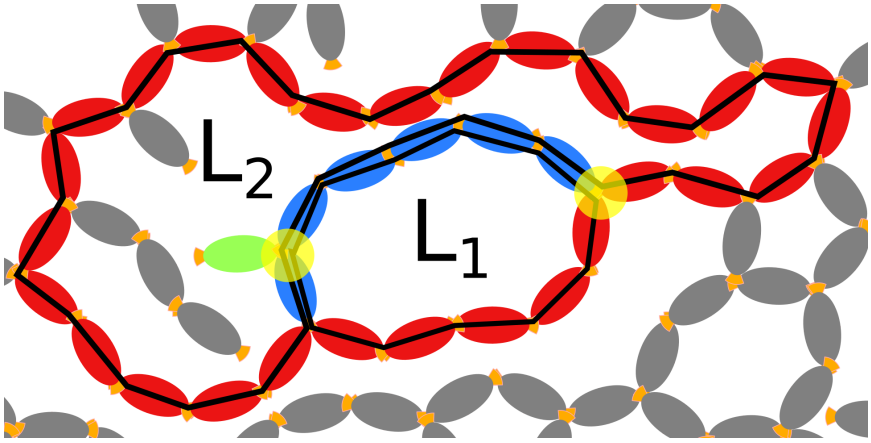


Figure 2.6.: A loop L_2 indicated by the black line that is not detected due to the particle indicated in green. When L_1 has been detected at any of the blue particle's positions, the second loop L_2 will not be found due to the branching point (in yellow) detected at the green particle. This green particle acts as the described 'dead-end'. Therefore L_2 is not found with any of the blue particles as the base particle (Loop-Error 1). Additionally, if the base particle is at any other position in L_2, the true loop L_2 is not detected either, due to the shorter path length of the loop artefact indicated in red.

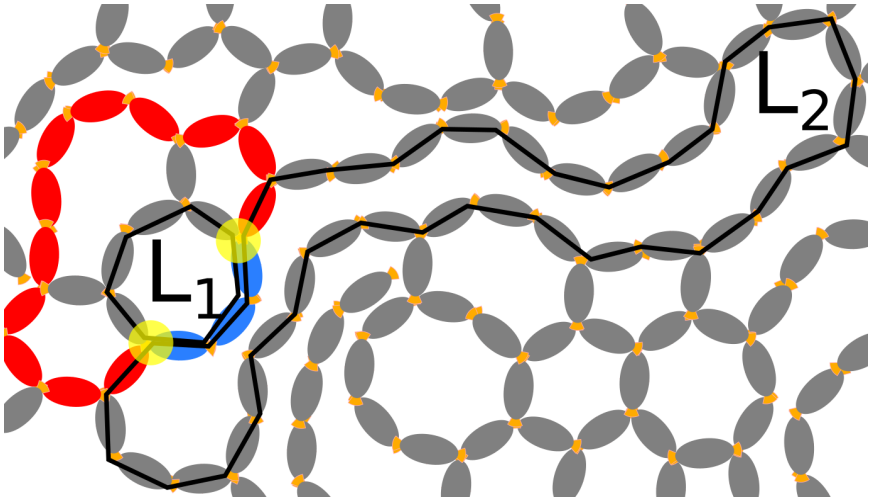


Figure 2.7.: A loop L_2 indicated by the black line is not detected for any of the blue particles as the base particle. This is due to the fact that a shorter path, indicated in red is found. The red particles create the loop artefact.

When loops of very dissimilar sizes are neighbours in the loop network, another type of error might occur. This problem occurs again in trying to find the second loop due to the loop detection relying on the shortest path method. As a demonstrating example, a cluster of small loops neighbouring a very large loop can be considered, and a particle at the border between these two domains. As the first loop, a small loop is detected. In trying to detect the second loop, the shortest path method might not find the second shortest path via the very large loop but rather via the cluster of smaller loops, as can be seen in fig. 2.7.

The described limitations of the counting algorithm create erroneous loops, called loop artefacts. Further, another error might occur, which is a true loop not being detected at all. However, for a loop to be not detected at all means, it is not detected with any of the loops particles as the base particle. Concluding, the error associated with the existence of loop artefacts is predominant.

Therefore the second part of the algorithm - also relying solely on graph properties - deals with the detection of artefacts, i.e. falsely characterized loops from the first part - the loop counting. We find this artefact-cleaning algorithm to be working quite well, but it is still not free of any errors.

Artefact-Cleaning Algorithm. The limitation of the proposed loop detection algorithm for configurations consisting of loops with very dissimilar sizes requires a cleaning procedure of the loop collection from loop artefacts created by the loop-counting algorithm.
As described in the limitations of the loop-counting algorithm (see fig. 2.6 and fig. 2.7), artefacts are created by the algorithm relying on finding the adjacent loop to the base particle by using the shortest path method. It should be noted here that besides creating artefacts, which this part of the algorithm tries to eliminate, the loop-counting algorithm further does not guarantee to find all loops. So even if the cleaning procedure worked perfectly, it could not be guaranteed that the final loop collection is complete.
The cleaning procedure was designed to detect loop artefacts within the primary loop collection L, defined as the set of all loops - legit loops and artefacts - detected by the loop-counting algorithm. Based on this loop collection, the artefact-cleaning algorithm explores the neighbourhood of a selected test loop that will be called l_0. The exploration of this loops neighbourhood is done by building a sub-network, called skeleton, of loops bordering l_0. These loops that border l_O might lie within l_0 or outside of l_0. Further, the algorithm building the skeleton (loop neighbourhood) to loop l_0 guarantees that all skeleton loops lie either inside l_0 or fully outside of l_0. By exploring the size of the skeleton compared to the size of the test loop l_0, a criterion is formulated by which l_0 is either identified as a legit loop or an artefact.

In more detail, the algorithm follows the following steps:

1. start with particle index `i=1`
2. determine $L(p_i) = \{l | p_i \in l\}$, i.e. the set of all loops containing particle p_i
3. if $|L(p_i)| \leq 2$ - the number of loops in $L(p_i)$ is less or equal to two - start from step 1. with `i+=1`
4. if $|L(p_i)| > 2$ - the number of loops in $L(p_i)$ is larger than to two - continue with the next step:
5. all ordered pairs of loops from the set $L(p_i)$ are created, and referred to as `loop0` for the test loop and `loop1`.
6. for each pair, the recursive function `is_artefact` is called to test whether `loop0` is an artefact or not:
   The recursive function `is_artefact` builds the loop neighbourhood, `loops` recursively with `loop1` as the initial loop. The recursive function follows these steps:
   i. The *skeleton particle set*, referred to as `skeleton` is built as the the difference of the particle sets defined by `loops` and `loop0`
   ii. the break conditions are tested in the given order:
      - if the size of `skeleton` is greater than a certain threshold, depending on the loop size of `loop0`, the test loop is **not an artefact**
      - if all particles in `skeleton` have been proposed in step iii. and the size of *skeleton* is smaller than this threshold, the test loop **is an artefact**
      - if in the preceding iteration, another loop was detected as artefact, due to a cross-over with the test loop, both loops are **returned as artefacts**
   iii. if none of the breaking conditions is true, a particle p_j from `skeleton`, that has not yet been checked, is picked
   iv. each loop `loop` from $L(p_j)$ that is new to `skeleton` is tested for a cross-over with the test loop `loop0`
   v. if there is a cross-over detected, both loops `loop` and `loop0` are marked as artefacts
   vi. `checked_parts` is extended by p_j and `loops` is extended by `loop`
   vii. the recursive function `is_artefact` is called with the extended set of `loops`

7. all loops marked as artefact are immediately deleted from the loop collection L and not considered in the following iterations

8. the cleaning-algorithm proceeds by starting over from step 1. with `i+=1`

Remarks: The threshold, mentioned in the breaking conditions for the recursive function `is_artefact`, is estimated as the maximum number of particles that could fit inside the test loop. The estimate assumes a circular shape of the test loop and the particles placed within the test loop at a packing fraction, $\eta = 0.7$. The additional testing for a cross-over of `loop0` with every new loop in step 5. ensures that all neighbouring loops are either fully inside or outside the test loop and `loop0` is indeed an inner or outer boundary of the skeleton. This is hugely important since the criterion of whether a test loop is an artefact or not relies on an estimate of the size of the neighbouring particle network with the test loop `loop0` as the boundary. If the skeleton (i.e. the explored neighbouring particle network) always lies outside the test loop and can be continually grown without a size limit, the test loop is classified as a legit loop. If otherwise, when all particles within the skeleton have been proposed in step iii to expand the neighbourhood, and the skeleton size is still below the described threshold, the test loop is classified as an artefact. Also, the cross-over testing serves as a second artefact criterion: the fact that two closed paths that cross each other contradict the definition of a legit loop is used as such, and both of these paths are classified as artefacts.
Finally, it is important to note that if a test loop `loop0` passes the artefact-cleaning algorithm with one instance for an initialization loop `loop1`, it does not necessarily pass the cleaning for every one of these loops. If a loop is detected as an artefact, it is instantly marked as a real artefact by the algorithm.

Limitations to the algorithm. The proposed loop-detection algorithm, including the loop-counting and artefact-cleaning algorithm, is still not perfect in detecting loops. For specific use of this algorithm, a test configuration should be analyzed manually to see for which configurations the loop detection algorithm is not reliable.

Extension to other systems / Generalization. The proposed algorithm for counting loops was specifically designed for the system presented in part III. The algorithm, as presented here, relies on network and particle characteristics as for the particle model proposed in part III. For example, with two patches, the number of loops adjacent to one particle is two or less. With more than two patches, this number increases, which must be considered in the loop detection algorithm when extending it to other patchy systems.

3. Analysis of Simulation Trajectories

This chapter presents a short summary of methods used in this thesis for the analysis of simulation trajectories. These methods include basic block averaging, as discussed in section 3.1 and analysis of the autocorrelation time section 3.2 as well as error analysis (section 3.3) and truncation analysis (section 3.4).

A Monte-Carlo simulation starts from an initial state and samples neighbouring states in configuration-space. This phase-space trajectory can be visualized via the progression of a collective variable[1] or observable, such as the total energy. Pure visual analysis of the processes of different collective variables (e.g., total energy, volume, ...) can already provide substantial information.
If a chosen collective variable fluctuates around a constant value over the course of the simulation time (mc-time), we can assume that an equilibrium stage is reached. However, a detailed analysis of such scenarios is indispensable and not covered in this thesis.

3.1. Block-Averaging

In specific cases, taking block averages might give better insight into the system's equilibration behaviour. For this analysis, a block size s_{block} is specified, and the simulation trajectory, visualized by using a collective variable, is grouped within subsequent blocks of this size. For each block, the mean and standard deviation of the chosen collective variable is computed. Analyzing the simulation process using the block averages with the help of capturing the fluctuations in the error bars makes the progression of the mean clearer. As a crude condition for equilibration, this type of analysis can be used and from it determined, if for a block size s_{block} similar to the autocorrelation time $\tau_{\text{a.c.}}$ (see section 3.2) the block-averages remain the same over time within the standard error bars.

Block averaging is also applied as a powerful tool to find the best estimate of the error to a sequence of correlated data [16]. As described in this contribution, the correlations will lead to an underestimation of the statistical error, which can be resolved using this rigorous form of block-size analysis. This type of analysis,

[1] a variable of the form $c(\vec{r}^N, \phi^N)$

however, is not used in this thesis. For the presented projects, an error analysis was performed using independent trajectories as described in section 3.3.

3.2. Autocorrelation Time

The auto-correlation time $\tau_{\text{a.c.}}$ is estimated from the temporal auto-correlation function,

$$f_c(t) = \langle c(t_o)c(t_o + t)\rangle_{t_o}, \tag{3.1}$$

by determining the first incident where $f_c(\tau_{\text{a.c.}}) = 0$. Here, the collective variable $c(\vec{r}^N(t), \phi^N(t))$ is stated as $c(t)$. As time we mean the Monte-Carlo time. The quantity t_o is a reference time in the trajectory and $\langle\cdot\rangle_{t_o}$ the average taken over different reference times t_o in this trajectory.
The auto-correlation time $\tau_{\text{a.c.}}$ is used to assess Monte Carlo simulation parameters, such as the total number of cycles and sampling step size. For systems with long correlation times, it is further necessary to perform a number of independent simulations on the same system parameters in order to collect sufficiently many uncorrelated data points.

3.3. Error Analysis

The estimator for the error to the mean $\langle x\rangle$ of an observable x taken from an uncorrelated sample of size n is given as [3, p.31]:

$$err \approx \frac{\sigma_x}{\sqrt{n}} \tag{3.2}$$

where σ_x is the sample standard deviation to $\langle x\rangle$, which is taken as an estimate to the standard deviation σ of the underlying distribution.
When dealing with small sample sizes, the estimate given in eq. (3.2) will underestimate the true error [17]. Here it is common to use the sample standard deviation as the estimator of the error:

$$err \approx \sigma_x \tag{3.3}$$

For both types of error estimators, eq. (3.2) and eq. (3.3), it is important to use an uncorrelated sample. If the sample is correlated, both of these estimators will underestimate the true error.
MC-simulation trajectories can be subjected to large correlations. Different methods for dealing with strong correlations exist: The error estimate can be corrected using the auto-correlation time [3, p.32]. Another method would use block-size analysis to find the best estimate for the error to a correlated trajectory [16]. Generally, when strong correlations emerge, it is necessary to run several

independent simulations. The averages from data acquired from independent MC-runs are used to build an uncorrelated sample.

3.4. Truncation Analysis

The sampling of configurations in a simulation process starts after the equilibration phase, `disc_cycles` (→ `module_mc.F90`). The equilibration phase in the simulation method is used to adjust simulation parameters, such as maximum displacements. The system itself might not be yet equilibrated when the sampling starts. When simulating self-assembly processes such as nucleation, the onset of the cluster growth is non-deterministic and follows stochastic principles. Hence the true equilibrium configuration will be reached at a later point in the simulation process. In these cases, the statistical analysis of the target structure will be performed over a truncated trajectory with the truncation point chosen where the target structure is reached. The truncation point tp is given as a number within $(0, 1)$, indicating a fraction of the total simulation length. Here, the location of this truncation point was determined by visual analysis of the evolution of some observable over the course of the simulation time. Similarly, a study of the mean's dependency on the truncation point can be performed by truncating the trajectory at several truncation points tp. The truncation point at which the mean reaches a plateau should then be evaluated.

4. Free Energy Computations

In Statistical Physics, the Helmholtz free energy is the natural thermodynamic potential of the canonical ensemble. Hence the free energy function of a system provides complete information. Here, we try to compute the free energies for specific configurations in order to determine the true ground state at a particular thermodynamic state point.

More advanced simulation techniques must be used to uncover the Helmholtz free energy of a thermodynamic state. Here I will cover some basic methods which were used within this thesis, namely *Umbrella Sampling*, *Thermodynamic-* and *Hamiltonian Integration*. All these techniques are standard methods and are known from literature [1][2][3].

The next method covered in this chapter is the *Einstein Molecule Method* (see section 4.4), which combines *Umbrella Sampling* and *Hamiltonian Integration* and was developed by Noya and Vega [18, 19] as a variation of the original method, published by Frenkel and Ladd in 1984 [20]. This method is used to compute free energies of solids using the Einstein Crystal as a reference system [1, p. 243]. As an extension to the published methods, I will propose the use of the Möbius-transformation technique, section 4.4.3 in the numerical integration part of the Einstein Molecule method that allows for fewer integration points by restricting the sampling on the important parts of the integrand and thus leads to faster convergence.

Finally, section 4.5 covers a specific correction of finite-size effects arising in the simulation-based methods.

4.1. Thermodynamic Integration

4.1.1. Integration along Isotherms

To determine thermodynamic potentials like the Helmholtz free energy, one can use the partial derivatives of the potential and then perform a numeric simulation for sampling points along an integration path. A numeric integration technique will then determine potential differences [1, p. 242].
The thermodynamic integration relation for integrating along an isotherm can

be derived from the differential form of the Helmholtz free energy [21]:

$$dA = d(U - TS) \tag{4.1}$$

Together with the first, $dU = dQ - pdV$, and second law of thermodynamics for reversible processes, $dQ = TdS$, the total differential of the Helmholtz free energy can be written as:

$$dA = -pdV + SdT \tag{4.2}$$

and hence, the partial derivative of A for a constant temperature process ($T = const$) is

$$-p = \left(\frac{\partial A}{\partial V}\right)_T. \tag{4.3}$$

This relation integrated along an isotherm gives

$$A(V_2, T) = A(V_1, T) - \int_{V_1}^{V_2} p(V, T)dV \tag{4.4}$$

If a system with a known Helmholtz free energy can be identified as state 1, the free energy of state 2 can be computed by computing a series of ensemble averages from Monte Carlo simulations, either in the NVT ensemble (with the integrand = $\langle p \rangle dV$) or NpT ensemble (with the integrand = $pd\langle V \rangle$), to evaluate the integral.
For two-dimensional systems and in reduced units the above expression can be reformulated:

$$A(\rho_2^*, \beta) = A(\rho_1^*, \beta) + \int_{\rho_1^*}^{\rho_2^*} \frac{p^*(\rho^*, \beta)}{\eta_0 {\rho^*}^2} d\rho^*, \tag{4.5}$$

with η_0 the packing fraction at closest packing and ρ^* the reduced number density of particles. Further the pressure p^* in its reduced, dimensionless form is given by $p^* = \frac{A_{\text{part}} p}{k_B T}$, with A_{part} as the area of a particle in two dimensions..

4.1.2. Integration along Isochores

In a similar fashion as for isotherms, the thermodynamic integration rule for isochores can be derived [1, p. 242].
Via the partition function in the canonical ensemble, $Z(N, V, T)$ and its relation to the free energy $A = -k_B T \ln Z(N, V, T)$, the relation:

$$U = \frac{\partial(\beta A)}{\partial \beta} \tag{4.6}$$

can be derived, assuming $U = \langle \mathcal{V} \rangle$, where $\mathcal{V}$ describes the potential energy within the hamiltonian $\mathcal{H} = \mathcal{T} + \mathcal{V}$. By integrating both sides, this relation can be used to express a free energy difference by an integral over temperatures:

$$\frac{A(V,T_2)}{k_\mathrm{B}T_2} = \frac{A(V,T_1)}{k_\mathrm{B}T_1} - \int_{T_1}^{T_2} \frac{U(T)}{k_\mathrm{B}T^2} dT. \tag{4.7}$$

4.2. Umbrella Sampling

The free energy difference between two states, now called state 1 and state 2 can also be expressed via the systems partition functions Z_1 and Z_2 by [1, p. 192]:

$$A_2 - A_1 = -k_\mathrm{B}T \ln\left[\frac{Z_2}{Z_1}\right] \tag{4.8}$$

$$= -k_\mathrm{B}T \ln\left[\frac{\frac{1}{N!}\int d\vec{r}^{\,N} \exp\left(-\beta U_2(\vec{r}^{\,N})\right)}{\frac{1}{N!}\int d\vec{r}^{\,N} \exp\left(-\beta U_1(\vec{r}^{\,N})\right)}\right] \tag{4.9}$$

$$= -k_\mathrm{B}T \ln\left[\frac{\int d\vec{r}^{\,N} \exp\left(-\beta(U_2(\vec{r}^{\,N}) - U_1(\vec{r}^{\,N})\right) \exp\left(-\beta U_1(\vec{r}^{\,N})\right)}{\int d\vec{r}^{\,N} \exp\left(-\beta U_1(\vec{r}^{\,N})\right)}\right] \tag{4.10}$$

$$= -k_\mathrm{B}T \ln\left\langle \exp\left(-\beta(U_2(\vec{r}^{\,N}) - U_1(\vec{r}^{\,N}))\right)\right\rangle_1 \tag{4.11}$$

This expression can be used to compute the Helmholtz free energy difference by sampling in only one state, here, state 1. The expression, however, is only valid if the two states satisfy the condition that their probability distributions have sufficient overlap - meaning that regions of high probability in configuration space are the same. In other terms, the behaviour of the two systems must be similar. If two systems with distinct probability distributions are to be compared, the relation can be established by dividing the whole path into separate steps and applying umbrella sampling for each step [1, p. 193].

4.3. Hamiltonian Integration

As shown in [2, p. 298] a system can be transferred from one state A into another state B with respective Hamiltonians $\mathcal{H}_A$ and $\mathcal{H}_B$. The process underlying Hamiltonian integration is described by:

$$\mathcal{H}(\lambda) = (1-\lambda)\mathcal{H}_A + \lambda\mathcal{H}_B \tag{4.12}$$

with $\lambda \in [0,1]$.
If only the interaction energies U changes from state A to B then $\mathcal{H}(\lambda)$ can be

rewritten as:

$$\mathcal{H}(\lambda) = T + (1-\lambda)U_A + \lambda U_B \tag{4.13}$$

with the kinetic energy $T = \sum_i p_i^2/2m$ and $U(\lambda) = (1-\lambda)U_A + \lambda U_B$ being the interaction potential.

The Helmholtz free energy of the combined system is then given by:

$$\begin{aligned} A(N,V,T,\lambda) &= -k_\mathrm{B}T \ln Z & (4.14)\\ &= -k_\mathrm{B}T \ln\Big(\frac{1}{N!}\int \exp\left(-\beta\mathcal{H}(\lambda)\right) d\vec{r}^N d\vec{p}^N\Big) & (4.15)\\ &= -k_\mathrm{B}T \ln\Big(\frac{\Lambda_{dB}}{N!}\int \exp\left(-\beta U(\lambda)\right) d\vec{r}^N\Big). & (4.16) \end{aligned}$$

It can be seen that only the interaction potential $U(\lambda)$ remains as an explicit dependency in the expression for the free energy A. The integration over all momenta of the N particles yields the thermal de-Broglie wavelength Λ_{dB} which will be set to 1 from here on.

In order to find an expression for the Helmholtz free energy that can be used in simulation, the derivative of the free energy is taken with respect to λ:

$$\begin{aligned} \frac{\partial A(N,V,T,\lambda)}{\partial\lambda} &= -k_\mathrm{B}T\frac{1}{Z}\frac{1}{N!}\int(-\beta)\frac{\partial U(\lambda)}{\partial\lambda}\exp\left(-\beta U(\lambda)\right)d\vec{x}^N & (4.17)\\ &= \frac{1}{Z}\frac{1}{N!}\int\frac{\partial U(\lambda)}{\partial\lambda}\exp\left(-\beta U(\lambda)\right)d\vec{x}^N & (4.18)\\ &= \Big\langle\frac{\partial U}{\partial\lambda}\Big\rangle_{N,V,T,\lambda}. & (4.19) \end{aligned}$$

By integrating both sides over the interval $\lambda \in [0,1]$, we arrive at:

$$A(N,V,T,\lambda=1) = A(N,V,T,\lambda=0) + \int_{\lambda=0}^{1}\Big\langle\frac{\partial U}{\partial\lambda}\Big\rangle_{N,V,T,\lambda} d\lambda. \tag{4.20}$$

This expression is useful for simulations since the integrand,

$$\begin{aligned} \Big\langle\frac{\partial U}{\partial\lambda}\Big\rangle_{N,V,T,\lambda} &= \langle U_B - U_A\rangle_{N,V,T,\lambda} & (4.21)\\ & & (4.22) \end{aligned}$$

is easily accessible by means of simulation.

Now that *umbrella sampling* and *Hamiltonian integration* were discussed in a

general manner, the following section will deal with the application of those two methods to create a suitable integration path to compute Helmholtz free energies for solids.

4.4. Einstein Molecule Method

The *Einstein Molecule Method* [18][19] uses *umbrella sampling* and *Hamiltonian integration* (as described above) to establish a path from the ideal Einstein Molecule as a reference state to the target structure, which is a particle configuration in a solid-state. Using this approach, the Helmholtz free energy splits into the following constituents:

$$A = A_0 + \Delta A_1 + \Delta A_2 \tag{4.23}$$

where A_0 denotes the Helmholtz free energy of the reference state and ΔA_1 and ΔA_2 account for an *umbrella sampling* step and a step of *Hamiltonian integration* respectively along the path from the reference system to the wanted structure.

4.4.1. Reference State

The reference state, as proposed in the original version [20] of this method, is the Einstein Crystal of N particles in a defined lattice configuration. The particles are bound to their fixed lattice sites by harmonic springs. The lattice sites comprise an equilibrium position for the locations as well as for the orientations for all particles. In the Einstein Molecule method, the reference state is the Einstein Molecule, which resembles the Einstein Crystal very closely: the lattice sites are not fixed but are carried along with reference to one particle (particle with index $i = 1$), which therefore is named the carrier particle. All lattice positions are referenced to this particle's position. This special form was named Einstein Molecule by Vega and Noya in their original publication of the Einstein Molecule method [18].

The Hamiltonian, used as the Einstein Molecule Hamiltonian within this thesis, with spring constant Λ_E is given by:

$$\mathcal{H}_\text{ein-mol} = \sum_{i=1}^{N} \frac{\vec{p}_i{}^2}{2m} + \Lambda_\mathrm{E}\Big(\sum_{i=2}^{N} (\vec{r}_i - \vec{r}_{i,0})^2 + \sum_{i=1}^{N} (\phi_i - \phi_{i,0})^2 \Big). \tag{4.24}$$

Note that the momenta are considered for all particles, but the components of the harmonic potential are zero for the first particle since it is the carrier of the lattice.

Orientational Einstein Energy. In the Einstein Molecule Hamiltonian in eq. (4.24), the orientational part of the potential energy was chosen as $\propto \Delta\phi_i^2$, reflecting a $\mathcal{C}1$-symmetry of the underlying particle. This part of the potential must always be chosen according to the particle's rotational symmetry. For more information and potentials reflecting different types of rotational symmetries, see [22].

4.4.2. Procedure

The Einstein Molecule method aims at transferring the Hamiltonian of an ideal Einstein Molecule into the Hamiltonian for the wanted solid system step by step. By tracing the changes in the Helmholtz free energy along each step, the free energy difference between the reference and the final system can be computed. The steps that are taken in the transformation are the following:

0.1 Ideal Einstein Molecule

0.2 constraint Einstein Molecule (with position of the carrier particle fixed)

1.0 constraint Einstein Molecule with solid interaction (add solid interactions)

2.1 constraint solid (decrease Einstein Molecule interactions stepwise)

2.2 free solid (release carrier particle)

reference system [0.1]

The reference system is the ideal Einstein Molecule. The Helmholtz free energy of the ideal Einstein Molecule of point particles moving and rotating in two dimensions is given by:

$$A_{\text{id\_ein}}[Nk_BT] = \frac{1}{N}\ln\rho + \left(1-\frac{1}{N}\right)\ln\left(\frac{\beta\Lambda_{\text{E}}}{\pi}\right) + \frac{1}{2}\ln\left(\frac{\beta\Lambda_{\text{E}}}{\pi}\right) + \ln(2\pi), \quad (4.25)$$

accounting for A_0 in eq. (4.23). Here the thermal de-Broglie wavelength $\Lambda_{\text{dB}} = 1$. Details on the derivation of this expression can be found in appendix A.1.

step 1 [$0.1 \to 0.2$]

Constraining the free Einstein Molecule by fixing the position of the carrier particle corresponds to subtracting the respective term in the Helmholtz free energy due to the movement of the carrier particle:

$$A_{\text{fix\_ein}} - A_{\text{id\_ein}} = -k_{\text{B}}T\ln\left(\frac{Z_{\text{fix\_ein}}}{Z_{\text{id\_ein}}}\right) \quad (4.26)$$

$$= k_{\text{B}}T\ln\left(\frac{V}{\Lambda_{\text{dB}}}\right) \quad (4.27)$$

This step is specific to the Einstein Molecule method, in contrast to the original method proposed by Frenkel and Ladd.

step 2 $[0.2 \rightarrow 1.0]$

In this step, the solid interaction is added on top of the constraint Einstein Molecule interactions. The induced free energy change is computed using umbrella sampling. Hence:

$$A_{\text{fix\_ein,sol}} - A_{\text{fix\_ein}} = -k_{\text{B}}T \ln \left\langle \exp\left(-\beta(U_{\text{fix\_ein,sol}} - U_{\text{fix\_ein}})\right)\right\rangle_{\text{fix\_ein}} \tag{4.28}$$

$$= -k_{\text{B}}T \ln \left\langle \exp\left(-\beta U_{\text{sol}}\right)\right\rangle_{\text{fix\_ein}} \tag{4.29}$$

step 3 $[1.0 \rightarrow 2.1]$

In this step the Einstein field is gradually turned off by increasing λ from 0 to 1. The change in the free energy is computed via Hamiltonian integration - with interactions $U_{\text{A}} = U_{\text{sol}} + U_{\text{fix\_ein}}$ in state A and $U_{\text{B}} = U_{\text{fix\_ein}}$ in state B. The interaction energy of the combined system is thus $U(\lambda) = U_{\text{sol}} + (1-\lambda)U_{\text{fix\_ein}}$:

$$A_{\text{fix\_sol}} - A_{\text{fix\_ein,sol}} = -\int_{\lambda=0}^{1} \langle U_{\text{ein}} \rangle_{N,V,T,\lambda} d\lambda \tag{4.30}$$

The integration parameter λ is used to tune the Einstein field spring constant Λ_{E}. In practice the integral in eq. (4.30) is evaluated numerically. Potentially a suitable transformation method of the integration parameter λ can be used.

step 4 $[2.1 \rightarrow 2.2]$

In the last step the carrier particle is released again. The carrier particle is able to move freely throughout the box and the solid becomes the unrestricted target solid. The corresponding free energy difference is given by:

$$A_{\text{sol}} - A_{\text{fix\_sol}} = -k_{\text{B}}T \ln\left(\frac{V}{\Lambda_{\text{dB}}}\right) \tag{4.31}$$

Summarizing all steps $[0.1 \rightarrow 2.2]$

Adding up the free energy differences of all steps, the resulting Helmholtz free energy for the target state becomes:

$$A_{\mathrm{sol}} = \underbrace{A_{\mathrm{id\_ein}}}_{=A_0} + \underbrace{A_{\mathrm{fix\_ein}} - A_{\mathrm{id\_ein}}}_{=k_{\mathrm{B}}T\ln(\frac{V}{\Lambda_{\mathrm{dB}}})} + \underbrace{A_{\mathrm{fix\_ein,sol}} - A_{\mathrm{fix\_ein}}}_{=\Delta A_1} + \tag{4.32}$$

$$\tag{4.33}$$

$$+ \underbrace{A_{\mathrm{fix\_sol}} - A_{\mathrm{fix\_ein,sol}}}_{=\Delta A_2} + \underbrace{A_{\mathrm{sol}} - A_{\mathrm{fix\_sol}}}_{=-k_{\mathrm{B}}T\ln(\frac{V}{\Lambda_{\mathrm{dB}}})} \tag{4.34}$$

and finally:

$$A_{\mathrm{sol}} = A_0 + \Delta A_1 + \Delta A_2 \tag{4.35}$$

with the terms now in units of $Nk_{\mathrm{B}}T$ given by:

$$\frac{A_0}{Nk_{\mathrm{B}}T} = \frac{1}{N}\ln(\rho) + \left(\frac{3}{2} - \frac{1}{N}\right)\ln(\frac{\beta\Lambda_{\mathrm{E}}}{\pi}) + \ln(2\pi) \tag{4.36a}$$

$$\frac{\Delta A_1}{Nk_{\mathrm{B}}T} = -\frac{1}{N}\ln\left\langle \exp(-\beta U_{\mathrm{sol}})\right\rangle_{\mathrm{fix\_ein}} \tag{4.36b}$$

$$\frac{\Delta A_2}{Nk_{\mathrm{B}}T} = -\frac{1}{Nk_{\mathrm{B}}T}\int_{\lambda=0}^{1}\langle U_{\mathrm{ein}}\rangle_{N,V,T,\lambda}d\lambda \tag{4.36c}$$

4.4.3. Practical Realisation

Numerical evaluation ΔA_1

For the numerical evaluation of ΔA_1 a trick is used proposed in ref. [18]. The expression for ΔA_1 in eq. (4.36b) can reach very large values due to overlapping particles in the sampled configurations. To avoid this problem the solid potential energy in the exponent, U_{sol}, is shifted by the constant ground state energy of the target lattice structure, U_{lat}. The final expression then reads:

$$\Delta A_1 = U_{\mathrm{lat}} - k_{\mathrm{B}}T\ln\left\langle \exp(-\beta(U_{\mathrm{sol}} - U_{\mathrm{lat}}))\right\rangle_{\mathrm{fix\_ein}} \tag{4.37}$$

which can be handled better numerically.

Choice of Λ_{E}

The numerical values of the analytical term A_0, as well as both terms ΔA_1 and ΔA_2 determined from simulation, depend explicitly on the Einstein spring constant Λ_{E}. However, the final result for the total Helmholtz free energy $A(N, \rho, T)$ must not depend on a specific choice of the Einstein spring constant Λ_{E}. In prac-

tice, this requirement cannot be strictly fulfilled as a consequence of different numerical issues, as discussed below. Hence one has to choose a range of values for the spring constant within a range for which this is true, must be chosen.
Taking a too small spring constant Λ_E leads to numerical errors in the final results because the Einstein Molecule [**state 0.1**] and the mixed system with Einstein Molecule and solid interaction [**state 0.2**] will have probability distributions in phase space that have not sufficient overlap. Hence the chosen states are not suitable for umbrella sampling (see section 4.2). On the other hand, a spring constant Λ_E that is too large will create an unnecessarily large interval for the Hamiltonian integration in ΔA_2 and therefore will demand a higher amount of computational resources for the same accuracy of determining the numerical integral.
Hence the spring constant Λ_E should be chosen in between these two extremes. Moreover, in a correct application, the independence of the total Helmholtz free energy of the spring constant Λ_E must be checked in applying the Einstein Molecule method and if proven right, indicates a suitable choice of spring constant and a reliable result for the Helmholtz free energy.

Scaling of the spring constants

Studying the model of hard disks in the hexagonal lattice configuration, a nice behaviour of function $\Delta A_1(\Lambda_E, \rho^* = const)$ depending on the spring constant Λ_E at a constant reduced particle density ρ^* was observed. When introducing a scaling to the spring constants according to

$$\Lambda'_E = (\frac{1}{\sqrt{\rho^*}} - 1)^2 \Lambda_E, \tag{4.38}$$

it can be observed, that all $\Delta A_1(\Lambda'_E, \rho^*)$ will fall on the same master-curve for different ρ^*. This can be understood, as $\frac{1}{\sqrt{\rho^*}}$ is the lattice constant in the hexagonal lattice, i.e. the inter-particle distance of neighbouring disks, this value subtracted by the disk diameter $\sigma = 1$ gives the surface to surface distance of two neighbouring disks. And this distance is entering the scaling factor. The same observations in three dimensions were made in a study on hard spheres by [23].

Numerical Integration Methods - ΔA_2

The integral ΔA_2, see eq. (4.36c), is rewritten as:

$$\frac{\Delta A_2}{N k_B T} = -\frac{1}{N k_B T} \int_0^{\Lambda_E} [\langle \Delta R^2 \rangle_\lambda + \langle \Delta \Phi^2 \rangle_\lambda] d(\lambda \cdot \Lambda_E) \tag{4.39}$$

with $\langle \Delta R^2 \rangle_\lambda$ and $\langle \Delta \Phi^2 \rangle_\lambda$ as the mean-squared displacements of the particles from the lattice sites and orientations at state λ respectively (omitting the thermodynamic state N, V, T in the notation). The numerical integration is carried out

over n_{sp} in the interval $[0, \Lambda_E]$ sampling points creating a partition of the interval of integration into sub-intervals. The accuracy of the numerically computed value of ΔA_2 depends on **1)** the number of sample points n_{sp}, **2)** the value of the spring constant Λ_E as the upper bound of the interval of integration $[0, \Lambda_E]$, **3)** the specific numerical integration method used, possibly including **4)** additional refined methods of numerical integration, using transformations of the integrand.
Concerning the last two points - within the projects of this thesis the numerical integration method used is the *Gauss-Legendre quadrature* [24, p. 144]. Due to the particular shape of the integrand (see fig. 4.1) different transformations of the integrand prior to applying the numerical integration method were applied:

Depending on the underlying system, the dimension, and the particle model, the integrand in ΔA_2 can display a very large peak Making a straightforward application of any numerical integration method and gaining converging results is very hard and time-consuming.
Within this thesis, the integrand usually showed a very pronounced, strongly localized peak at $\lambda = 0$: This is due to the fact that for this λ-value the particles regain their full mobility when losing their constraint to their lattice positions via the spring constants. Here the mean squared displacement of the particles might grow considerably. This effect can be clearly seen in part II.
No standard numerical integration method with a fixed sampling point distribution is able to cope with such a pronounced peak unless an extremely fine grid is used; however, such a procedure is highly inefficient as it is computationally expensive, and the high density of grid points is redundant for a big part of the λ-range, where the integrand is flatter. To remedy this shortcoming, a suitable transformation of the λ-axis is needed that stretches areas under the peak and compresses areas with an almost constant integrand. In this way, the sample points are re-distributed along the λ-axis to regions where they are required. Two possible transformation methods are presented here:

Logarithmic transformation A logarithmic transformation of the integration variable λ for applications of the Einstein Molecule method was proposed in [1, p. 260] and used in [18][19]. Using a transformation parameter c, the transformation is given by:

$$\lambda \longrightarrow \ln(\lambda + c). \tag{4.40}$$

Any constant factor of λ (e.g. by Λ_E) will only shift this transformation by a constant and will therefore leave the results unaffected:

$$\lambda \longrightarrow \ln(\Lambda_E \lambda + c') = \ln(\Lambda_E) + \ln(\lambda + c). \tag{4.41}$$

The success of this transformation, therefore, depends on a suitable choice of c and on the number of sampling points n_{sp}.

Möbius-transformation. The Möbius transformation presented in [25] applied to the integration variable λ effectively distributes the majority of sampling points at the peak. The Möbius transformation applies to an interval $[-1, 1]$. Hence in a first step a linear transformation $[0, \Lambda_E] \to [-1, 1]$ is applied by transformation rule $\lambda \to x = (2\lambda - 1)$ is applied:

$$\Delta A_2 = -\frac{1}{2N} \int_{-1}^{1} [\langle \Delta R^2 \rangle_\lambda + \langle \Delta \Phi^2 \rangle_\lambda] d(2\lambda - 1). \tag{4.42}$$

This integrand with $x \in [-1, 1]$ and its peak at $x_0 = -1$ is then subjected to the Möbius transformation $\chi : x \mapsto u$ with transformation parameter ξ:

$$u = \chi(x, \xi) = \frac{x - \xi}{1 - x\xi}. \tag{4.43}$$

The according inverse-transformation $\phi : u \mapsto x$ reads:

$$x = \phi(u, \xi) = \frac{u + \xi}{1 + u\xi}. \tag{4.44}$$

With this transformation the integral takes the form:

$$\Delta A2 = -\frac{1}{2N} \int_{-1}^{1} [\langle \Delta R^2 \rangle_\lambda + \langle \Delta \Phi^2 \rangle_\lambda] \phi'(u) du \tag{4.45}$$

$$= -\frac{1}{2N} \int_{-1}^{1} [\langle \Delta R^2 \rangle_\lambda + \langle \Delta \Phi^2 \rangle_\lambda] \frac{1 - \xi^2}{(1 + u\xi)^2} du. \tag{4.46}$$

Lastly, the Gauss-Legendre quadrature rule is applied to the integral with weights w_i and values for the abscissae u_i:

$$G_n = \sum_{i=1}^{n} w_i g(u_i) \tag{4.47}$$

The effect of the Möbius integration on the integrand form can be seen in an example in fig. 4.1. The integrand is a taken from a system of hard ellipses treated in part II.

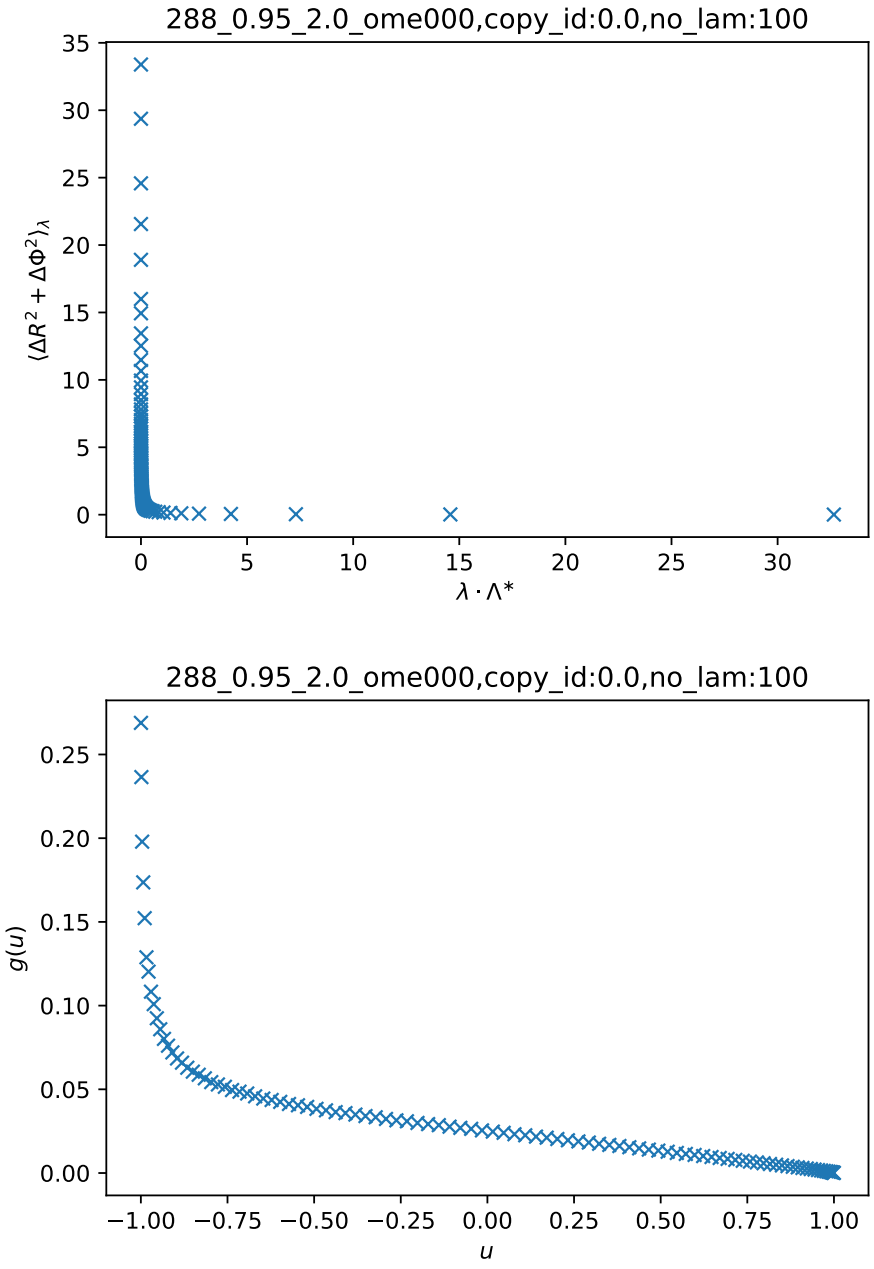


Figure 4.1.: Integrand using $n_{sp} = 100$ grid points before (top) and after (bottom) a Möbius transformation of the integrand. The Möbius transformation was applied with parameter $\xi = 0.993$ for a system of $N = 288$ particles in lattice state $\omega = 30°$, full information on the underlying system in part II - chapter 4.

4.5. Finite-Size Corrections

Computation of the free energy based on numerical simulations is subjected to finite-size effects. The proposed method determines the free energy of a solid only for a specific system size N, usually in a simulation box with periodic boundary conditions. When applying finite-size corrections, for which Bruce and Wilding give a precise summary in [26, p. 22], the object of interest is the thermodynamic limit, i.e. $N \to \infty$, for the intensive form of the Helmholtz free energy $N^{-1}A(N)$, also named the free energy density. To compute the thermodynamic limit properly from free energies at finite ensemble sizes, finite-size corrections must be applied. To achieve this goal, several computations of the free energy at different system sizes are necessary. The size dependency can then be studied by using fitting models or a convergence criterion.

If it is observed that the free energy is not dependent on the size N, it is possible to directly determine the thermodynamic limit from simulation results. If a system size is reached that yields the same result for the intrinsic function $N^{-1}A(N)$ as a considerably smaller system, it can be concluded that the thermodynamic limit has been reached (as observed in [10, p. 3004]).

Usually, such a large system size cannot be simulated within a reasonable time span. Here finite-size corrections and fitting procedures must be applied to determine the thermodynamic limit [26]. For the finite-size correction, a function $f^{f.s.}(N^{-1})$ is proposed that can be fitted to the simulation data of the free energy $\beta N^{-1}A(N)$, and the thermodynamic limit is then approached through extrapolation. Finite-size corrections are discussed and proposed in literature [27][23].

For the problem of hard ellipse lattices (see part II - chapter 4) specifically, we propose a fitting function for the free energy of the form :

$$A_\alpha^{f.s.}(N)/Nk_\mathrm{B}T = a_\alpha + c_\alpha \cdot \frac{N^{b-1}}{\kappa_{\mathrm{box},\alpha}} \tag{4.48}$$

with $b < 1$, α indicating the type of solid phase and $\kappa_{\mathrm{box},\alpha}$ giving the aspect ratio of the simulation box as fraction of L_x over L_y.

The application of this particular finite-size correction to the free energy differences can be followed in part II - chapter 5.

5. Structural Analysis

Analysis of structural information of configurations is performed in post-processing. In the structural analysis, spatial and orientational correlations between the particles are identified. Standard distribution functions such as the pair-distribution and radial distribution are discussed. Suitable order parameters are used to further analyze the structures. Definition and implementation of the nematic order parameter are discussed. In addition to these standard methods, a new order parameter for loop networks is proposed. The computation of the loop order parameter relies on the also newly proposed loop-detection algorithm presented in chapter 2 - section 2.4.

5.1. Pair- and Radial Distribution Function

Definition The 2-particle distribution function or pair-distribution function $g^{(2)}(\vec{r}_1, \vec{r}_2)$ is defined in Hansen and McDonald [28, p. 33] via the n-particle densities, $\rho_N^{(n)}(\vec{r}^{\,n})$, specifically by the 1- and 2-particle densities $\rho_N^{(1)}$ and $\rho_N^{(2)}$, by:

$$g^{(2)}(\vec{r}_1, \vec{r}_2) = \frac{\rho_N^{(2)}(\vec{r}_1, \vec{r}_2)}{\rho_N^{(1)}(\vec{r}_1)\rho_N^{(1)}(\vec{r}_2)}, \tag{5.1}$$

In [28] it is further shown that from eq. (5.1) the radial distribution function, $g(r = |\vec{r}_1 - \vec{r}_2|)$ can be expressed for homogeneous and isotropic systems via δ-distributions of the two particle positions as:

$$g(r) = \frac{1}{\rho N}\Big\langle \sum_{i=1}^{N}\sum_{\substack{j=1 \\ j\neq i}}^{N} \delta(\vec{r} - \vec{r}_j + \vec{r}_i)\Big\rangle. \tag{5.2}$$

With simulation results at hand, a different definition of the radial distribution function on a discretized r-axis, is more useful. Using central bins $b_l = (l + \frac{1}{2})\Delta b$ of size Δb, the definition reads [2, p.272]:

$$g(b_l = r \pm l\frac{\Delta b}{2}) := \frac{n(b_l)}{n^{id}(b_l)}, \tag{5.3}$$

with $n(b_l)$ being the average number of particles in bin b_l, given as $n(b_l) = \text{hist}(b)/\text{N}/\#\text{conf}$, and $n^{id}(b_l)$ as the corresponding quantity in the ideal gas,

$n^{id}(b) = \rho \Delta A$. The radial distribution function therefore captures the structures deviation from the ideal gas, i.e. random distribution [28].

The two definitions in eq. (5.2) and eq. (5.3) can be related by taking some mathematical reformulation. The steps to be taken are the discretization of the r-axis via binning, $b_l = (l + \frac{1}{2})\Delta b$ with $l \in \mathbb{Z}^+$. Also the ensemble average is approximated by the sample mean $\langle A \rangle \approx \frac{1}{n} \sum_{k=1}^{n} A(\mu_k)$ of a sample of n configurations $\mu_k = (\vec{r}_{1,k}, ..., \vec{r}_{N,k})$. Following the principles of importance sampling [4], the sample has to be generated from a probability distribution $p(\mu) \sim \frac{\exp(-\beta\Phi(\mu))}{\int \exp(-\beta\Phi(\mu)) d\mu}$. Evaluation of the summation over δ-distribution, $\delta(r - |\vec{r}_j - \vec{r}_i|)$ comes down to creating the histogram hist(b_l) of particle pairs separated by distances $r = |\vec{r}_j - \vec{r}_i|$. In creating the histogram hist(b_l) it has to be considered that the probability of counting pair separations r of a specific bin b_l is also depending on the bin's area $\Delta A(b_l)$. The bin area grows with the radius, and in the histogram this effect has to be repaired by normalizing to a count per unit-area.

5.2. Nematic Order Parameter

The nematic order parameter in two dimensions is defined as the largest eigenvalue and the director of the field (or orientations) as the corresponding eigenvector of the matrix $Q_{\alpha\beta}$ [29][30]:

$$Q_{\alpha\beta} = \frac{1}{N} \sum_{i=1}^{N} \left[2u_\alpha(i) u_\beta(i) - \delta_{\alpha\beta} \right] \tag{5.4}$$

$$= \frac{1}{N} \sum_{i=1}^{N} Q^i_{\alpha\beta} \tag{5.5}$$

Here N is the number of particles, $\alpha, \beta = (1, 2)$ indicates the first or second cartesian coordinate and $\vec{u}_(i)$ is the director of the i-th particle. The matrix $Q^i_{\alpha\beta}$ takes the following form:

$$Q^i_{\alpha\beta} = 2u_\alpha(i) u_\beta(i) - \delta_{\alpha\beta} \tag{5.6}$$

$$= \begin{bmatrix} 2u_1^2 - 1 & 2u_1 u_2 \\ 2u_1 u_2 & 2u_2^2 - 1 \end{bmatrix}. \tag{5.7}$$

Using parameterization in polar coordinates,

$$u_1(i) = \vec{u}(i) \cdot \hat{e}_1 = |\vec{u}(i)| \cos(\phi) = \cos(\phi) \tag{5.8}$$

$$u_2(i) = \vec{u}(i) \cdot \hat{e}_2 = |\vec{u}(i)| \sin(\phi) = \sin(\phi) \tag{5.9}$$

for substitution, the matrix terms then give:

$$Q^i_{\alpha\beta} = \begin{bmatrix} \cos(2\phi) & \sin(2\phi) \\ \sin(2\phi) & -\cos(2\phi) \end{bmatrix}. \tag{5.10}$$

Before looking at the eigenvalues of $Q_{\alpha\beta}$ which define the nematic order parameter, it shall be noted that the eigenvalues of $Q^i_{\alpha\beta}$ are $\lambda = \pm 1$. Using the derived expression for matrix $Q^i_{\alpha\beta}$ the eigenvalues Λ of the full matrix $Q_{\alpha\beta}$ can be determined from the characteristic polynomial, giving:

$$\Lambda = \pm\frac{1}{N}\sqrt{\left[\left(\sum_i \cos(2\phi)\right)^2 + \left(\sum_i \sin(2\phi)\right)^2\right]}, \tag{5.11}$$

with sums extending overall N particles. This can also be expressed as:

$$\Lambda = \pm\frac{1}{N}\sqrt{\sum_i \sum_j \cos\big(2(\phi_i - \phi_j)\big)}. \tag{5.12}$$

Implementation. The nematic order parameter can be implemented either by calculating the full matrix $Q_{\alpha\beta}$ from the system's configuration and then computing eigenvalues using numerical methods. Alternatively, it can be determined by calculating the largest, positive eigenvalue for $Q_{\alpha\beta}$ according to eq. (5.12) directly:

```
1     Lambda = N
2     do i=1,N
3         do j=i+1,N
4             Lambda = Lambda + 2*cos(2*(phi_i-phi_j))
5         end do
6     end do
7     Lambda = sqrt(Lambda)/N
```

In the first line, `Lambda` is initialized with `N`, accounting for the excluded terms $i = j$ in line 2-7.

5.3. Loop Order Parameter

Based on the loops and their sizes, gathered by the loop-detection algorithm (see chapter 2 - section 2.4), a loop order parameter Γ was formulated.
The target order parameter Γ should achieve unity for a perfectly ordered system. Perfect order is here claimed to be a loopy network of a uniform loop size. Disorder within the configuration in terms of loops of smaller or larger size than the targeted loop size should be detected by the order parameter giving a value $\Gamma < 1$.

The described behaviour of the order parameter Γ was achieved by calculating a weight factor $w_{i,l'}$ for each particle i. The index l' takes values $l' = 1, 2$, as each particle is part of at most two loops, named $L_{i,l'}$. The particle's weight factor depends on the loop sizes $|L_{i,l'}|$. The weight is determined by a target distribution, e.g. a Gaussian distribution $G(x|\mu, \sigma)$ with mean $\mu = 6$ and standard deviation $\sigma = 0.5$. The weight factor is used to detect loops of target size (here $|L| = \mu = 6$) and assigning a maximum weight, $w = 1$ to them. Loops with size smaller or larger than the target size are penalized by assigning a smaller weight:

$$w(i, l') = \exp \frac{[-(|L_{i,l'}| - \mu)^2}{\sigma^2}] \propto G\Big(x = |L_{i,l'}|\Big|\mu, \sigma\Big). \tag{5.13}$$

The local loop order parameter $\gamma(\vec{r}_i)$ is formulated for a particle with index i and at a position $\vec{r}_i$. As stated above, for the case where each particle is adjacent to at most two loops - written as $L_{i,l'}$ with index $l' = 1, 2$ - the local loop order parameter $\gamma(\vec{r}_i)$ is then stated as an average value for the two weights taken from the loops adjacent to particle i:

$$\gamma(\vec{r}_i) = \frac{1}{2} \sum_{l=1}^{2} w(i, l'). \tag{5.14}$$

Finally, the global loop order parameter Γ, is defined by an average over the local order parameters:

$$\Gamma = \frac{1}{N} \sum_{i=1}^{N} \gamma(i). \tag{5.15}$$

For means of implementation, the global order parameter can also be written as a sum over all loops L_l, with $l = 1, ..., N_L$ and N_L indicating the total number of loops:

$$\Gamma = \frac{1}{2N} \sum_{l=1}^{N_L} |L_l| \cdot w(l). \tag{5.16}$$

The underlying distribution to assigning the weight factors to the loops can also be chosen differently. While a δ-distribution would solely count loops of exactly the target size, the Gaussian distribution with a very narrow width (small standard deviation) assigns a weight also to loops of similar size. In some systems, it might be interesting to also render the tail of the loop size distribution within the order parameter. For such systems, the asymmetrical log-normal distribution can be helpful [31]. When utilizing the log-normal distribution or any other asymmetrical distribution, it should be noted that the mode instead of the mean of the distribution should be chosen as the target loop size.

5.4. Degree Order Parameter

Another order parameter that measures order on the basis of the number of bonds per particle is defined in the following. For determining the number of bonds per particle, the particle-based graph is used, and the *networkx* [8] function for the node degree is applied to each node. The degree of a node is defined as the number of edges incident to that node [15].
The order parameter D_d for particles with d bonds is then defined as:

$$D_d = \frac{N_d}{N}, \tag{5.17}$$

where N_d is the number of particles of degree d and N is the total number of particles.
Using this type of order parameter, structural order is directly related to the number of particles with a specific number of bonds. This type of order parameter is useful only if a unique mapping between the number of bonds for each particle and a specific particle arrangement is possible.

5.5. Largest Cluster Size

When observing self-assembly processes, the largest cluster size is a prevalent measure of the stages in the process. [32]. The largest cluster size can be used as an absolute size or as a relative size comparing the size of the largest cluster to the total number of particles in the system:

$$C^{\max} = \max_{i \in N_{\text{clust}}} |C_i| \tag{5.18}$$

$$c^{\max} = \frac{C^{\max}}{N} \tag{5.19}$$

For determining the collection of clusters C_i, a *networkx* method [8] is used on the particle-based graph (see chapter 2 - section 2.4.1).

5.6. Loop & Cluster Size Distributions

Apart from scalar quantities such as order parameters, also distributions of quantities such as loop size, loop numbers, cluster size and cluster numbers can be determined and analyzed.

Part II.

Lattices of Hard Ellipses - A Study of Entropies

1. Introduction

Hard-core particles were present from the very beginning when computer simulations entered as a method of understanding liquid properties. The first computer simulations of a liquid were performed for hard disks in 1953 by Metropolis, Rosenbluth&Rosenbluth and Teller&Teller [4]. A decade later, in 1957, another milestone in the history of computer simulations occurred with the observation of hard-core spheres crystallizing at high densities [33]. This observation was ground-breaking as the ordering of particles into lattices was believed to be driven by the particle's attractive forces. As hard-core spheres have purely repulsive interactions, this explanation did not hold. Alder and Wainwright, who first observed this behaviour, used Monte-Carlo simulations. Using Molecular Dynamics showed the same behaviour and was achieved by Hoover, and Ree [34] almost a decade later in 1968.
Although a theory on the ordering of hard-core particles existed as early as 1949 with Onsager's theory of nematic fluids [35], the observations of the crystallization of hard spheres were particularly important for a wider acceptance, even though not immediate, of understanding entropy as a thermodynamic potential capable of driving systems in ordered phases [36].

The principle of entropic ordering, first appearing in Onsager's theory of nematic fluids [35] but understood and accepted only later with [34], is fundamental and comes into play for all types of interactions, not just hard-core interactions. In thermal systems, i.e. systems with attractive and/or repulsive interactions other than hard-core interactions, the entropy is in interplay with other thermodynamic potentials, e.g. the internal energy, to minimize the underlying thermodynamic potential. This could be the Helmholtz Free Energy for systems described by the canonical ensemble or the Gibbs Free Energy for systems described by the isobaric-isothermal ensemble.

With the crystallization of hard spheres, another important chapter in the history of computer simulations of molecular systems was opened - the chapter on studying crystalline phases. For liquid phases, the advance in computational methods was most interesting as fundamental theories of liquids were very limited for practical purposes. With the use of thermodynamic integration from the ideal gas as a reference state, most fundamental properties such as the free energy could be computed. This method was extended to crystalline phases in 1984 by Frenkel and Ladd. They proposed a method relying on the Einstein Crystal as a

reference system to compute the free energy of a crystal via a suitable thermodynamic integration path [20]. This method is widely used and was reformulated in a slightly different form by Vega and Noya [18] in 2007. Finally, this method helped to solve a decades-long dispute on the true lattice ground state of hard spheres. Noya and Almarza computed free energies showing that the fcc lattice state is entropically favoured over the hcp lattice state by a minute free energy difference [23].

The particle type studied in this project is the hard-core ellipse. Ellipses with a hard-core interaction first became a subject to computer simulations in 1970, when Vieillard-Baron proposed an overlap criterion for particles of elliptic shape. First simulation studies were conducted by Vieillard-Baron [37] on the isotropic-nematic transition, as well as the solid-nematic melting transition and made predictions on the transition densities. In later publications on the hard ellipse model, the focus remained on these two transitions. Important contributions following Vieillard-Baron were made by Cuesta, and Frenkel [38] as well as Ward and Lado [39]. In the next century, Davatolhagh and Foroozan [40] described the diffusive behaviour in isotropic and nematic phases of hard ellipses and Luo, Sagis and Ilg [41] proposed expressions for the free energy in dilute and nematic phases and compared these to free energies obtained from simulations, using the thermodynamic integration method. An extensive study on hard ellipses was performed in 2014 by Bautista-Carbajal, and Ordiozola [42] in which the authors computed phase diagrams of hard ellipses with aspect ratios $1 < \kappa < 5$ employing the replica-exchange Monte Carlo method [43].

Despite the importance of the hard ellipses as a model system for nematic phases, the following pages will focus on the crystalline phase of ellipses. Already with the very first publication on hard-ellipses studied by Monte-Carlo simulation methods by Vieillard-Baron [37], the infinitely degenerate ground-state - at least for closest packing - of the ellipse solid was observed. Further, a construction of these infinitely many lattice states of ellipses was described. An in-depth study of these infinitely many lattice configurations has not been published to the best of my knowledge. A clear understanding of the crystalline ellipse phase is missing. Frenkel and Cuesta [38] as well as Bautista-Carbajal and Ordiozola [42] use different crystalline lattice states for their studies on the solid-nematic transition. While the first used the later defined *parallel* lattice state in their studies of the solid-nematic melting transition, the also later defined *diagonal* lattice was used by the latter.

The research project presented here addresses the crystalline phases of hard-core elliptic particles in two dimensions and tries to provide a missing link in the understanding of the ellipse crystalline phase. In chapter 2 the overlap criterion by Vieillard-Baron [44] is presented. The infinite number of possible lattice

states already described by Vieillard-Baron [37] is further increased by introducing a second lattice parameter acting on the positional order of the ellipses (see chapter 3 - section 3.1). Within the huge parameter space representing all ellipse lattices, two candidate lattice structures, namely the parallel and diagonal lattice configuration, are isolated (see chapter 3 - section 3.2). Helmholtz free energies of these candidate structures are computed within the canonical ensemble (see chapter 4) using the Einstein Molecule method proposed by [18] as a variant of the Frenkel Ladd method [20]. Finally, a prediction of the free energy difference between the parallel and diagonal lattice configuration in the thermodynamic limit is attempted using finite-size corrections (see chapter 5).

2. Hard Ellipses Model

As early as 1970 J. Vieillard-Baron put forward a criterion to check for overlap between two ellipses of same aspect ratio in a closed-form expression [44]. Using this criterion allowed him to conduct computer simulations of hard elliptic particles and made him a pioneer in studying assembly scenarios of anisotropic particles. To this day, the Vieillard-Baron criterion is used widely due to its low computational cost and its simple, analytical form. Another criterion that can be used for determining overlap of two ellipses is the recently published formula for the distance of closest approach [45] between two ellipses. This method even allows to identify overlap of ellipses with distinct aspect ratios.

2.1. Units

The chosen unit of length within the simulations is:

$$2\sqrt{ab} = 1. \tag{2.1}$$

Here, the parameters a and b refer to the semi-major and semi-minor axis of the ellipse respectively. The unit of length is the natural extension of the typical unit of length $\sigma = 1$ for isotropic hard core particles (i.e. disks), with σ being the diameter of the disks. The unit in eq. (2.1) represents the geometric mean of the ellipse's two axes, the major axis $2a$ and minor axis $2b$. Therefore for disks with $a = 0.5$ and $b = 0.5$ the length unit $\sigma = 1$ is restored.
The ellipse aspect ratio is given as:

$$\kappa = \frac{a}{b}. \tag{2.2}$$

With the length unit at hand (see eq. (2.1)), the following relations for the semi-major and semi-minor axes a and b are derived:

$$a = \frac{\sqrt{\kappa}}{2} \tag{2.3}$$

$$b = \frac{1}{2\sqrt{\kappa}}. \tag{2.4}$$

2.2. Overlap Criterion by Vieillard Baron

In the MC Program written in the course of this thesis, the criterion by Vieillard-Baron [37][44] is applied to check the overlap between two ellipses. The original overlap criterion is based on the following analytical expressions:

$$\Psi = 4(f_1^2 - 3f_2)(f_2^2 - 3f_1) - (9 - f_1 f_2)^2. \tag{2.5}$$

Specifying the two ellipses with $\alpha = 1$ or 2, the functions f_1 and f_2 read:

$$f_\alpha = 1 + G - \left[\frac{(\vec{\mathbf{r}} \cdot \tilde{\mathbf{u}}_\alpha)^2}{a^2}\right] - \left[\frac{(\vec{\mathbf{r}} \cdot \vec{\mathbf{u}}_\alpha{}')^2}{b^2}\right], \tag{2.6}$$

with

$$G = 2 + \left[\frac{a}{b} - \frac{b}{a}\right]^2 \sin^2\theta. \tag{2.7}$$

Here, $\mathbf{u}_\alpha$ and $\mathbf{u}'_\alpha$ are the unit vectors pointing in direction of the major axis and minor axis of ellipse α respectively. Further, θ indicates the angle enclosed by the two ellipses' directors $\vec{\mathbf{u}}_1$ and $\vec{\mathbf{u}}_2$.

Using eq. (2.5) to (2.7), Vieillard-Baron's criterion states that for overlap of two ellipses $\alpha = 1$ and 2, the relation $\Psi > 0$ must hold and that either $f_1 < 0$ or $f_2 < 0$.

Implementation. The Monte-Carlo Program stores particle orientations in degrees with respect to the the x-axis. Hence the reformulation using the orientation of the ellipse in terms of ϕ reads:

$$f_\alpha = 1 + G - \tag{2.8}$$

$$- \left[\frac{\mathbf{r} \cdot \hat{e}_x \cos(\phi_\alpha) + \mathbf{r} \cdot \hat{e}_y \sin(\phi_\alpha)}{a}\right]^2 - \left[\frac{\mathbf{r} \cdot \hat{e}_x \cos(\phi_\alpha + \pi) + \mathbf{r} \cdot \hat{e}_y \sin(\phi_\alpha + \pi)}{b}\right]^2 \tag{2.9}$$

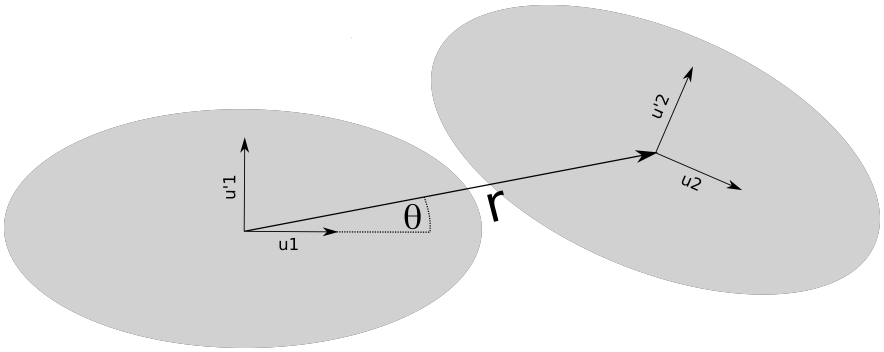


Figure 2.1.: Parameters that enter the Vieillard-Baron criterion for overlap.

3. Ellipse Lattices

In search of the true lattice ground state of hard ellipses, a parameterization of the possible candidate structures is put forward, with the mathematical details covered in part I - section 2.1.2. In this chapter the described lattice parameterization is once more motivated and specified. Following the lattice parameterization, the optimization procedures for finding entropically favoured lattice-types is presented.

3.1. Lattice Parameterization

A parameterization for two-dimensional lattices of ellipses is specified. These types of lattices can be constructed by suitable transformations of the hexagonal lattice of disks. Details on this construction as well as its implementation for purposes of simulations can be found in part I - section 2.1.2.

An ellipse can be created starting from a disk by using an affine transformation, i.e. an area-preserving linear transformation, that stretches the disk along one axis and compresses it along the orthogonal axis. When applying this transformation to a lattice of disks, the direction of this transformation can be chosen using any angle $\omega \in (0, \frac{\pi}{6})$. For each angle ω, a different ellipse lattice type is produced, resulting in an infinite number of possible ellipse lattice configurations, parameterized by ω. This construction of these types of ellipse lattices was first described by Vieillard-Baron [37]. The interval for ω reflects the six-fold rotational symmetry - or $\mathcal{C}_6$-symmetry - of the hexagonal lattice. Here, the resulting lattice state is named ω which also serves us as the first lattice parameter characterizing the ellipse lattice.
In the next step, the six nearest-neighbour distances within the ω-lattice are inspected more closely. In a hexagonal lattice of disks, these six distances are equal. Due to the symmetry of the ω-lattice, for this types of lattices, three distinct nearest-neighbour distances occur in the general case. However, when $\omega = 0$ and $\omega = \frac{\pi}{6}$, the number of distinct nearest-neighbor distances reduces to two. In both cases, the lost symmetry in contrast to the hexagonal lattice allows for another affine transformation that acts on the positions of the particles solely and can change these interconnected distances. For the latter case, this transformation is formulated in part I - chapter 2 - eq. (2.14). The positional alignment of the particles is taken into account via the second lattice parameter τ.

The parameterization (ω, τ) covers all possible two-dimensional lattices of ellipses. In the following, a simulation-based lattice optimization (see part I - section 2.2) is used to probe the parameter space spanned by ω and τ and find the entropically most favourable lattice types.

3.2. Lattice Optimizations

By applying the lattice MC method (see part I - section 2.2) at constant volume the entropically most favourable lattice configurations within the (ω, τ)-lattice space are searched. The lattice MC methods are proposed as independent variations of ω and τ. Therefore the optimization is approached in two steps: first, the ω-optimization is carried out, followed by a τ-optimization. The optimal lattice states are reflected in the resulting probability distribution function (pdf) over the lattice parameters ω and τ.

The above optimization procedure was carried out in detail for three different reduced particle densities, namely $\rho^* = 0.97, 0.98$ and 0.99 as well as for three different ellipse aspect ratios, namely $\kappa = 2, 4$ and 6. Throughout, a system of $N = 128$ particles was studied, arranged in a lattice with $N_x = 8$, $N_y = 16$.

Context. The optimization problem is approached by varying the two lattice parameters ω and τ separately while keeping the other lattice parameter constant. Although this setup is not complete, it is deemed justified in this particular case: The first optimization step of the parameter ω clearly indicates two lattice candidates in the resulting probability distribution of lattice states with all of the other candidate structures strongly suppressed (see fig. 3.2). With this clear indication of two very distinct optimal lattice parameters, $\omega_{\text{opt}} = 0$ and $\omega_{\text{opt}} = \frac{\pi}{6}$, which will be named the diagonal and parallel configuration respectively, the second optimization step with respect to the parameter τ is only applied to these two optimal ω-parameters.

ω - Optimization

In a first step all states along the ω-axis, with $\omega \in (0, \frac{\pi}{6})$ are probed via ω MC-moves while keeping a constant lattice parameter $\tau = 1$. A number of 60 independent simulations were run, each over 5.0E+05 cycles, with a probability of proposing an ω-move of `probOm=0.05` and probabilities for a translation or rotation move of `probT=probR=0.475`. Each of the 60 simulations started from a different initial lattice configuration, where these initial ω-states covered the total ω-axis $(0, \frac{\pi}{6})$ uniformly. For the first 2.5E+04 cycles, the lattice was equilibrated at its initial ω-state by proposing particle translations and rotations only. The total equilibration phase extended over 5.0E+04 cycles, where in the second half

of the equilibration phase, also ω-moves were allowed.
In fig. 3.1, trajectories of the lattice parameter ω over the course of the simulation time for all simulations are shown. From inspecting the simulation processes, the truncation point for trajectories was set to 50% or at 2.5E+05 cycles, indicated as green vertical lines in the figure. Only samples beyond this point were considered in the evaluation of the probability distribution function, or pdf. The corresponding pdfs are depicted in fig. 3.2. It is clearly the case that lattice states with ω values other than $\omega = 0$ and $\omega = \frac{\pi}{6}$ are strongly suppressed. The system strives to either a lattice state with $\omega = 0$, which we call the diagonal state or to a state with $\omega = \frac{\pi}{6}$, which we call the parallel state. Each of these two states is favoured among all densities and ellipse aspect ratios invested here. It is further observed that the shape of the peak in the pdf is broader and lower for all diagonal states and narrower and higher for all parallel states. These observed tendencies of the peaks deforming with increasing aspect ratio are that they get higher and narrower. The same tendency is observed for increasing densities.

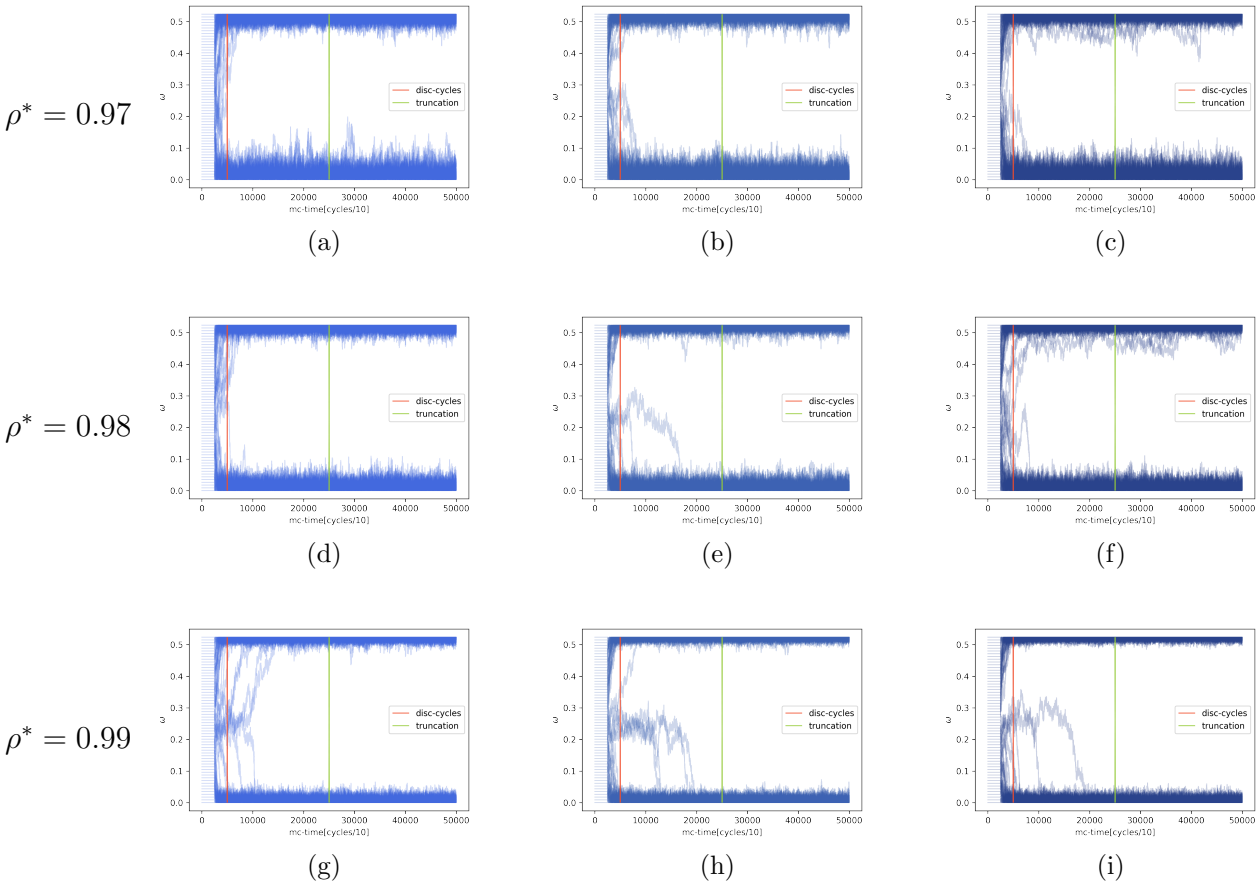


Figure 3.1.: All simulation trajectories showing the varying ω over MC-time for the ω-optimization. The simulations were done at denisities $\rho^* = 0.97$, 0.98 and 0.99 (rows) and ellipse aspect ratios $\kappa = 2$, 4 and 6 (columns). For each system, 60 independent simulations where carried out with distinct initializations for ω. The red bar indicates the end of the equilibration phase and the green bar the start of the sampling.

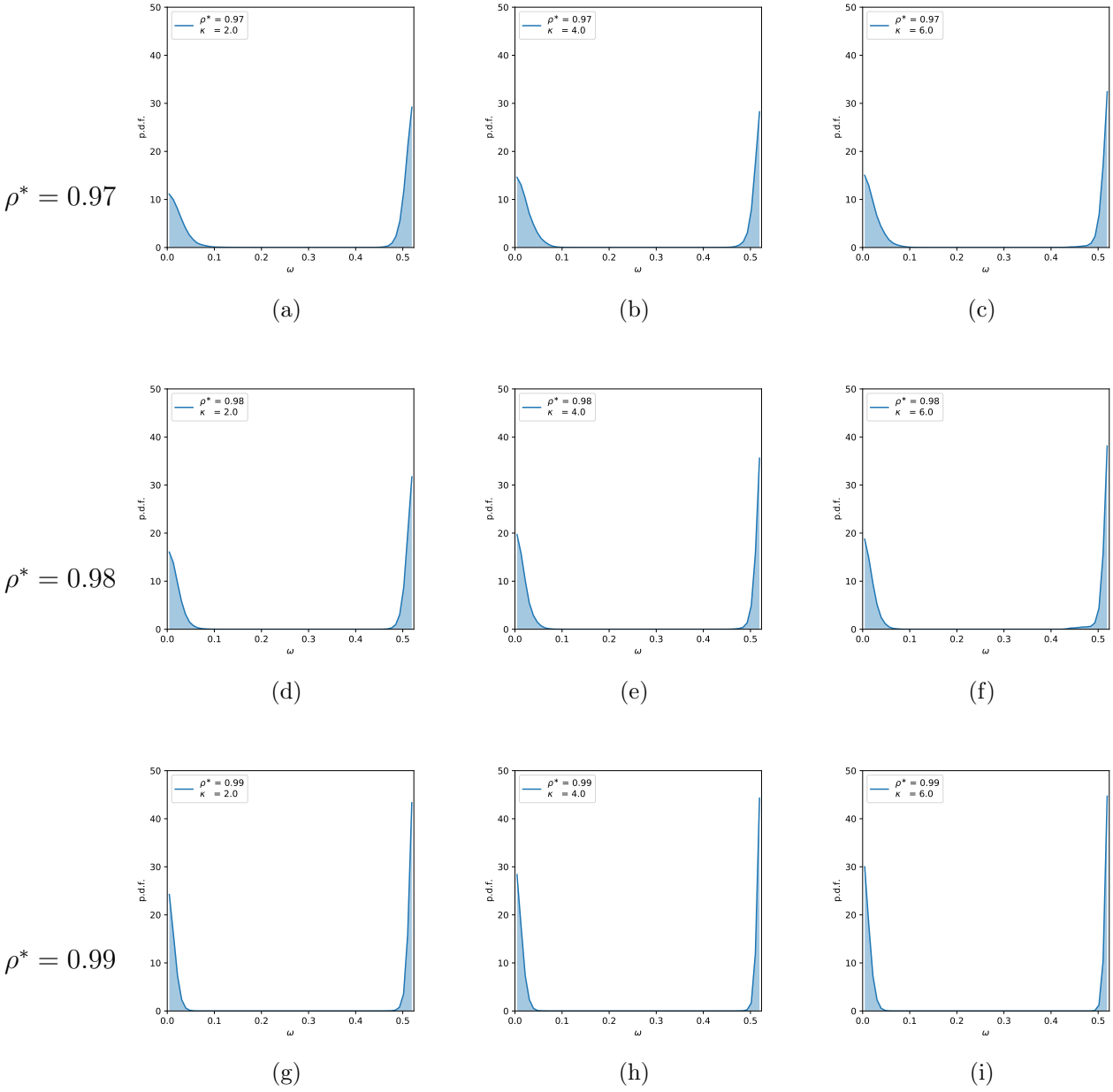


Figure 3.2.: Probability distribution functions (pdf) from MC-simulations with varying lattice parameter ω for denisities $\rho^* = 0.97$, 0.98 and 0.99 (rows) and ellipse aspect ratios $\kappa = 2$, 4 and 6 (columns).

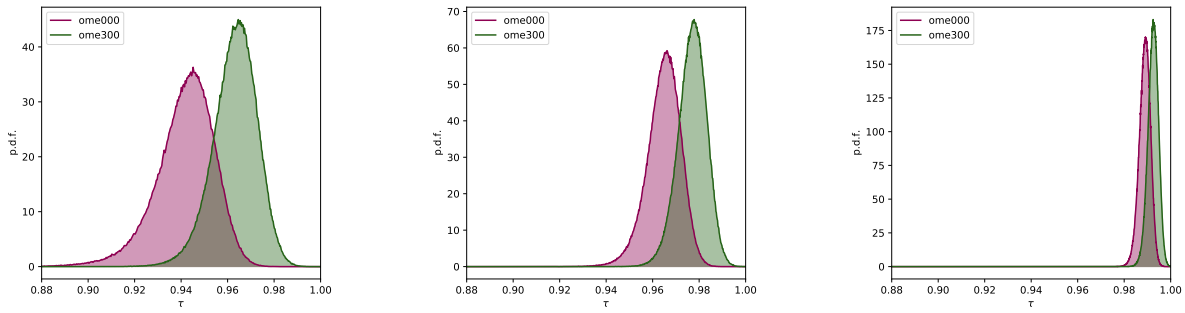
τ - Optimization

In a second step, the lattice parameter τ is optimized for each of the two optimal ω-values. The lattice MC method (see section 2.2) with varying lattice parameter τ and fixed lattice parameters $\omega = 0$ and $\omega = \frac{\pi}{6}$ was used. A total number of ten simulations were run in parallel for each lattice type and each density, all with an initial value of $\tau = 1$. Similar to the ω-optimization, the first 2.0E+05 cycles were run without lattice moves. This is so that the lattice relaxes within the proposed lattice state. Only after these cycles, lattice moves varying parameter τ are proposed with a probability of 0.05, compared to probabilities 0.475 for both translation and rotation moves. After a total of 2.0E+06 discarded cycles, the sampling began with a step-size for sampling lattice parameters τ of 2.0E+02 cycles. With a total length of 2.0E+07 cycles, each simulation results in a total size of 9.0E+04 sampled lattice parameters.

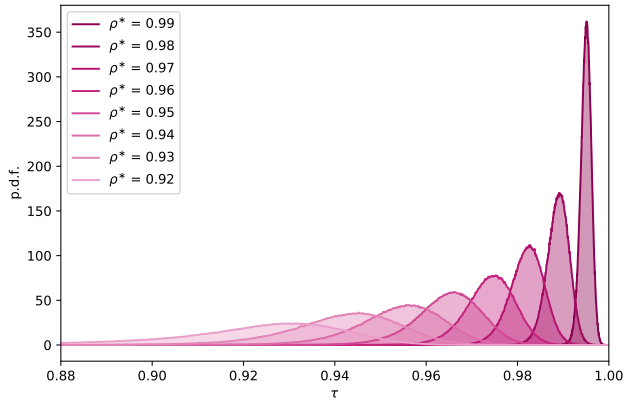
Figure 3.3 shows the pdf over the lattice parameter τ for different densities ρ^* and lattice states ω. It can be observed that for all densities and lattice states, the pdf yields a non-symmetric distribution with a tail towards smaller values of τ. The value for τ giving the maximum probability is identified as the optimal lattice parameter. All achieved parameters from this optimization can be seen in table 3.1.

	ω_{opt}	$\tau_{\mathrm{opt}}(N=128)$	$\tau_{\mathrm{opt}}(N=100)$
$\rho^*=0.92$	0°	0.9295(34)	0.9306(04)
	30°	0.9559(06)	0.9552(63)
$\rho^*=0.93$	0°	0.9448(05)	0.9445(60)
	30°	0.9644(41)	0.9663(62)
$\rho^*=0.94$	0°	0.9574(38)	0.9565(60)
	30°	0.9726(10)	0.9726(77)
$\rho^*=0.95$	0°	0.9660(37)	0.9651(15)
	30°	0.9770(55)	0.9784(39)
$\rho^*=0.96$	0°	0.9752(18)	0.9749(58)
	30°	0.9830(69)	0.9840(41)
$\rho^*=0.97$	0°	0.9826(36)	0.9819(14)
	30°	0.9881(74)	0.9887(45)
$\rho^*=0.98$	0°	0.9891(68)	0.9892(75)
	30°	0.9924(90)	0.9932(73)
$\rho^*=0.99$	0°	0.9951(80)	0.9949(08)
	30°	0.9966(36)	0.9968(93)
$\rho^*=0.995$	0°	0.9976(78)	0.9975(83)
	30°	0.9983(43)	0.9984(39)

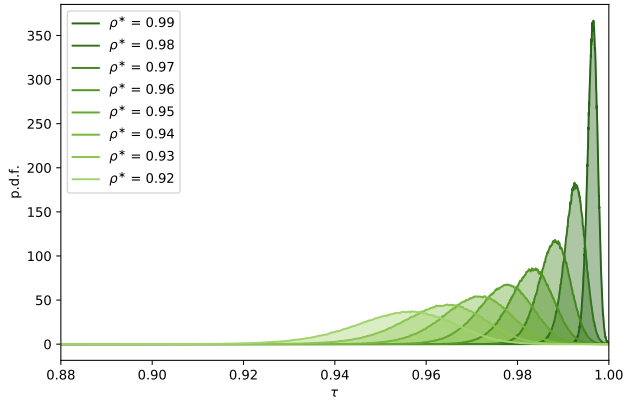
Table 3.1.: Results for the optimal lattice parameters ω and τ for ellipse lattices of $N=100$ and $N=128$ particles at densities $\rho^*=0.93$ up to 0.995 and ellipses with aspect ratio $\kappa=2$ as obtained from the optimization procedures using the lattice MC method.



(a) (b) (c)



(d)



(e)

Figure 3.3.: Results showing the pdfs from MC-simulations with varying lattice parameter τ with ellipse aspect ratio $\kappa = 2$. Top: denisities $\rho^* = 0.93$, 0.95 and 0.98 for parallel (green) and diagonal (pink) lattice types. Center: diagonal lattice type for eight different densities, as labeled. Bottom: results for the parallel lattice type at the same densities.

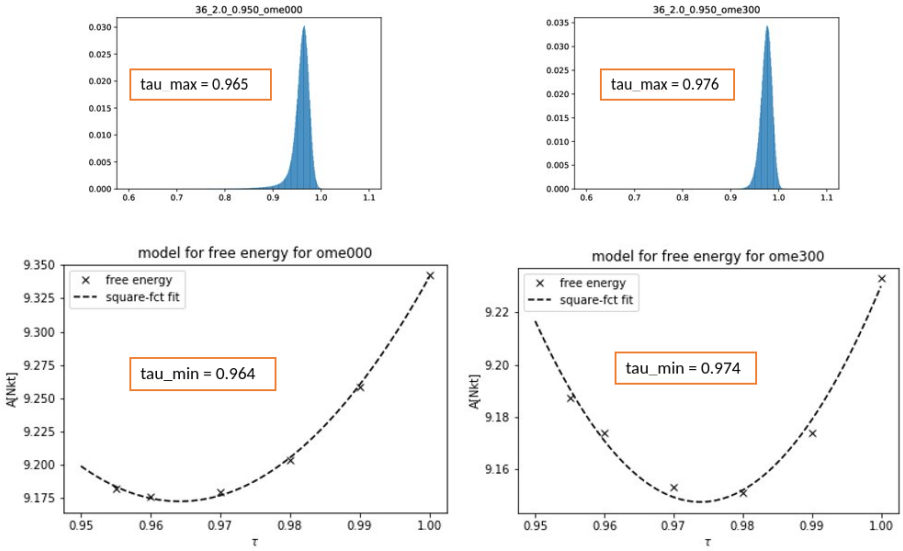


Figure 3.4.: Consistency check between lattice-optimization using τ-moves (top) and direct free energy computations (bottom) for a lattice of $N = 36$ ellipses with aspect ratio $\kappa = 2$ in the diagonal (left) and parallel (right) lattice sate.

3.3. Consistency Check

In an attempt to check the consistency of the τ-optimization approach with the underlying thermodynamic potential of the free energy, a number of free energy computations were carried out for the two lattice states - parallel and diagonal - at a density $\rho^* = 0.95$ for a system of $N = 36$ particles with aspect ratio $\kappa = 2$. Free energies were computed for this rather small system size at different lattice parameters τ, using the Einstein Molecule method (part I - section 4.4). The lattice state τ with minimal free energy was evaluated by minimizing the quadratic function fitted to the data points, as can be seen in fig. 3.4. The lattice parameter τ_{min} that minimizes the free energy functional was then compared to the lattice parameter τ_{max} that maximizes the probability distribution function obtained from the τ-lattice MC procedure. In both lattice systems, i.e. the parallel and the diagonal state, we found that the varying lattice parameter approach produces the same results as the free energy approach with an accuracy of $\approx 99\%$.

4. Free Energy Computation

The free energy computation for the two ground-state candidates - the diagonal and the parallel lattice - is based on the Einstein Molecule method, described in detail in part I - section 4.4. This chapter reports simulation details and results on the total free energies $A_{\mathrm{tot}}[Nk_{\mathrm{B}}T]$ (see part I - eq. (4.35)) of these two lattice types at two densities $\rho^* = 0.95$ and 0.98, for ellipses with $\kappa = 2$ and for six system sizes, $N = 32$, 72, 128, 200, 288 and 392. The prediction of the free energy and the corresponding error are estimated by performing ten independent computations for each state-point. With all these efforts taken, the results will show that the parallel lattice state is the entropically favoured lattice state, i.e. the lattice state with the lowest free energy. This is true in lattices of both densities and for all system sizes considered in this study.

4.1. Prerequisites

4.1.1. Simulation Box Shapes

Numerical results of the free energy using a simulation-based method always depend on the simulation box size and shape. These features are responsible for errors, which amongst others, occur due to the aforementioned finite size of the simulation box and its shape. The error associated with the size of the simulation box can be taken into account by applying finite-size corrections. The effect of the simulation box shape can be considered by comparing free energy computations performed for different box shapes. Within this project, the finite-size corrections were computed using different box sizes. An analysis of different box shapes is not included in this thesis but is considered in a forthcoming publication [46].

The finite-size correction method, as discussed in part I - section 4.5 is applied by computing the free energies at a number of different system sizes, in terms of the particle number N, and fitting the relation proposed in part I - section 4.5 - eq. (4.48). As can be seen from the correction model the simulation box aspect ratios for each lattice, the parallel and diagonal individually, must be the same across all system sizes N for a correct fitting procedure. As the ultimate aim is to compute free energy differences between the diagonal (d) and parallel (p) lattice type, $\Delta A_{\mathrm{p-d}} = A_{\mathrm{p}} - A_{\mathrm{d}}$, the box aspect ratios $\kappa_{\mathrm{box,p}}$ and $\kappa_{\mathrm{box,d}}$, within

eq. (4.48) (in part I) of the parallel and diagonal lattices should ideally be the same.
For the following computations of the free energies of lattices of ellipses with aspect ratio $\kappa = 2$, we have selected a box shape for each lattice that is sufficiently similar to a quadratic shape and at the same time are similar to each other. The simulation box shape most similar to a quadratic shape for the diagonal lattice of ellipses with $\kappa = 2$ is a box hosting a $(N_x = 2, N_y = 4)$-lattice arrangement of particles. The same lattice arrangement is picked for the parallel lattice. From these box shapes, cascades of larger simulation boxes can be constructed (see fig. 4.1):

N=32 $box^{i=1} = 2 \times box^{i=0} = (4, 8)$

N=72 $box^{i=2} = 3 \times box^{i=0} = (6, 12)$

N=128 $box^{i=3} = 4 \times box^{i=0} = (8, 16)$

N=200 $box^{i=4} = 5 \times box^{i=0} = (10, 20)$

N=288 $box^{i=5} = 6 \times box^{i=0} = (12, 24)$

N=392 $box^{i=6} = 7 \times box^{i=0} = (14, 28)$

Due to stacking order and periodic boundary conditions the condition that N_x for the diagonal and N_y for the parallel lattice are even numbers must be fulfilled. This however rules out half-step enlargements increases of the box size in the parallel lattice. For the diagonal lattice the following box shapes are allowed:

N=18 $box^{i=1} = 1.5 \times box^{i=0} = (3, 6)$

N=50 $box^{i=1} = 2.5 \times box^{i=0} = (5, 10)$

N=98 $box^{i=2} = 3.5 \times box^{i=0} = (7, 14)$

N=162 $box^{i=3} = 4.5 \times box^{i=0} = (9, 18)$

N=242 $box^{i=4} = 5.5 \times box^{i=0} = (11, 22)$

N=338 $box^{i=5} = 6.5 \times box^{i=0} = (13, 26)$

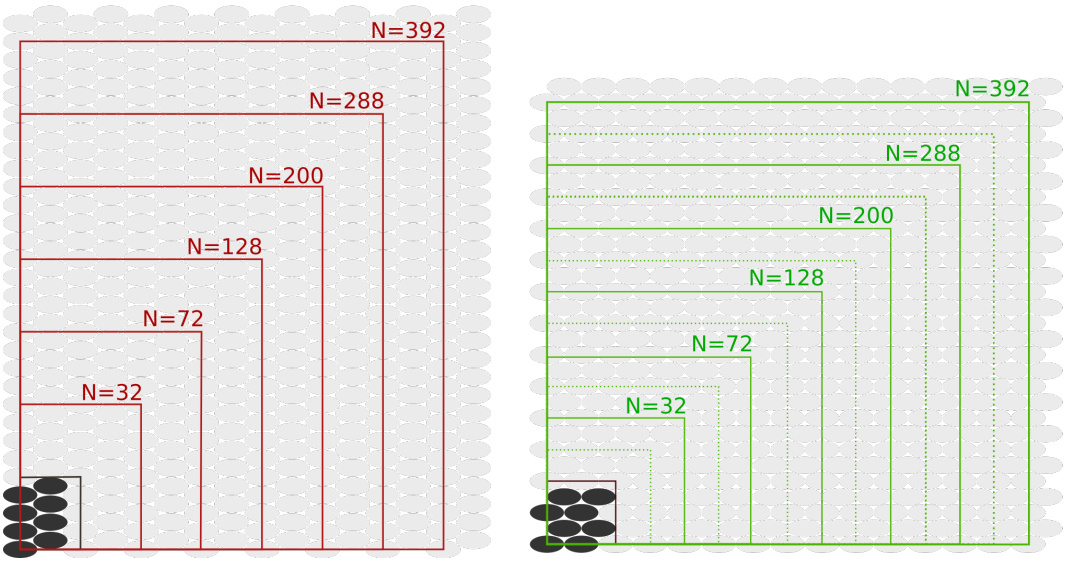


Figure 4.1.: Cascades of simulation boxes in the parallel (red) and diagonal lattice (green) starting from the initial boxes $box^{i=1} = (N_x, N_y) = (2, 4)$.

4.1.2. Autocorrelation Analysis

Prior to computing the contribution ΔA_2 (part I - eq. (4.36c)) to the total free energy by simulation of all state-points along the thermodynamic integration axis from the mixed, $\mathcal{H}(\lambda) = \lambda\mathcal{H}_{\text{ein}} + \mathcal{H}_{\text{sol}}$, to the pure state, $\mathcal{H}(\lambda = 0)$, the auto-correlation times $\tau^{\text{rot}}_{\text{a.c.}}$ and $\tau^{\text{trans}}_{\text{a.c.}}$, estimated from the autocorrelation function $f_{\text{rot}}(t) = \langle U_{\text{ein,rot}}(t_o) U_{\text{ein,rot}}(t_o + t) \rangle$ and $f_{\text{trans}}(t)$ accordingly, were studied in preceding simulations for each of these states. The autocorrelation time is observed to be the longest for states with $\lambda \approx 0$. Therefore the maximum auto-correlation time corresponds to the mixed state with the lowest contribution of the Einstein crystal and highest contribution of the pure hard core particle interactions, hence almost the free solid.

Simulation Details. Simulations for the evaluation of the auto-correlation time were done for two densities $\rho^* = 0.95, 0.98$ and all investigated system sizes, namely $N = 32$, 72, 128, 200, 288 and 392. For each set of density and system size a total of five independent simulations were performed with Einstein spring constants $\Lambda' = 46$, 47, 48, 49 and 50 (see part I - chapter 4 - eq. (4.38)). The mean auto-correlation time is then retrieved from taking the mean of the auto-correlation functions. The simulation parameters for system sizes $N = 32$ and $N = 72$ are set to a total simulation length of 1.0E+07 cycles, a parameter step size of 1000 cycles and a total of 1.0E+06 discarded cycles. For system sizes $N = 128$, $N = 200$ and $N = 288$ the corresponding parameters are 2.0E+07 cycles for the simulation length, a step-size of sampling parameters of 2.0E+03

and a total of 2.0E+06 discarded cycles. For the largest system size of $N = 392$ these parameters are 3.0E+07 cycles, a parameter sampling step-size of 3.0E+03 and a total of 3.0E+06 discarded cycles.

Results. From the simulation trajectories the autocorrelation functions (part I - section 3.2 - eq. (3.1)) of the total Einstein energies relating to translations, $E_{\text{trans}} = \sum_{i=1}^{N} \Lambda \Delta r^2$ and rotations are determined. The best estimate of the autocorrelation time $\tau_{a.c.}$ is taken as the time at which the mean of the autocorrelation functions reaches a value of zero. A selection of autocorrelation functions for each of the five independent simulations, as well as the average autocorrelation function are depicted in fig. 4.2 and 4.3. These figures show the mixed state, $\mathcal{H}(\lambda)$ with λ closest to zero, which corresponds to $\lambda_{\text{ID}} = 1$. The system sizes are $N = 72$ and $N = 288$. The autocorrelation times are indicated by vertical red lines.

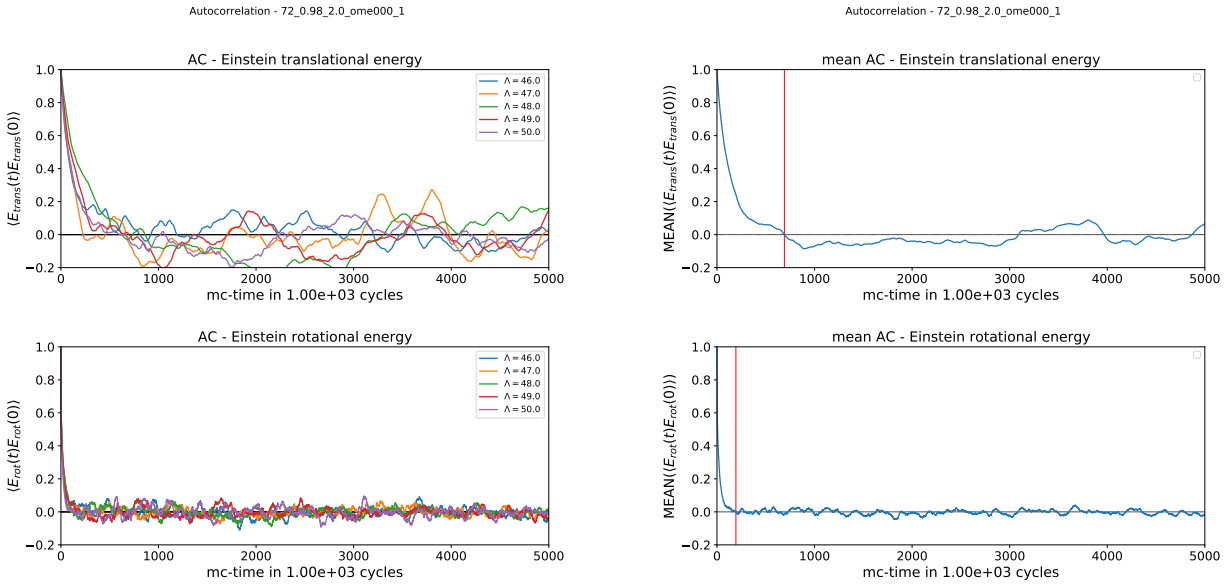


Figure 4.2.: Translational (top) and rotational (bottom) autocorrelation functions of five independent runs (left) and the respective mean (right) for $\lambda_{\mathrm{ID}} = 1$, $\rho^* = 0.98$ at a system size of $N = 72$ in the diagonal lattice state $\omega = 0$. Estimates of the autocorrelation times are indicated by red vertical lines.

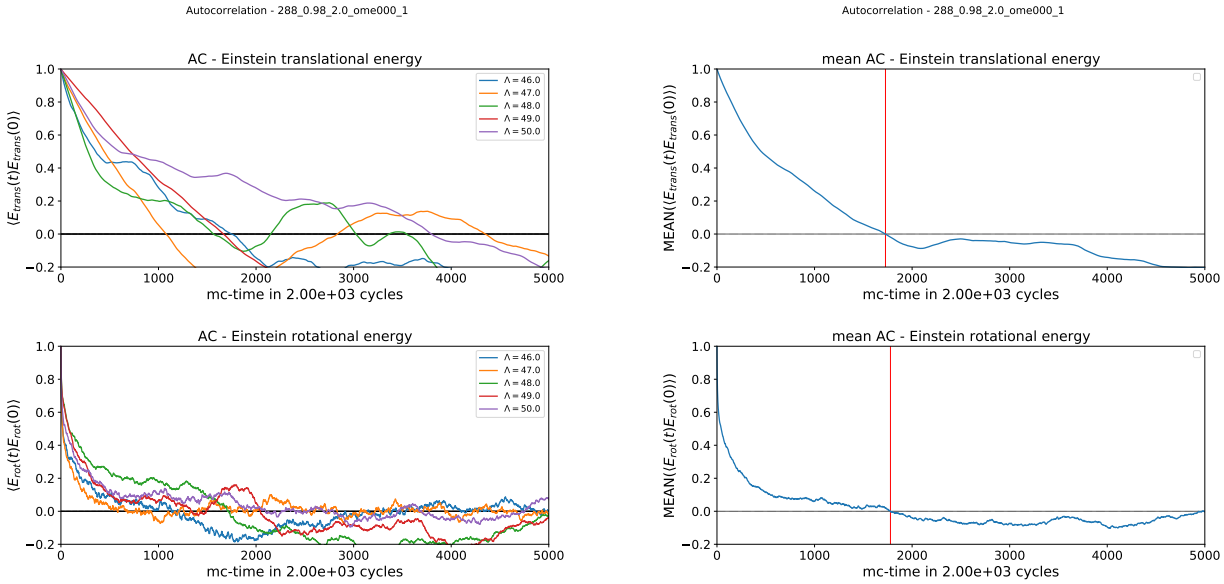


Figure 4.3.: Translational (top) and rotational (bottom) autocorrelation functions of five independent runs (left) and the respective mean (right) for $\lambda_{\mathrm{ID}} = 1$, $\rho^* = 0.98$ at a system size of $N = 288$ in the diagonal lattice state $\omega = 0$. Estimates of the autocorrelation times are indicated by red vertical lines.

The corresponding autocorrelation times that are extracted from the means of the autocorrelation functions for translation and rotational contributions of the Einstein energy are summarized in Appendix A.2. These values serve as reference values for determining the simulation parameters in the computation of free energies (see next section 4.2).

4.2. Simulation Details

Reference State. The reference state, the Einstein Molecule, is defined, as given in part I - eq. (4.24), but with the orientational part of the Einstein Hamiltonian chosen as:

$$\mathcal{H}_{\text{ein,rot}} = \Lambda_{\text{E}} \sum_{i=1}^{N} \left((\phi_i - \phi_{i,0}) \mod \pi \right)^2, \tag{4.1}$$

reflecting the ellipses' $\mathcal{C}2$ symmetry.

Spring Constants. According to the procedure described in part I - section 4.4 - section 4.4.3, appropriate Einstein spring constants were determined. Here, scaled spring constants Λ'_{E} are used, see part I - eq. (4.38). Numerical values for these scaled spring constants $\Lambda'_{\text{E}} = 46, 47, 48, 49$ and 50 are utilized.

Numerical Integration. For the computation of ΔA_2 (part I - eq. (4.36c)) a total number of $n_{\text{sp}} = 100$ sampling points in the numerical integration were used. The numeric integration relies on the Gauss-Legendre quadrature. Before applying the quadrature rules, the integrand is further transformed using the proposed Möbius transformation with a transformation parameter $\xi = -0.9993$ close to the sharp peak at the transformed coordinate $u = -1$ for all systems (see part I - section 4.4 - section 4.4.3).

Simulation Parameters. The simulation parameters, such as the total simulation length (`no_cycles`), the sampling step width (`par_step`) and number of discarded cycles (`disc_cycl`) were set according to the autocorrelation time $\tau_{\text{a.c.}}$, estimated in section 4.1.2. The parameter step size `par_step` was set similar to $\tau_{\text{a.c.}}^{\text{trans}}$. The total simulation length was set at approximately a thousand times the parameter step size. For very large systems, i.e. $N = 288$ and $N = 392$, a simulation length shorter than this was realized, due to limited computational resources. Depending on the system size and therefore the total simulation length, the number of discarded cycles is 1% or 4% of the total number of cycles. The autocorrelation differs among the λ-states, and so the simulation parameters given in table 4.1 are dependent on λ_{ID}. Here λ_{ID} is a numbering of the total 100 different λ-values within the interval of integration $\lambda \in [0, 1]$.

	λ_{ID}	no_cycles	par_step	disc_cycl
$N = 32, 72$	1-10	8.0E+08	8.0E+05	8.0E+06
	11-40	2.0E+08	2.0E+05	2.0E+06
	41-100	5.0E+07	5.0E+04	5.0E+05
$N = 128, 200$	1-10	2.0E+09	6.0E+06	2.0E+07
	11-40	1.0E+09	1.0E+06	1.0E+07
	41-100	2.0E+08	2.0E+05	2.0E+06
$N = 288$	1-10 ($\rho^* = 0.98$)	2.0E+09	8.0E+06	8.0E+07
	1-10 ($\rho^* = 0.95$)	3.68E+09−4.80E+09	8.0E+06	8.0E+07
	11-40	2.0E+09	2.0E+06	2.0E+07
	41-100	2.0E+08	2.0E+05	2.0E+06
$N = 392$	1-10 ($\rho^* = 0.98$)	2.0E+09	8.0E+06	8.0E+07
	1-10 ($\rho^* = 0.95$)	2.8E+09−3.36E+09	8.0E+06	8.0E+07
	11-40	2.0E+09	2.0E+06	2.0E+07
	41-100	2.0E+08	2.0E+05	2.0E+06

Table 4.1.: Simulation parameters for ΔA_2 for different system sizes and for different λ_{ID} at densities $\rho^* = 0.95$ and $\rho^* = 0.98$ and for the diagonal and parallel lattice state.

All relevant simulation parameters for the evaluation of ΔA_1 are collected in table 4.2.

Additional Information. For computational efficiency, testing overlap between particle pairs is restricted to nearest neighbours only. For each particle, a total of six next neighbours are used to compute the pair interaction potential.

	no_cycles	par_step	disc_cycl
$N = 32, 72$	5.0E+07	5.0E+04	5.0E+05
$N = 128$	2.0E+08	2.0E+05	2.0E+06
$N = 200, 288$	1.0E+07	1.0E+05	1.0E+05
$N = 392$	2.0E+05	2.0E+05	2.0E+03

Table 4.2.: Simulation parameters for ΔA_1 for different system sizes at densities $\rho^* = 0.95$ and $\rho^* = 0.98$ and for the diagonal and parallel lattice state.

4.3. Results

For each of the five values of the spring constant Λ'_{E} two independent computations were performed, resulting in a total of ten independent computations for ΔA_2 for each lattice state and for each density. For the contribution ΔA_1, in total, five independent computations were made, one for each Λ'_E. For each of the ten computations, a total free energy $A_{\mathrm{tot}} = A_0(\Lambda'_E) + \Delta A_1(\Lambda'_E) + \Delta A_2(\Lambda'_E)$ is computed, where all contributions explicitly depend on the Einstein spring constant Λ'_E. The term $A_0(\Lambda'_E)$ is the free energy of the reference system and is derived for the two-dimensional Einstein Crystal in appendix A.1. The final total free energy is estimated as the mean of these ten computed free energies. The error to this mean is estimated using the standard deviation σ (see part I - chapter 3 - section 3.3).

Truncation Analysis. The truncation analysis (see part I - chapter 3 - section 3.4) was performed for the quantity ΔA_2. The values for $\Delta A_2(tp)$ are computed from the simulation trajectories truncated at the given truncation point tp for every λ-state in the integration interval, $\lambda \in (0, 1)$. In total 21 different truncation points within the interval $\mathrm{tp} \in (0.0, 0.5)$ were considered.
As an example, the truncation analysis of a system size of $N = 392$ at $\rho^* = 0.95$ for the diagonal and parallel lattice state is shown in fig. 4.4. The full results of this truncation analysis are shown in Appendix A.3. For almost all systems, the final quantity $A_{\mathrm{tot}}(\mathrm{tp}) = A_0 + \Delta A_1 + \Delta A_2(\mathrm{tp})$ shows only a weak dependence on the truncation point. Truncation points for the final results were chosen as $\mathrm{tp} = 0.3$ in systems with almost no dependence on the truncation length. In systems with a recognizable dependence, the truncation point was chosen as specified in table 4.3.
Table 4.3 summarizes the final results of the total free energies for each of the two lattice types, parallel (p) and diagonal (d) and the corresponding error estimates for all studied system sizes and densities. Further, table 4.4 and fig. 4.5 collect the computed free energy differences between the diagonal and parallel lattice states.

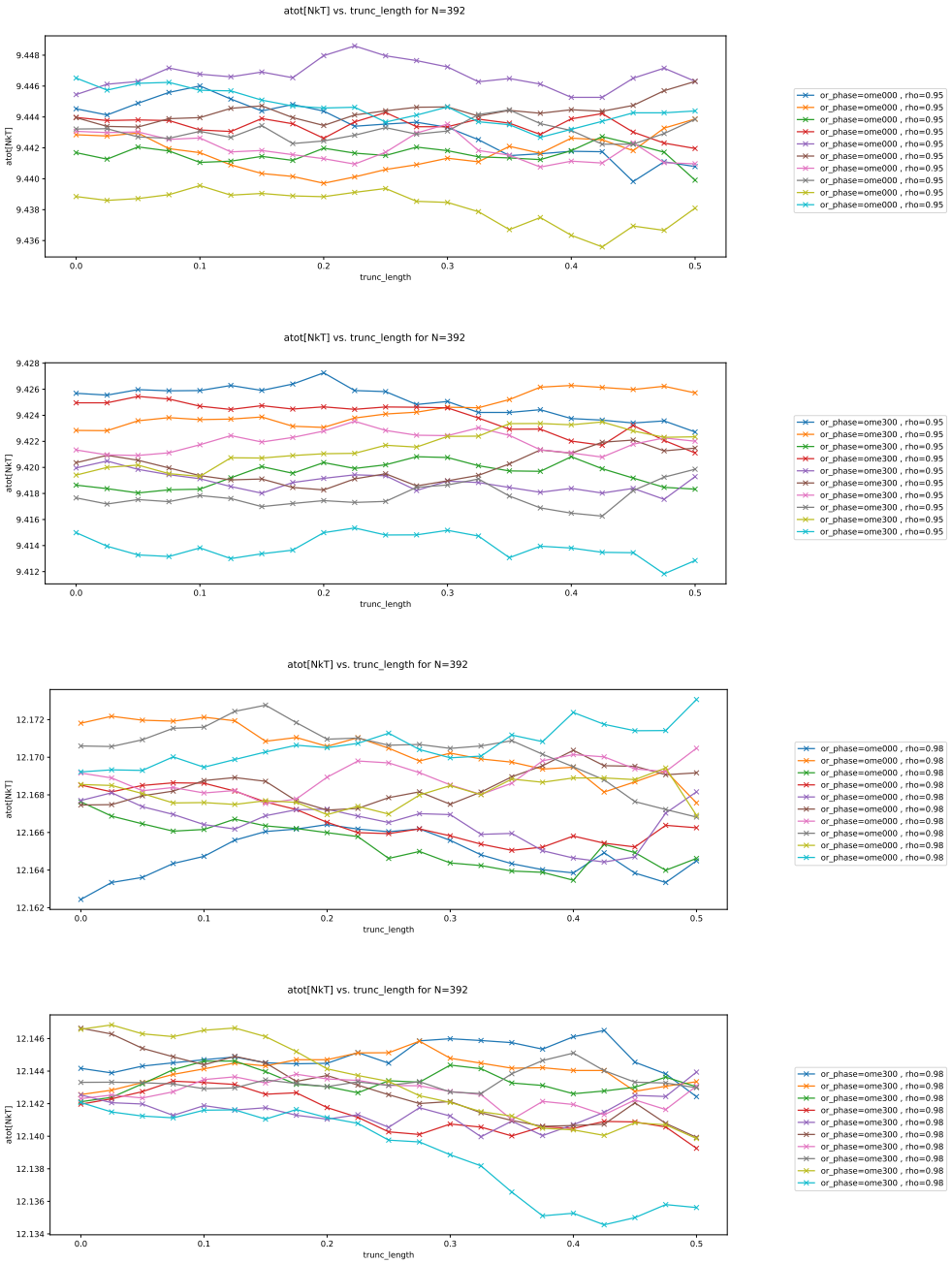


Figure 4.4.: Truncation analysis on $A_{\mathrm{tot}}[Nk_{\mathrm{B}}T]$ for all ten independent simulations individually, at system size $N = 392$. First two panels show $\rho^* = 0.95$, second and fourth $\rho^* = 0.98$. The first and third panel show results of the diagonal lattice configuration ($\omega = 0°$) and second and fourth the results on the parallel lattice configuration ($\omega = 0°$).

N	ρ^*	lattice	tp	$A_{\mathrm{tot}}[Nk_{\mathrm{B}}T]$	σ	copies
32	0.95	d	0.0	9.154(3)	0.005(3)	10
32	0.95	p	0.0	9.057(8)	0.011(7)	10
32	0.98	d	0.0	11.753(9)	0.006(5)	10
32	0.98	p	0.0	11.589(0)	0.011(1)	10
72	0.95	d	0.0	9.318(4)	0.006(9)	10
72	0.95	p	0.0	9.267(7)	0.005(6)	10
72	0.98	d	0.0	11.980(3)	0.005(0)	10
72	0.98	p	0.0	11.897(2)	0.005(1)	10
128	0.95	d	0.3	9.383(9)	0.004(5)	10
128	0.95	p	0.3	9.347(6)	0.005(6)	10
128	0.98	d	0.3	12.075(8)	0.004(5)	10
128	0.98	p	0.3	12.020(9)	0.004(5)	10
200	0.95	d	0.3	9.413(9)	0.004(2)	10
200	0.95	p	0.3	9.384(8)	0.004(9)	10
200	0.98	d	0.3	12.124(5)	0.004(7)	10
200	0.98	p	0.3	12.082(7)	0.004(7)	10
288	0.95	d	0.3	9.436(1)	0.003(5)	10
288	0.95	p	0.3	9.408(1)	0.005(0)	10
288	0.98	d	0.3	12.151(5)	0.005(8)	10
288	0.98	p	0.3	12.120(3)	0.003(4)	10
392	0.95	d	0.4	9.442(4)	0.002(3)	10
392	0.95	p	0.4	9.420(7)	0.003(5)	10
392	0.98	d	0.4	12.167(9)	0.003(0)	10
392	0.98	p	0.4	12.141(7)	0.002(9)	10

Table 4.3.: Results for average free energies A_{tot} and standard deviations, σ from a total of ten copies for densities $\rho^* = 0.95$ and 0.98 in parallel (p) and diagonal (d) state.

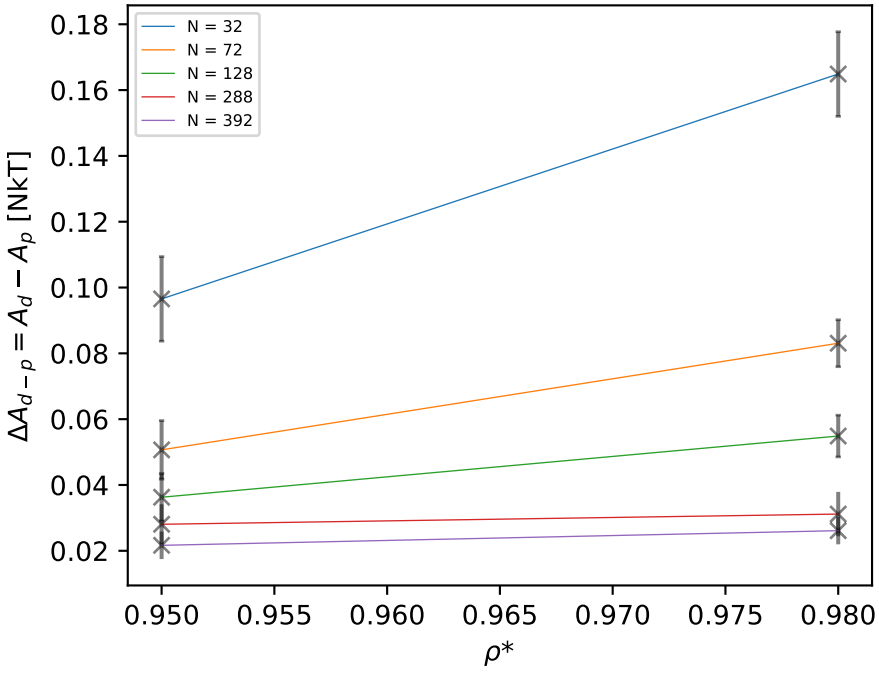


Figure 4.5.: Resulting free energy differences between the diagonal and parallel lattice state, ΔA_{d-p}, for all system sizes and at densities $\rho^* = 0.95$ and $\rho^* = 0.98$.

N	ρ^*	$\Delta A_{\text{d-p}}[Nk_{\text{B}}T]$	σ	copies
32	0.95	0.096(5)	0.012(9)	10
32	0.98	0.164(9)	0.012(9)	10
72	0.95	0.050(7)	0.008(9)	10
72	0.98	0.083(1)	0.007(1)	10
128	0.95	0.036(3)	0.007(2)	10
128	0.98	0.054(9)	0.006(3)	10
200	0.95	0.029(1)	0.006(5)	10
200	0.98	0.041(8)	0.006(7)	10
288	0.95	0.028(1)	0.006(1)	10
288	0.98	0.031(2)	0.006(7)	10
392	0.95	0.021(7)	0.004(2)	10
392	0.98	0.026(1)	0.004(2)	10

Table 4.4.: Results for average free energy differences $\Delta A_{\text{d-p}}$ and standard deviations, σ from a total of ten copies for densities $\rho^* = 0.95$ and 0.98 at different system sizes.

4.4. Analysis of Simulation Trajectories

Here a closer look at the simulation trajectories for prediction of the total free energies A_{sol} is made. The simulation trajectories for one of the ten independent simulations in the system of size $N = 288$ in the diagonal lattice state $\omega = 0°$ is shown for values of $\lambda_{ID} = 1$, 50 and 100 in fig. 4.6. From this analysis, the slow relaxation behaviour of the lattice at $\lambda \approx 0$ becomes evident once more. A full account of this analysis on the system with $N = 288$ in both lattice states and at both densities can be seen in appendix A.4.

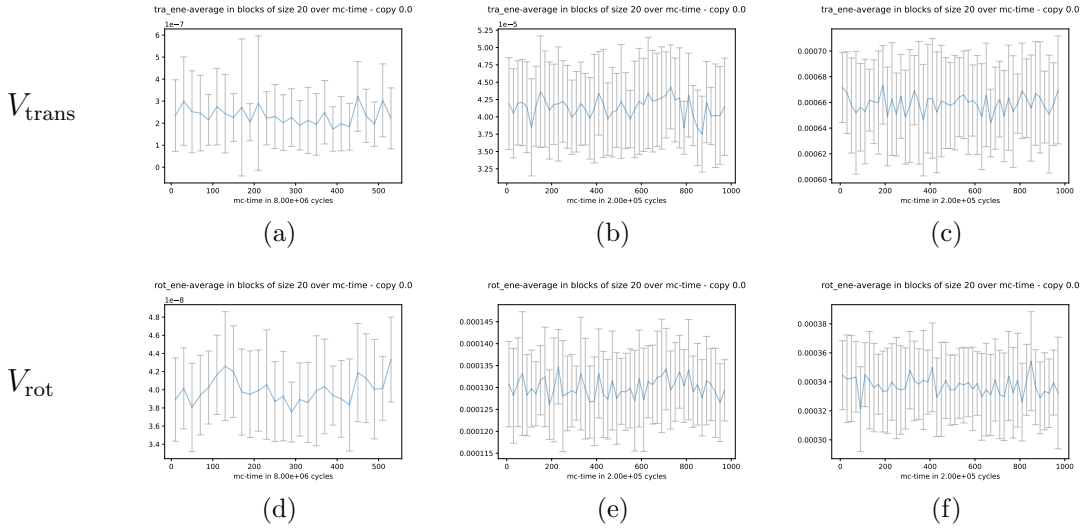


Figure 4.6.: A system of $N = 288$ particles in the diagonal lattice state ($\omega = 0°$) at $\rho^* = 0.95$. Top row: trajectories of the translational Einstein energy - contribution $\lambda\Lambda'\Delta R^2$ to the systems energy for different values of $\lambda \in (0, 1)$, given by their IDs $\lambda_{\text{ID}} = 1$, 50 and 100 (from left to right). Bottom row: The same trajectories as in the top row but for the rotational Einstein energy contribution. The x-axes are given in units of the sampling step size (=8E+6, 2E+5 and 2E+5 cycles from left to right). The block-analysis is performed with a block size of $s_{\text{block}} = 20$. Units of the y-axes is $k_{\text{B}}T$.

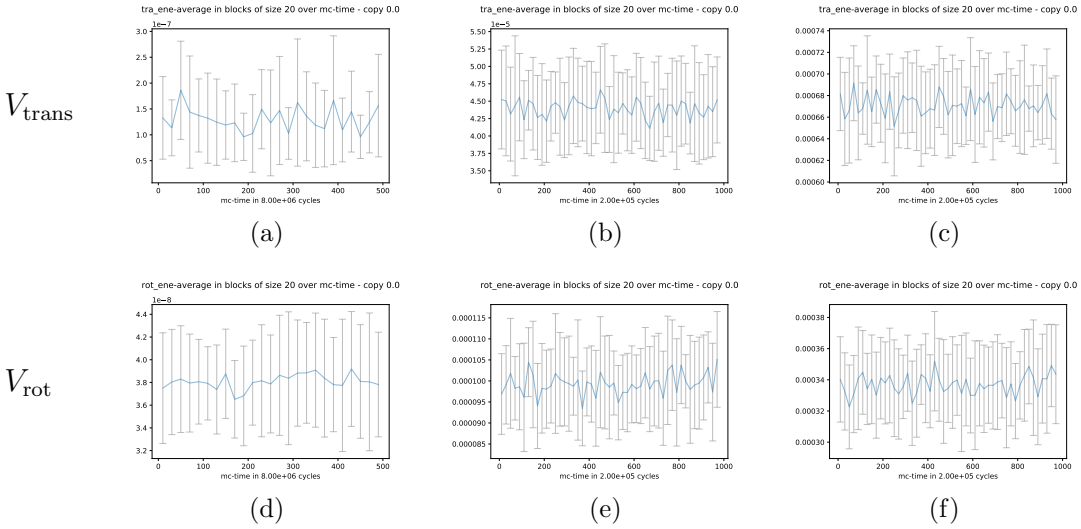


Figure 4.7.: The same system parameters as for fig. 4.6 but for the parallel lattice state $\omega = 30°$.

5. Finite Size Corrections

In this chapter, it is attempted to find the thermodynamic limit of the difference of the total free energies between the diagonal and parallel lattice state, $\Delta A_{\text{d-p}}$. The thermodynamic limit of this quantity can only be defined in its intensive form:

$$\lim_{N\to\infty}\left(\frac{\Delta A_{\text{d-p}}(N)}{Nk_{\text{B}}T}\right) \tag{5.1}$$

In this chapter I use finite-size corrections based on the finite-size model introduced in part I - eq. (4.48) for the total free energy. This model is used for both lattice types, the parallel and diagonal one, indicated via the subscripts $\alpha = p, d$. Taking the difference of the proposed finite-size models (part I - eq. (4.48)) the fitting model for $\Delta A_{\text{d-p}}(N)$ becomes:

$$\frac{\Delta A^{\text{f.s.}}_{\text{d-p}}(N)}{Nk_{\text{B}}T} = (a_p - a_d) + (c'_p - c'_d)N^{b-1}. \tag{5.2}$$

This is the form of the finite size correction model that is used from here on. From this fitting procedure, a value of the parameter b will be obtained, which is very close to zero. Therefore a second fitting through linear regression was applied to $\Delta A_{\text{d-p}}(N)$.

In the following sections, the described fitting procedures are performed, and the results for the fitting parameters are given. A discussion of the results is presented in the last chapter (see chapter 7).

5.1. Fitting the Finite-Size Correction Model

The data for the difference in free energy density $\Delta A_{\text{d-p}}/Nk_{\text{B}}T$ as given in table 4.4 is fitted to the free-parameter model:

$$f(N) = a + cN^{b-1}, \tag{5.3}$$

according to eq. (5.2). With $b \leq 1$ the fitting parameter a corresponds to the thermodynamic limit of $\Delta A_{\text{d-p}}$:

$$\lim_{N\to\infty} \Delta A_{\text{d-p}}/Nk_{\text{B}}T = a = a_{\text{d}} - a_{\text{p}}. \tag{5.4}$$

Results for fitting $\Delta A_{\text{d-p}}$ versus the system size at both densities $\rho^* = 0.95$ and $\rho^* = 0.98$ are shown in fig. 5.1. One plot shows $\Delta A_{\text{d-p}}$ versus N for showing the asymptotic behaviour (i.e. parameter a) as $N \to \infty$ by the red horizontal line. It corresponds to the thermodynamic limit and is the resulting prediction for the free energy difference between the diagonal and parallel configuration for infinite system size. The other plot shows $\Delta A_{\text{d-p}}$ versus N^{-1} for demonstration of the near-linear behaviour, as parameter b is indeed very small. Fit parameters are summarized in table 5.1.

ρ^*	a	b	c
0.95	0.018(2)	-0.065(5)	3.141(8)
0.98	0.011(6)	0.080(6)	3.705(6)

Table 5.1.: Fitting parameters of $\Delta A_{\text{d-p}}$ to $f(N) = a + c \cdot N^{b-1}$ for $\rho^* = 0.95$ and 0.98.

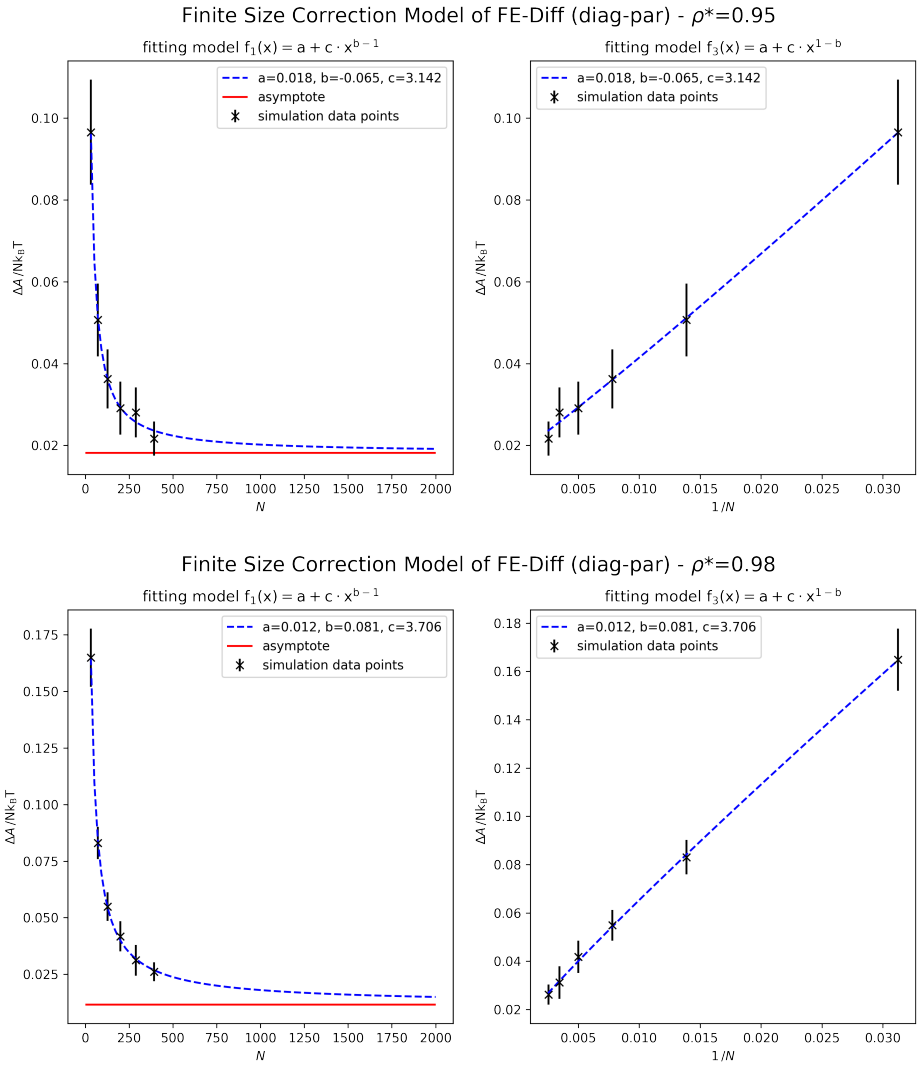


Figure 5.1.: Parameters of fitting model to data points for $\Delta A_{\text{d-p}}/Nk_{\text{B}}T$ versus N^{-1} for $\rho^* = 0.95$ (top,left) and $\rho^* = 0.98$ (bottom,left). Fitting model versus N in the right column, showing the fitting parameter a as the asymptote in red.

5.2. Linear Regression

Since the second fitting parameter $b \approx 0$ (see table 5.1), the finite size correction model eq. (4.48) suggests a nearly linear functional form for $\Delta A_{\text{d-p}}(N)$. In this section, two linear regression models $f_1(N) = k_1 \cdot N + d_1$ and $f_2(N^{-1}) = k_2 \cdot N^{-1} + d_2$ are applied to the data points $\Delta A_{\text{d-p}}(N)$, using the finite size correction model as in eq. (5.2) but now assuming $b = 0$.

The linear regression model is a way of considering also the acquired error bars in the final predictions. The regression model makes predictions on the parameters k and d as well as of the confidence interval Δ_{k} and Δ_{d} for a confidence level of 90%. Numeric values of this procedure are collected in table 5.3 and table 5.2.

Figure 5.2 shows the linear regression models with data points and error bars for both densities $\rho^* = 0.95$ and 0.98. The two linear regression models make consistent predictions for the thermodynamic limit $\lim_{N\longrightarrow\infty} \Delta A_{\text{d-p}}/N k_{\text{B}} T$ which are given together with confidence intervals of 90% as $k_1 \pm \Delta_{\text{k}}$ in the first and $d_2 \pm \Delta_{\text{d}}$ in the second model.

ρ^*	k_1	$\pm\Delta_k(90\%)$	d_1	$\pm\Delta_d(90\%)$
0.95	0.0163	0.0033	2.6103	0.7349
0.98	0.0136	0.0023	5.1130	0.5134

Table 5.2.: Parameters or the linear regression procedure $f_1(N) = k_1 N + d_1$ for $\rho^* = 0.95$ and 0.98.

ρ^*	k_2	$\pm\Delta_k(90\%)$	d_2	$\pm\Delta_d(90\%)$
0.95	2.5433	0.1441	0.0166	0.0021
0.98	4.7870	0.1575	0.0159	0.0023

Table 5.3.: Parameters or the linear regression procedure $f_2(N^{-1}) = k_2 N^{-1} + d_2$ for $\rho^* = 0.95$ and 0.98.

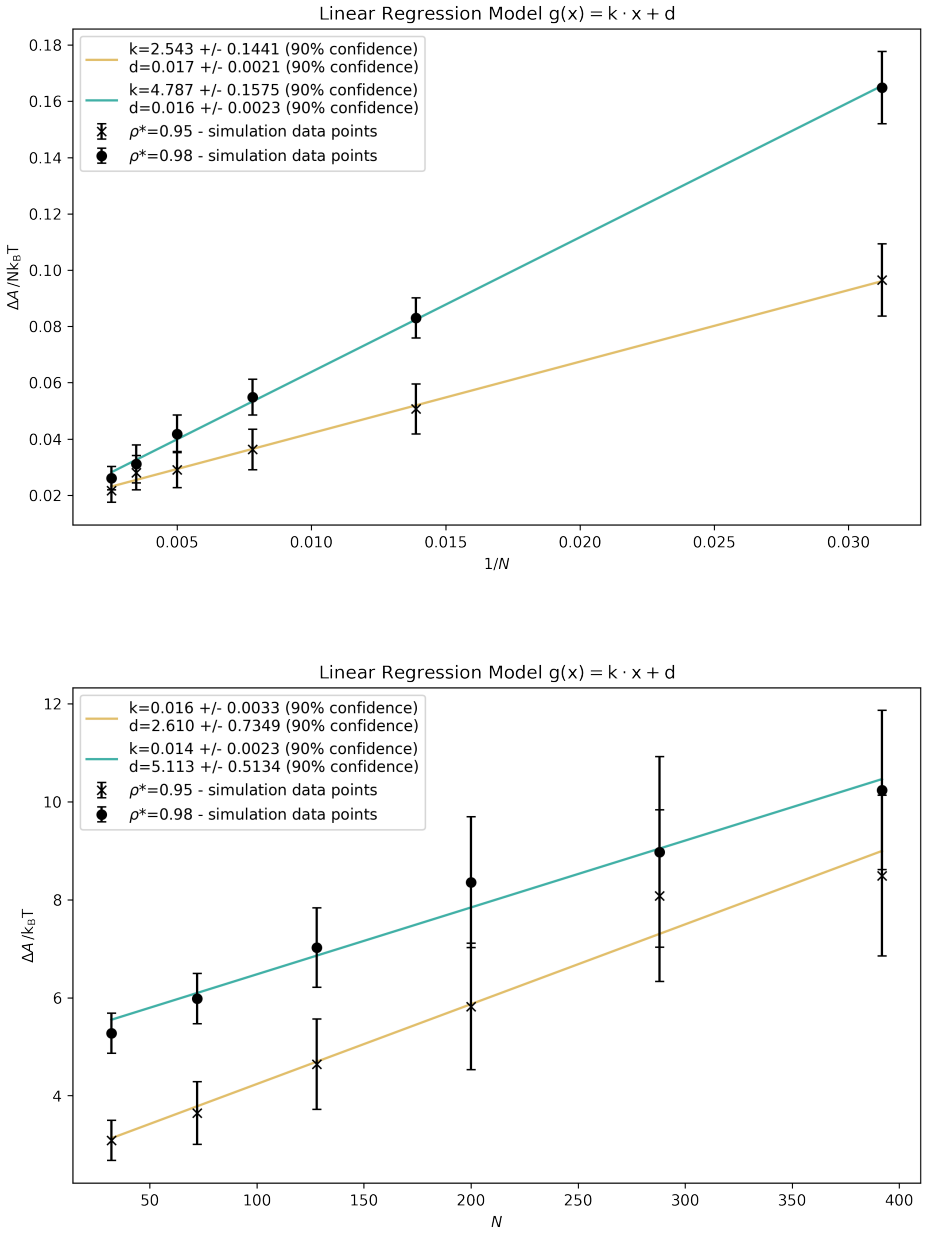


Figure 5.2.: Both linear regression models, $f_1(N)$ in the top and $f_2(N^{-1})$ on the bottom. The linear regression model is given for $\rho^* = 0.95$ in yellow and $\rho^* = 0.98$ in blue.

6. Lattice Deformations

A first attempt to capture the different lattice deformations is made by plotting configurations of particles and indicating their rotational and translational displacements. A colour gradient is used to emphasize particles with strong rotational or translational displacements in the lattice. Plots are shown in fig. 6.1 and fig. 6.2 for a system size of $N = 392$ and at density $\rho^* = 0.98$.

Particle-Translations. For viewing collective particle translations in the lattices, the Einstein crystal energy contributions given as λV_{trans} is used as a measure for each particle's displacement from its lattice site. The depicted configurations are taken from simulations in the mixed state, $\mathcal{H} = \mathcal{H}_{\text{lat}} + \lambda \cdot \mathcal{H}_{\text{ein}}$, with $\lambda \approx 0$. A collection of lattice configurations for the parallel and diagonal lattices of size $N = 392$, are depicted in fig. 6.1 all colored using the same scale.
It can be seen that particles in the parallel lattice configuration undergo collective displacements of greater magnitude than in the diagonal lattice configuration. Further, these snapshots make clear that these collective displacements are mostly oriented along the x- and y-directions in both lattice types. This could be a defining lattice property, but equally, there might also be different directions of wave propagation that are suppressed within the simulation box due to incompatibility with the periodic boundary conditions.
The predominant lattice deformation in the diagonal lattice seems to be the collective shifting of chains of ellipses, i.e. neighbouring ellipses in x-direction.
The above analysis can serve only as a starting point for a considerably more detailed analysis to answer questions on the characteristic vibrational modes, directions of propagation and wave properties, such as the existence of transversal versus longitudinal waves.

Particle-Rotations. For detecting collective particle rotations, the quantity ϕ_i^2, where ϕ_i is the i-th particle's orientation with respect to the $x-$axis, is used in a colour-coded representation. A collection of snapshots can be seen in fig. 6.2. Here, larger orientational freedom of particles in the parallel lattice can be clearly seen. In both lattices, the re-orientations seem to occur collectively. While for the parallel lattice, these collective orientational displacements occur in columns (i.e. along the vertical axis for the here presented snapshots) and in rows (i.e. along the horizontal axis) of ellipses.

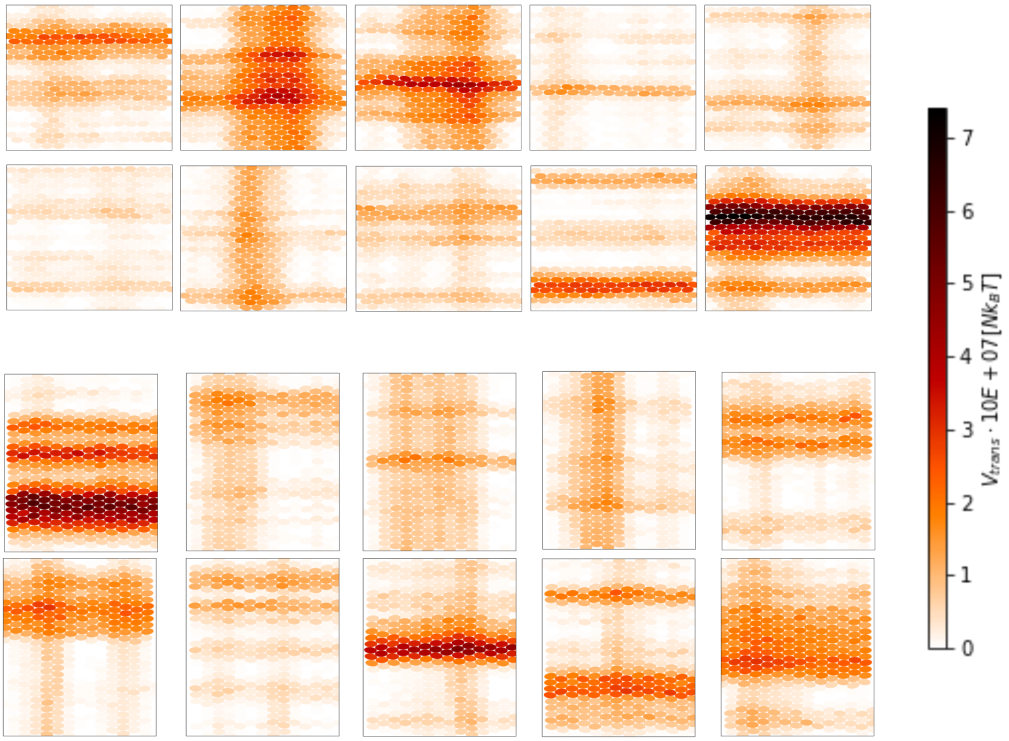


Figure 6.1.: Lattice deformations in the diagonal (top panel) and parallel (bottom panel) lattice according to positional displacement, using each particle's contribution $\lambda\Lambda\Delta R_i^2$ to the Einstein translational energy as the color code, with ΔR_i being the i-th particle's squared displacement. The ten snapshots for each lattice state depict one snapshot for each of the ten independent simulations on a system of size $N = 392$ particles at density $\rho^* = 0.98$.

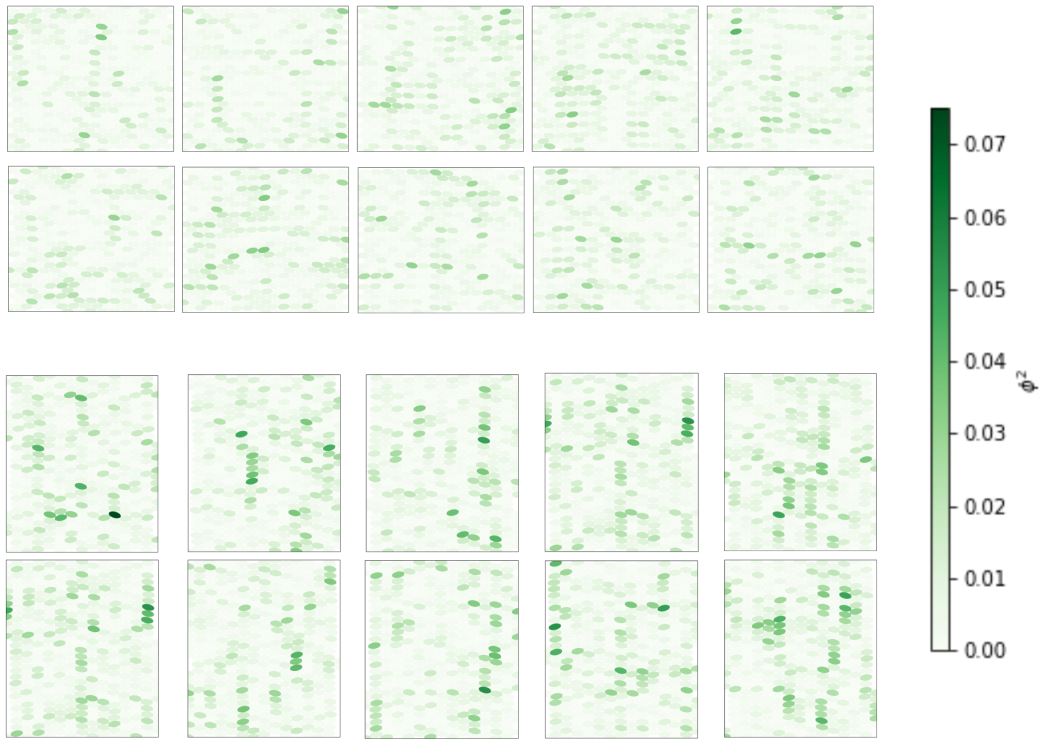


Figure 6.2.: Lattice deformations in the diagonal (top panel) and parallel (bottom panel) lattice according to orientational displacements with reference to the x-axis. Each particle's squared orientational displacement ϕ_i^2 is indicated by a green-scale. The ten snapshots for each lattice state depict one snapshot for each of the ten independent simulations of a system size $N = 392$ particles at density $\rho^* = 0.98$.

7. Discussion & Outlook

The main result of the simulation studies covered in this thesis is the positive free energy difference between the diagonal and parallel lattice state at finite sizes, as shown in chapter 4, as well as in the thermodynamic limit when employing a suitable finite-size correction model, as shown in chapter 5. These results expose the parallel lattice state as the true lattice ground state of the ellipse solid for ellipses with aspect ratio $\kappa = 2$ and at the studied densities $\rho^* = 0.95$ and 0.98.

Results have shown that the difference in the free energy density, $\Delta A_{d-p}/Nk_{\mathrm{B}}T$ is lower for $\rho^* = 0.95$ than for $\rho^* = 0.98$ at finite sizes N. In the thermodynamic limit however $\Delta A_{\mathrm{d\text{-}p}}/Nk_{\mathrm{B}}T$ is lower for the higher density. The result in the thermodynamic limit is therefore conclusive, with the theoretical limit of the free energy difference going to zero as the density $\rho^* \longrightarrow 1$.
Further, the free energy difference at densities $\rho^* < 0.95$ is expected to remain positive as, eventually, when the nematic phase is entered, the ellipses are also aligned in parallel. This phase in particular does not appear for $\kappa = 2$ but only for $\kappa \gtrsim 2.4$ [38][42]. However, the preference for the parallel alignment within the nematic phase (for $\kappa \gtrsim 2.4$) and within the crystalline phase (for $\kappa = 2$) seems conclusive.
From qualitative results on lattice deformations, presented in chapter 6, it is hypothesized that the advantage of the parallel lattice state stems from wider rotational freedom of individual particles when aligned in a parallel manner. In the parallel lattice state, individual particles achieve larger orientational displacements. Further collective orientational displacements occur in columns of ellipses in the parallel lattice, whereas in the diagonal lattice state, these collective orientational displacements occur in lines but not with the same magnitude.

Open questions regarding the parallel and diagonal lattice state are numerous. In an attempt to uncover the importance of the entropy contributions originating from rotational versus translational movement of the particles alone, it could be interesting to compute free energy differences of toy models: one of them with only translational degree of freedom for the ellipse particles, the other one with only the rotational degree of freedom.
A more in-depth study of the lattice dynamics could shed more light on the reasons for the ellipses' preference for the parallel state. The hypothesis drawn from the present study is that the local free volume maximization is best in the parallel configuration. However, non-local dynamics, such as lattice vibrations, seem to

be more pronounced in the diagonal configuration. These two opposing effects might need more thorough investigation, specifically in light of the simulation box sizes, as the emergence of vibrational modes in the lattice is believed to be strongly dependent on the simulation box size and shape in relation to periodic boundary conditions. Therefore studying the effect of different box shapes and sizes, also including boxes that are not of a rectangular geometry, could lead to interesting insights.
Finally, it should be mentioned that the framework for studying the hard ellipses with $\kappa = 2$ could be easily extended to study the lattice conformations of the super-ellipses. Numerical overlap criteria for this shape are proposed in [47].

Part III.

Patchy Elliptic Particles - A Self-Assembly Study

1. Introduction

Patchy particle models have their roots in the 1980s, when primitive models for molecules such as water [48][49] were invented as a tool for trying to understand associated liquids in general [50]. In specific, a considerable amount of effort was spent to understand water and methanol from simulation-based approaches but also within theoretical frameworks [51]. These primitive models for molecules came with a hard-sphere core mimicking the very short-range repulsive forces between the constituents of the liquid as well as with localized and attractive short-range potentials modelling the bonding forces between the liquid molecules.

Through vast advances in the synthesis of colloidal and patchy particles beginning in the 1990s, patchy particles were realized in experiments on the micro- and nanometer scale. In 1987 Casagrande introduced the first synthesized Janus particles [52] as spherical colloids with one hemisphere hydrophilic and the other hydrophobic. Since then, various types of Janus particles have found a wide range of applications in the chemical industry [53][54]. Since the first synthesis of a Janus particle, researchers have succeeded in gaining very high control over the particle's shape, size and patch placements [54][55] in the synthesis process of colloidal patchy particles. Surface functionalizations are no longer restricted to hemispheres like in Janus particles but can be placed in almost arbitrary geometries onto colloidal particles via synthesis methods such as particle lithography or templating, to name a few [54]. These techniques allow for controlled surface functionalization of particles where patchy surface functionalization can be realized for example via DNA-strands or proteins [54]. Contrary to applying patches onto existing colloids, another route to synthesizing patchy particles is the synthesis from colloidal clusters, where some of the cluster particles act as patches [56][57].

The emerging structures from directed self-assembly of colloidal and patchy particles on surfaces can, for example, be used for lithographic masking and the fabrication of nanostructured materials [58]. Further, they can be used as templates for colloidal crystal growth, called epitaxy. Self-assemblies into open structures hold potential applications as selective membranes and filters [59]. Further, because of their optoelectronic properties, colloidal crystals in two- and three-dimensions can be used as photonic materials [60][61] and have found many applications, for example, as photodetectors [62]. Authors in [63] have given colloidal topological insulators as an example for a potential future application.

Applications of these new types of materials are manifold [58] where the applications given here are only a few examples. The interest in colloidal patchy particles has grown significantly as these types of particles are promising units for bottom-up materials design. The versatility of patchy particles with regard to different shapes, patch geometries, patch types and interactions opens a huge field of possibilities for designing new types of material.

Here, computational methods, such as molecular simulations and recently also machine-learning approaches [64], are an important tool for mapping phase diagrams, exploring design principles of particles and generally understanding the principles governing particle assemblies better. Recently [64][65] reverse- engineering methods became of interest, where methods are developed to find particle models matching an assembly with certain desired structural properties. Various computational efforts to understand the assembly processes of patchy particles better are therefore an important predecessor for real-life implementations.

Here again, simulation-based approaches provide a potential first step when exploring the vast possibilities of the assembled structures. Numerous efforts on studying colloidal and patchy particle assemblies in two and in three dimensions have been made, with just an overview given in literature reviews such as [65][66][67][68].
In two dimensions specifically, Karner, Bianchi and Dellago [69] intensively investigated the tilings of patchy rhombi on surfaces, and described design principles for those patchy rhombi in order to get assemblies of open and closed lattices. Further, assembly of patchy rhombi in chains was investigated [32]. Another study shows the assembly of regular patchy polygon-shaped platelets into porous as well as closed assemblies [70].
In experiments, a type of an open lattice structure in two dimensions has previously been assembled using a spherical patchy particle type, called the triblock Janus particle [71]. These particles assemble into open two-dimensional kagome lattices where these lattice structures were later also observed in computer simulations [72].

Colloidal self-assembly of ellipsoids into irregular but also open structures on different interfaces (for example water-air, water-oil) have been achieved in experiments by [73] using polystyrene ellipsoids of aspect ratio $\kappa \approx 5.1$. In these assemblies, one prevalent type of the colloid ellipsoids assembly is the open triangle shape with contact of the ellipsoids at the tips. Enforcing this type of arrangement using attractive patches could lead to stable porous lattice structures on surfaces. The stabilization of such patchy ellipses into regular porous two-dimensional lattices is one of the main results in the following self-assembly study. Further, a synthesis method of polymer ellipsoidal particles for large-scale fabrications was proposed in 2017 [74], making it a promising particle type for

future applications.

Within the work of my PhD, a patchy ellipse model combining anisotropic interactions via extended patches and anisotropic shape was put forward [75]. The patches placed on the hard-core ellipse are of Kern-Frenkel type [76], i.e., extended patches with constant depth and constant short-range interaction with angular dependency. For ellipses, an approximate constant short-range potential was designed by using osculating circles at the ellipses' vertices. The proposed model is shown to be very versatile in specific patch designs, in terms of radial and angular interaction range as well as the placements and combination of patches on the elliptic core.
Further, the self-assembly of two types of patchy ellipses with two opposed patches is studied. With these two patchy ellipses differing solely in the angular extent of the patches, very different self-assembly processes and structures are observed.

Throughout the next pages, chapter 2 discusses the patchy ellipse model, chapter 3 presents two specific particle models, MC-simulation details and specifics of analysis that are the basis for the self-assembly studies conducted on these two types of patchy ellipses. The results of these self-assembly studies are presented in chapter 4 and chapter 5. Finally, chapter 6 will give a short summary and discusses outlooks for potential future studies.

2. Elliptic Patchy Particle Model

The newly proposed model of patchy ellipses [75] was put forward as a model that captures shape- and interaction anisotropy of particles in two dimensions. The shape of the ellipses is controlled by the aspect ratio κ. The interaction anisotropy enters the model through patches of Kern-Frenkel type potential with an approximate constant range square-well interaction. The model offers four positions for possible patches, which are located at the vertices and co-vertices of the ellipse, as can be seen in fig. 2.1.

2.1. Model Description

This model was designed as an attempt for the generalization of the patchy disk model with Kern-Frenkel type potential [76] to anisotropic hard core interactions. We introduce the model's anisotropy parameter κ which is defined as the aspect ratio of the elliptic core particle. The core particles interaction is modelled as a hard-core interaction for which the Vieillard-Baron criterion for detecting overlaps of ellipses [37], as described in part II - chapter 2 was used.
The particle's patches are modelled as Kern-Frenkel type patches and are added to the hard elliptic core centered at the midpoint of the great- and small osculating-circles of the ellipse's vertices and co-vertices. Hence the proposed model provides four patch positions - for patches at the vertices ($i = a$) and for patches at the co-vertices ($i = b$) - with the usual model parameters of the Kern-Frenkel patch potential, which are the patch amplitude θ_i, interaction range δ_i and the square-well depth ϵ_i.
A representation of the particle's parameters for one patch at each vertex can be seen in fig. 2.1.

2.2. Approximation of Constant Range Potential through Osculating Circles

The osculating circles are the best approximation using a circular arches to the ellipse. Hence also the constant range interaction potential is only approximately of a perfect constant range. Some limit of this model must be specified. Therefore a function τ_i is introduced to capture the relative deviation of the present model from a perfect constant range interaction.
First, a full definition of the given geometry, the elliptic curve and osculating

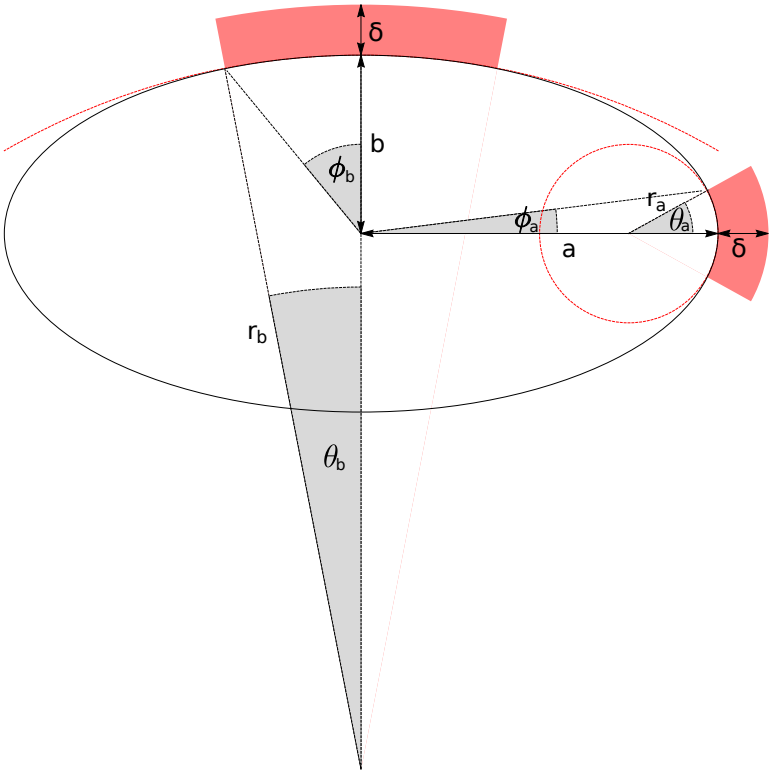


Figure 2.1.: Model specifications of the elliptic patchy particle (EPP): elliptic hard-core particle (in solid black lines) and Kern-Frenkel type patches centered at the vertex's and co-vertex's osculating circles $K_{a,1}$ and $K_{b,1}$ (in red lines), published in [75]

circles at vertices is given. Taking an elliptic particle with its center at the origin in and vertices aligned with the x- and y-axes of the reference frame S, the particle's shape is described by:

$$E: \quad \left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 = 1, \tag{2.1}$$

with a and b representing the semi-major and the semi-minor axes respectively. The aspect ratio κ of the ellipse is defined as $\kappa = \frac{a}{b}$.
The four osculating circles, one for each vertex are named $K_{a,1}$, $K_{b,1}$, $K_{a,2}$ and $K_{b,2}$. Equations for $K_{a,1}$ and $K_{b,1}$ in S are given as:

$$K_{a,1}: \quad \Big(x - (a - r_a)\Big)^2 + y^2 = r_1^2 \tag{2.2}$$

$$K_{b,1}: \quad x^2 + \Big(y - (b - r_b)\Big)^2 = r_2^2, \tag{2.3}$$

with center points $M_{a,1}$ and $M_{b,1}$ and radii r_a and r_b:

$$M_{a,1} = (a - r_a, 0)\,, \quad M_{b,1} = (0, b - r_b) \tag{2.4}$$

$$r_a = \frac{b^2}{a}\,, \quad r_b = \frac{a^2}{b}. \tag{2.5}$$

As this model aims to approximate a constant-range potential, an analytical expression for the deviation of the approximating osculating circle from the elliptical arch, defined as τ_a is required. The limits of the patch model can be defined, depending on a certain criterion for a maximum deviation, as will be specified in section 2.3.

In the next sections, analytical expressions for τ_a and τ_b, the deviations of the small and great osculating circle, respectively, from the elliptical arch are derived.

2.2.1. Small Osculating Circles K_a through V_a

A new reference frame S' with origin at the center of the small osculating circle $K_{a,1}$ is chosen. All coordinates and objects with respect to this frame are denoted with primes. An expression for $\tau_a(\phi')$ as the deviation of the approximated osculating circle $K_{a,1}$ through vertex $V_{a,1}$ from the elliptical arch E shall be derived.
Without loss of generality, it is sufficient to analyse a special ellipse with $a = \alpha$ and $b = 1$. The equations of this ellipse E and its small osculating circle $K_{a,1}$ are given by:

$$E: \quad \frac{1}{\alpha^2}\,(x' + (\alpha - \frac{1}{\alpha}))^2 + y'^2 = 1 \tag{2.6}$$

$$K_1: \quad x'^2 + y'^2 = \frac{1}{\alpha^2} \tag{2.7}$$

Transformation of cartesian into polar coordinates, $(x', y') \longrightarrow (r', \phi')$ leads to the expressions for r'_E and r'_{K_1}:

$$r'_E(\phi') = \frac{(\frac{1}{\alpha} - \alpha)\cos\phi' \pm \sqrt{(\alpha^2 - 1)\sin^2\phi' + \alpha^2}}{\cos^2\phi' + \alpha^2\sin^2\phi'} \tag{2.8}$$

$$r'_{K_1}(\phi') = \frac{1}{\alpha} \tag{2.9}$$

The function $\tau_a(\phi')$ defining the deviation of the osculating circle radius from the ellipse radius is then given by:

$$\tau_a(\phi') = r'_E(\phi') - r'_{K_a}(\phi'). \tag{2.10}$$

2.2.2. Great Osculating Circle K_b through V_b

For the calculation of τ_b, the reference frame S centered at the ellipse is chosen. The axes of the ellipse are $a = \alpha$ and $b = 1$ as above, however, the ellipse's orientation is chosen such that a aligns with the positive y- and b with the positive $x-$axis. The equations of the ellipse E and the great circle $K_{b,1}$ from the center point of the ellipse are given by:

$$E: \quad x^2 + \frac{1}{\alpha^2} y^2 = 1 \tag{2.11}$$

$$K_2: \quad \Big(x - (1-\alpha^2)\Big)^2 + y^2 = \alpha^2 \tag{2.12}$$

and in polar coordinates this leads to expressions for r_E and r_{K_2}:

$$r_E(\phi) = \frac{(\alpha^2 - 1)\cos\phi \pm \sqrt{\cos^2\phi - (\alpha^2 - 2)\sin^2\phi}}{\frac{1}{\alpha^2}\sin^2\phi + \cos^2\phi} \tag{2.13}$$

$$r_{K_2} = \alpha^2 \tag{2.14}$$

The deviation of the osculating circle radius from the ellipse radius is then given by:

$$\tau_b(\phi) = r_{K_2}(\phi) - r_E(\phi). \tag{2.15}$$

2.3. Maximum Patch Amplitudes $\theta_{a,max}$ and $\theta_{b,max}$

The actual interaction range of the ellipse's patches at vertex $i = a, b$ is $\delta_i(\phi)$, with ϕ as the angle in reference system S (see Figure 2.1) and is given as:

$$\delta_a(\phi') = (r'_{K_a} + \delta_{a,0}) - r'_E \tag{2.16}$$

$$\delta_b(\phi) = (r_{k_B} + \delta_{b,0}) - r_E, \tag{2.17}$$

with $\delta_{i,0}$ as the approximated constant interaction range of the Kern-Frenkel interaction potential for each patch i.
Hence the interaction ranges δ_i are related to the above-introduced expressions τ_i by:

$$\delta_a(\phi') = \delta_{a,0} - \tau_a \tag{2.18}$$

$$\delta_b(\phi) = \tau_b + \delta_{b,0}. \tag{2.19}$$

The breakdown of the approximation to this constant range interaction will be defined as the relative deviation $\frac{\epsilon_i}{\delta_{i,0}}$ exceeds a predefined ratio ϵ:

$$\tau_a(\phi') < \epsilon\delta_{a,0} \tag{2.20}$$
$$\tau_b(\phi) < \epsilon\delta_{b,0}. \tag{2.21}$$

The maximum patch amplitude $\theta_{\mathrm{max,i}}$ for each vertex is calculated, solving the maximization problem, stated as:

$$\arg\max_{\phi'}\Big(\tau_a(\phi') - \epsilon\delta_0\Big) \tag{2.22}$$
$$\arg\max_{\phi}\Big(\tau_b(\phi) - \epsilon\delta_0\Big). \tag{2.23}$$

This maximization gives the maximum angles $\phi'_{\mathrm{max,a}}$ and $\phi_{\mathrm{max,b}}$ for the small and great osculating circle, respectively. One obtains the maximum patch amplitudes $\theta_{\mathrm{max,a}}$ and $\theta_{\mathrm{max,b}}$ as angles measured from the center of the osculating circles by applying basic trigonometry.

2.4. Examples for Limiting Parameters

In table 2.1 maximum patch amplitudes $\theta_{\mathrm{i,max}}$, corresponding to $\phi_{\mathrm{i,max}}$, as well as the relative surface coverage p_i for two values of ϵ and three different particle models $\kappa = 2$, 4 and 6 are collected. For the relative surface coverage p_i, the ratio of the patches to the full ellipse's arc length was used. More specifically, the definition for p_i relates the patch's and the ellipse's arc lengths within the first quadrant only:

$$p_i = \frac{\theta_i r_{K_i}}{aE(e)}, \tag{2.24}$$

with half the patch's arc length for the angle θ_i in the numerator and the quarter the ellipse's arc length with semi-major axis a and eccentricity $e = \sqrt{1-\kappa^{-2}}$ in the denominator.

2.5. Classification of Elliptic Patchy Particles

The proposed model of elliptic patchy particles offers a great number of choices of patch placements and patch dimensions. Here, I want to propose a classification for particle models of this type, based on the placement of patches.
I determine three classes of patch arrangements, namely solo-patch, pair-patch and mixed arrangements (see fig. 2.2).
The class of solo-patch arrangements comprises four types, type A and type B, with only one of the respective patches, as well as two types of particles with both

		$\theta_{a,max}$	$\theta_{b,max}$	$\phi_{a,max}$	$\phi_{b,max}$	p_a	p_b
$\kappa = 2$	$\epsilon = 0.1$ %	29.30°	11.77°	7.20°	41.69°	0.11	0.36
	$\epsilon = 0.5$ %	45.29°	17.28°	10.86°	55.40°	0.21	0.63
$\kappa = 4$	$\epsilon = 0.1$ %	36.71°	6.03°	2.17°	61.51°	0.04	0.40
	$\epsilon = 0.5$ %	57.60°	8.76°	3.11°	71.55°	0.06	0.59
$\kappa = 6$	$\epsilon = 0.1$ %	43.05°	4.17°	1.09°	70.96°	0.02	0.43
	$\epsilon = 0.5$ %	67.97°	6.04°	1.50°	78.08°	0.03	0.62

Table 2.1.: Examples of the maximum patch amplitudes θ_i, respective angles ϕ_i and surface coverages p_i for ellipses of aspect ratio κ and for two values of the ratio ϵ defining the breakdown of the quasi-constant range potential [75]

patches AB involved. Two types of AB particles must be distinguished, AB+ indicating patch B placed at the next vertex from patch A in positive direction of rotation and AB- indicating patch B placed at the next vertex from patch A in negative direction of rotation.
Within the class of pair-patch arrangements, patches either of type A or B appear only in pairs. This class comprises three particle types, which are AA, BB and AABB particles, indicating a pair of A-patches, a pair of B-patches and a pair of A- and B-patches each, respectively.
Types of the mixed class are particles with a solo patch as well as a pair of patches involved. There are two types of mixed particles, AAB and ABB.
In total, nine types of particles can be classified into three different classes. Particles of the same type can be further distinguished according to patch amplitudes θ_a and θ_b as well as patch interaction ranges δ_a and δ_b.

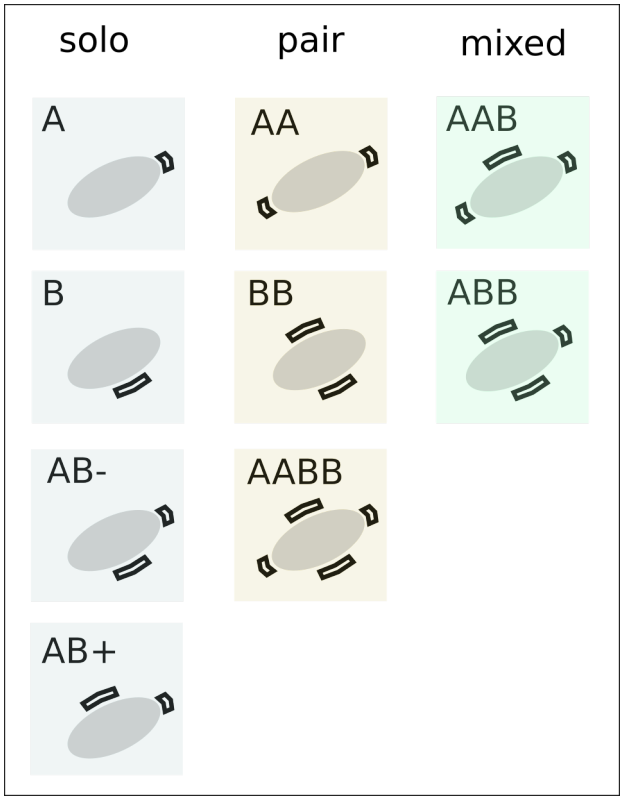


Figure 2.2.: Three classes of patchy particles depicting solo-patch, pair-patch and mixed arrangements, as described in the text.

3. Methods and Model

In the following chapter, the particle models and simulation details underlying the self-assembly studies of two types of patchy elliptic particles are presented.

3.1. A Note on Units

Throughout this simulation study, thermodynamic variables are given in their reduced form, i.e., as dimensionless quantities. Further, simulation units are used in which the Boltzmann constant is set to $k_\mathrm{B} = 1$.

The temperature $T[\frac{\epsilon}{k_\mathrm{B}}]$ in its reduced form is given by:

$$T^* = \frac{T}{\epsilon},$$

where ϵ is the bonding energy of two patchy particles, i.e., the square-well depth of the patch potential given in the Kern-Frenkel model. Since throughout all patchy particle models used in this thesis $\epsilon = 1$, the numeric values of the temperature T and its reduced form T^* are the same.

The chemical potential $\mu[k_\mathrm{B}T]$ in its reduced form is given by:

$$\mu^* = \beta\mu = \frac{\mu}{T},$$

where β is the inverse temperature.

Finally, the particle density in its reduced form is given by:

$$\rho^* = \frac{\rho}{\rho_0},$$

where $\rho_0 = \frac{2}{\sqrt{3}}$ is the density of the closest packed configuration of ellipses.

3.2. Particle Models

The particle models underlying this study are patchy elliptic particles of type AA, i.e., two opposite and symmetric patches located on the vertices of the ellipse's

major axis. The ellipses have an aspect ratio of $\kappa = 2$. For both patches, the depth of the square-well potential is $\epsilon = 1k_BT$. Two ellipse models, one with opening angle $\theta_a = 15°$ and the other with $\theta_a = 45°$ are studied, both with a patch interaction range of $\delta_a = 0.17$. This allows for a maximum of one bond per patch for 15°-particles and a maximum number of two bonds per patch for 45°-particles.

3.3. Simulation details

As a basis of the study, extensive MC-simulations in the grand-canonical ensemble on a grid of a total of 13 values of the chemical potential and eight values of the temperature were performed. At each thermodynamic state point, a total number of 12 independent simulations were performed.

The box shape is chosen to fit the target lattice state. For patchy ellipses with $\theta = 45°$ we assume this target state to be a porous network of ellipses forming loops of size six. We assume this to be the energetic ground state where all bonding sites are saturated. The positions of the ellipses in this network also fit the kagome lattice, nonetheless, I will refer to it as the porous network state with loops of size six. The aspect ratio for simulation boxes fitting this type of target lattice is the same as for a hexagonal lattice of disks. A simulation box with dimensions $L_x = 36$ and $L_y = 31.2$ was therefore chosen, fitting 1296 disks in a hexagonal lattice in the closest packed configuration. This box shape could theoretically host a total of 648 patchy ellipses in the described porous network state.
For the self-assembly simulation of patchy ellipses with $\theta = 15°$ the same simulation box size and shape is chosen. As these types of particles are expected to form chains, a simulation box fitting a specific lattice is not necessary.

The initial particle configuration at each thermodynamic state point is a set of randomly positioned particles in a dilute gas configuration. The dilute gas configuration is then abruptly exposed to the system variables μ^* and T^*. The goal here is to observe the self-assembled structures and possibly the self-assembly process and equilibrium target phases. In order to enhance the system's possibility to explore configuration space faster, the cluster-moving algorithm proposed by Whitelam et. al. [9] is used. Simulating systems of 15°-particles, this algorithm is used in an adapted form, proposing cluster-rotations that mimic chain motions (see part I - chapter 2 - section 2.3).

The relevant MC-simulation parameters are a number of 2.0E+05 initial MC-cycles for $\theta_a = 45°$ particles and 1.0E+05 initial cycles for $\theta_a = 15°$ particles. The initial cycles at the beginning of each MC-simulation are then discarded, and configurations are sampled only during cycles succeeding these initial discarded

cycles. This sampling of configurations is done every 2.0E+03 cycles. Each cycle in the MC-simulation consists of N proposed MC-moves, where $N = 1296$, i.e. the theoretical limit of ellipses fitting the area of the simulation box in a closed-packed configuration. A MC-move in the grand-canonical ensemble is either a translation or rotation move of a single particle proposed with a probability of each 0.38, or further a cluster-rotation or cluster-translation move, each proposed with a probability of 0.095. The remaining 0.05 is the chance for selecting a particle insertion or deletion move. The cluster moving scheme according to [9] (details in section 2.3) uses a parameter $\beta_f = 0.5$.

The final length of simulation runs varies with the thermodynamic state points. We have long runs, with no_cycles $>$ 6.0E+06 at negative chemical potential μ^* were the transition from the gaseous to solid phase is observed, medium simulation lengths, with 2.0E+06 $<$ no_cycles $<$ 6.0E+06 at intermediate chemical potentials and at low temperatures ($T^* < 0.1$). For thermodynamic state-points at high temperatures ($T^* > 0.1$) and at predominantly positive chemical potentials μ^* the simulation lengths for the trajectories are much shorter, at no_cycles $<$ 2.0E+06. Almost all simulations were run for approximately the same time of roughly 20 CPU-days. With roughly the same CPU-time for each simulation run, shorter simulation lengths are achieved for systems including a lot of particles, whereas longer simulation lengths are achieved for systems with fewer particles.

3.4. Truncation Analysis

For each simulation process, a truncation point was defined individually by inspecting the energy, number of particles and energy per particle trajectories. For systems that reach a steady state, i.e., no varying thermodynamic variables E_{pot}, N and E_{pot}/N, the truncation point was located at approximately where this steady state is reached. For systems not entering a steady state, i.e., systems with very slow relaxation behaviour, the truncation point was chosen such that at least the last 100 configurations of the process are included in the final analysis.
For the complete analysis following in this chapter, when taking an ensemble average over an observable, e.g., for the potential energy $\langle E_{\text{pot}} \rangle$, only configurations beyond the truncation point in the simulation process are taken into this average.

4. Results on Self Assembly of Patchy Ellipses with $\theta = 45°$

This chapter presents the results of the self-assembly study on the particle model of patchy ellipses with patch opening angle $\theta = 45°$.
In the following, the ensemble averages of the total energy, number of particles, energy per particle (section 4.2) and the global loop order parameter (section 4.3) shall be examined. Further, the assembly processes from a gaseous phase into a porous network are inspected more closely (section 4.4), and a structural analysis of the assembled networks by means of the radial distribution function is made (section 4.5). Another interesting process observed within the simulations is the process of self-healing, i.e., the rearrangement of particles in a condensed phase in order to achieve a more ordered phase (section 4.6). Overall, it is seen that the temperature, $T^* = 0.16$ brings forward the most ordered loop networks.

4.1. Snapshots & Simulation Trajectories

For elliptic patchy particles with a patch opening angle of $\theta_a = 45°$ it is known from chapter 2 that for each patch, a maximum number of two bonds is possible. An energetic ground state is achieved when all of the particle's patch bonding sites are saturated. The types of regular configurations for the patchy ellipse model at hand fulfilling the condition that all bonding sites are saturated can be concluded from the Euclidean tilings, where the edge of a polygon is characterized as the patchy ellipse particle [77]. Two tilings are compatible with a number of two bonds per patch, which are the hexagonal tiling and the truncated square tiling. Both will result in porous networks with loops of either size six or sizes four and eight.
In the following study, the **porous network of ellipses with loops of size six** - compatible with the hexagonal tiling - is identified as the target structure for most of the studied simulation state-points. When a network is formed with loops including not only loops of size six, I will talk about **loop defects** in the porous network, whereas when the system fails to build a completely saturated network structure, I will call these defects **network defects**. Further, particle configurations at high particle densities, either with an underlying network structure but with the pores being filled with particles or configurations completely lacking a network structure, will be called **crowded configurations**.

In fig. 4.1 a selection of snapshots for thermodynamic state-points, $\mu^* = -7.0$, -2.0 and 5.0 and $T^* = 0.055$, 0.1 and 0.16, is shown. The snapshots represent the last sampled configuration for the related simulation processes and at the respective thermodynamic state-point. Figure 4.1 gives an overview over all characterized phases - a gaseous phase (for $T^* = 0.16$, $\mu^* = -7.0$) and several types of condensed phases. Of these we characterize porous networks close to the energetic ground state (i.e. all bonding sites saturated) for $(\mu^*, T^*) = (-7.0, 0.1)$ and $(-2.0, 0.16)$. The most prevalent loop size in these networks is size six, however, loop-defects occur. Defective porous networks are characterized at $(\mu^*, T^*) = (-7.0, 0.055)$, $(-2.0, 0.055)$ and $(-2.0, 0.1)$ and finally crowded states arise at $\mu^* = 5.0$.

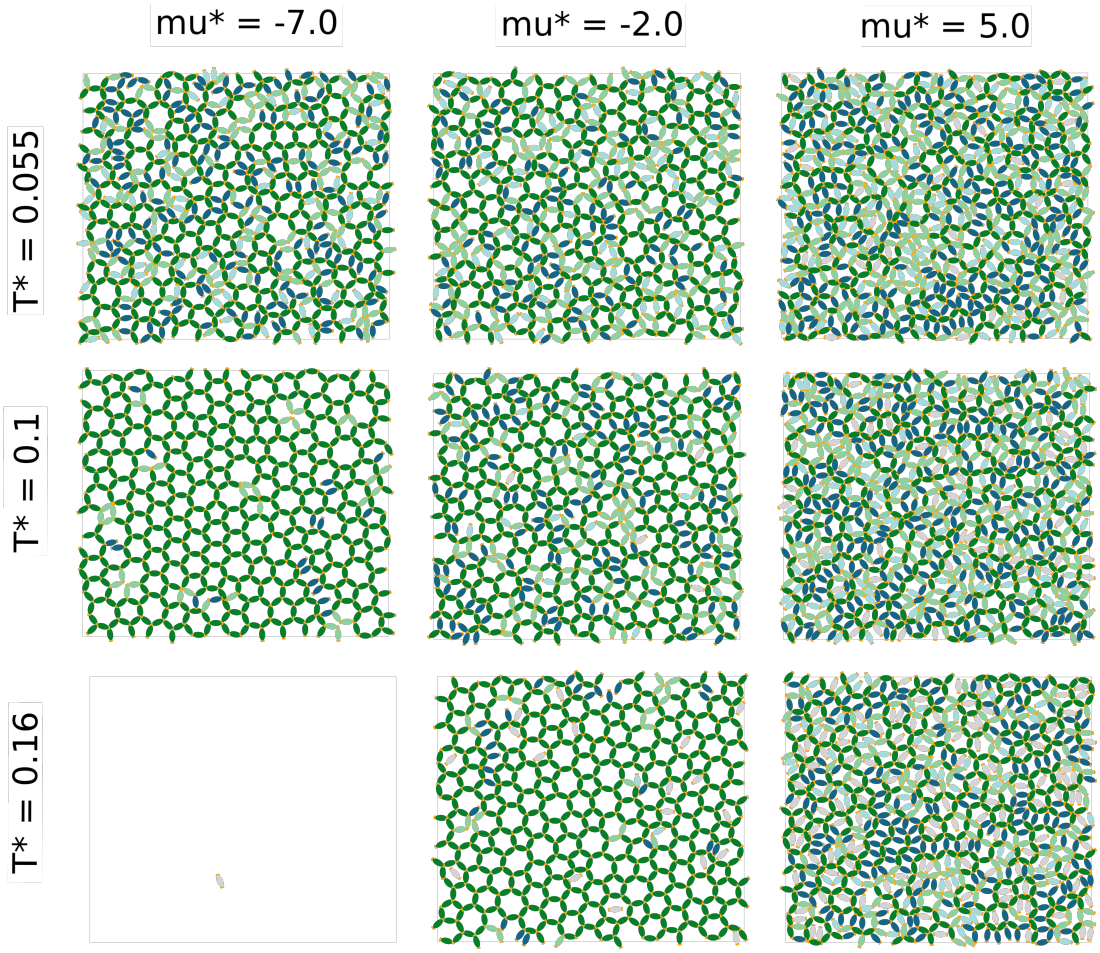


Figure 4.1.: Snapshots for three chemical potentials ($\mu^* = -7.0$, -2.0 and 5.0) and three temperatures ($T^* = 0.055$, 0.1 and 0.16 as labeled) with particles colored according to their bonding energy - no bonds (grey), one bond (light blue), two bonds (dark blue), three bonds (light green) and four bonds (dark green).

Trajectories. A complete record of all simulation trajectories showing the evolution of the total energy, the number of particles and energy per particle over MC-time can be seen in appendix B.1.

4.2. Ensemble Averages - State Variables

In the following, an overview of all thermodynamic state points included in this study is given by presenting state variables such as the potential energy, E_{pot}, the number of particles N and energy per particle E_{pot}/N computed as ensemble averages over all 12 independent trajectories. The results for these ensemble averages are shown in fig. 4.2 to fig. 4.4.

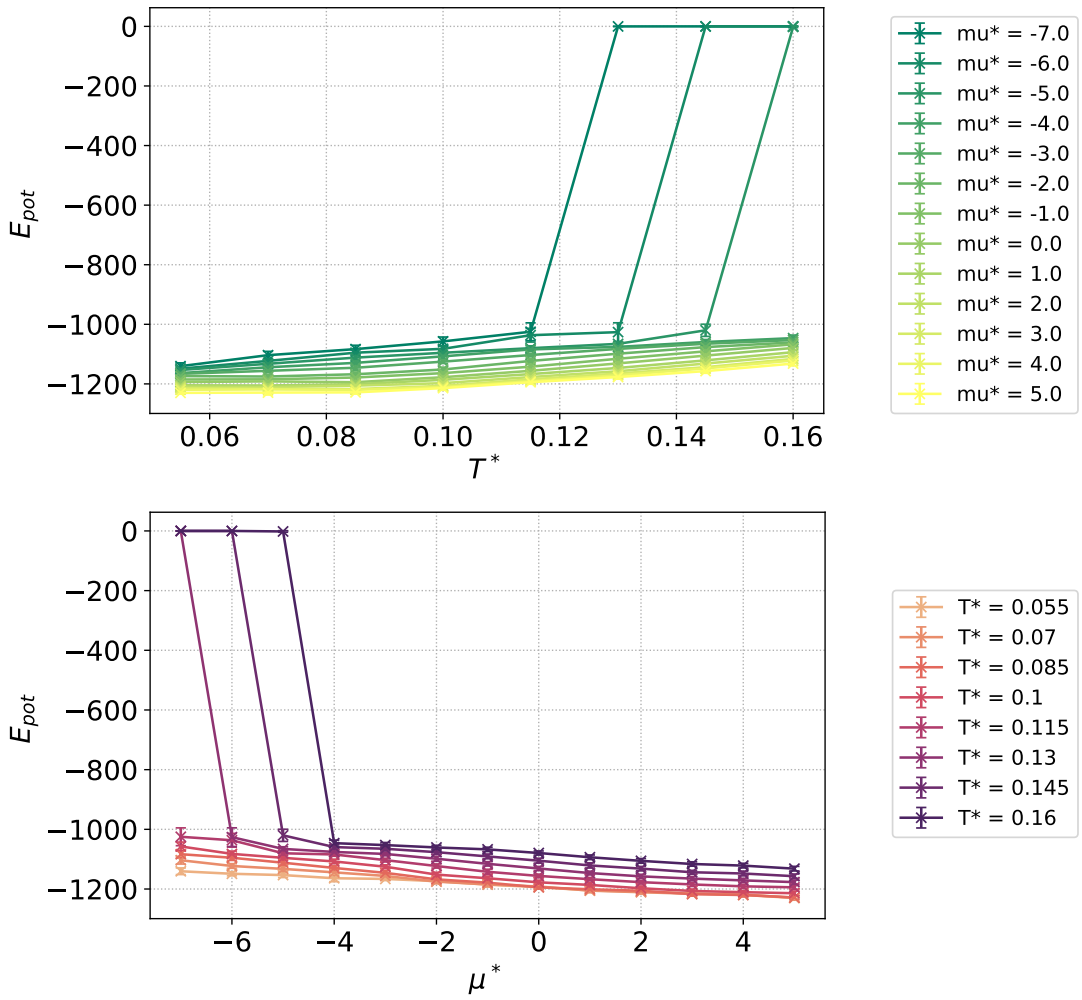


Figure 4.2.: Total energy E_{pot} in units of $k_{\mathrm{B}}T$ for constant μ^* across temperatures T^* (top) and for constant T^* across chemical potentials μ^* (bottom).

Viewing the potential energy, E_{pot} (fig. 4.2) for a single chemical potential, e.g. $\mu^* = -7.0$ along the temperature axis, we see that the potential energy is monotonically increasing with increasing temperature. From a temperature $T^* = 0.115$ to $T^* = 0.13$ a sudden change in the potential energy shows that from the condensed phase with $E_{\text{pot}} \ll 0$ the gaseous phase with $E_{\text{pot}} \approx 0$ is reached. The potential energy for any constant temperature is monotonically decreasing with increasing chemical potential. The characteristic feature of a monotonically increasing E_{pot} with increasing T^*, as well as monotonically decreasing E_{pot} with increasing μ^* is found for all state-points included in this study, while the phase transition from the condensed to gaseous phase can be observed only for the three lowest values of the chemical potential ($\mu^* = -7.0, -6.0, -5.0$) and for lowest three temperatures ($T^* = 0.055, 0.07, 0.085$).

In fig. 4.3 the number of particles is shown in the same representation as for the potential energy. The monotonically decreasing number of particles at constant chemical potential μ^* with temperature increasing T^* goes along with the increase in potential energy for the respective lines of constant μ^*. The states in the (μ^*,T^*)-plane, identified as gaseous states from the potential energy curves, reveal themselves as almost empty phases, with very few to no particles present in the simulation boxes.
From tracing the number of particles along the isotherms, it can be noted that for the highest chemical potential, $\mu^* = 5.0$, the number of particles for all temperatures reaches very similar values. All systems reach their maximum number of particles for this value of the chemical potential.

From the energies per particle E_{pot}/N seen in fig. 4.4 we again follow a line of constant $\mu^* = -7.0$ from low to high temperatures. Up to a temperature $T^* = 0.115$ the energy per particle is decreasing, almost reaching $E_{\text{pot}}/N = -2.0$. For higher temperatures, the system is in the gaseous phase. In this phase $E_{\text{pot}}/N < 0.0$, hints at the formation of small clusters in the otherwise almost empty state. Viewing isotherms in the condensed phases across μ^* and going from $\mu^* = -7.0$ to higher chemical potentials, the energies per particle are increasing along each of these isotherms. At the highest value of the chemical potential, $\mu^* = 5.0$, at which crowded states are observed, E_{pot}/N is almost constant across all temperatures. The lowest energy per particle is observed for thermodynamic state points with temperature, $T^* = 0.16$ and low chemical potentials, i.e., $\mu^* = -7.0$, -6.0, -5.0 and -4.0. At these state-points, the systems almost approach the minimal energy per particle $E_{\text{pot}}/N = -2.0$, which corresponds to exactly four bonds for each particle, i.e. completely saturated bonding sites. The minimal energy per particle $E_{\text{pot}}/N = -2.0$ corresponds to a loop network free of any network defects. As the configurations assembled in the simulations underlying this study are not free of those defects, this theoretical minimum is not achieved.

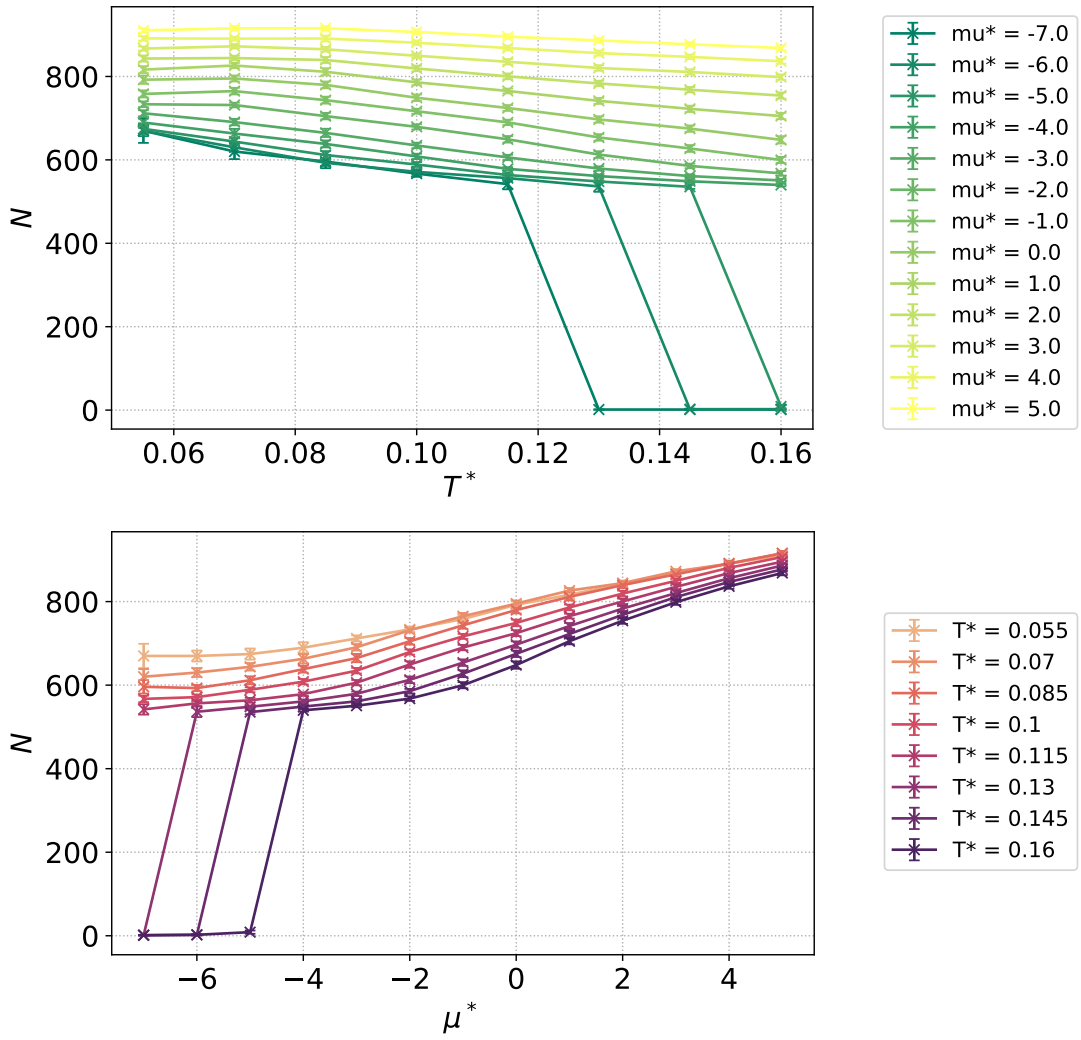


Figure 4.3.: Total number of particles N for constant μ^* across temperatures T^* (top) and for constant T^* across chemical potentials μ^* (bottom).

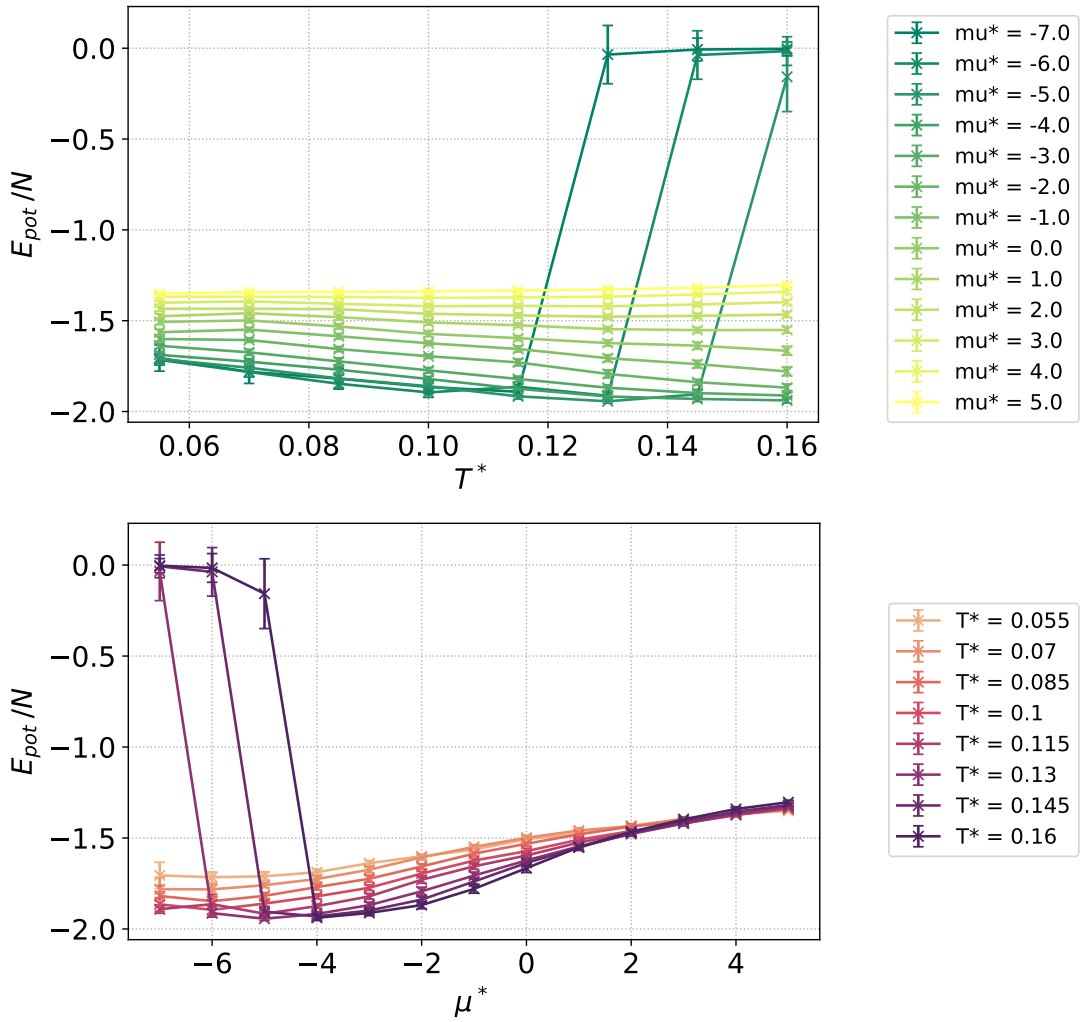


Figure 4.4.: Energy per particle E_{pot}/N in units of $k_{\text{B}}T$ for constant μ^* across temperatures T^* (top) and for constant T^* across chemical potentials μ^* (bottom).

4.3. Structural Analysis - Loop Order Parameter

For a first insight into the structure of the assembled states, the global loop order parameter Γ_l (defined in part I - chapter 5 - section 5.3) is determined. The loop order is a scalar quantity with $\Gamma_l \in (0, 1)$, where $\Gamma_l = 1$ corresponds to a network in which all loops are of size l. $\Gamma_l = 0$ corresponds to the absence of loops. The loop collections for each configuration used to compute the loop order parameter are determined using the loop-detection algorithm (part I - chapter 2 - section 2.4.2).

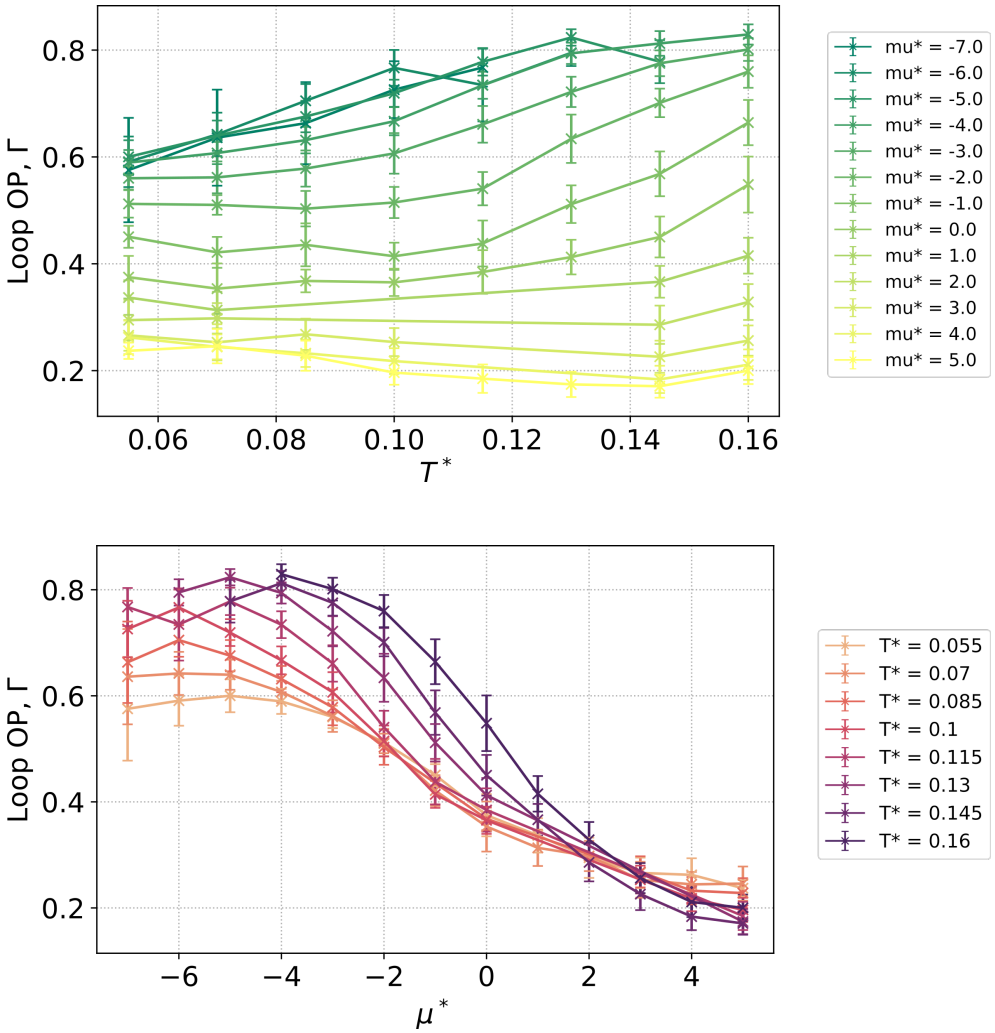


Figure 4.5.: The global loop order parameter Γ for all μ^* versus T^* (top) and all T^* versus μ^* (bottom). Each data-point represents the average of 12 independent simulation processes.

With a loop size of $l = 6$, for the analysis of the loop order parameter Γ_6, I use the log-normal distribution with a mode of $m = 6$ and a standard deviation

$\sigma = 0.5$. This choice of parameters makes the loop order parameter very specific to loops of size six, which is the kind of structural order that we want to analyze in these types of networks.
Figure 4.5 shows all computed loop order parameters Γ_6 for state points in the (μ^*,T^*)-plane. The highest loop order parameter is achieved at negative chemical potentials, $\mu^* = -7.0, -6.0, -5.0$ and -4.0. With increasing chemical potential, the loop order parameter drops to $\Gamma_6 \approx 0.2$ for all temperatures. A global maximum of the loop order parameter is achieved with $\Gamma_6 \approx 0.82$ for the temperature $T^* = 0.16$ and for the chemical potential $\mu^* = -4.0$.

4.4. Analyzing the Assembly Processes into Porous Networks

The self assembly process from a gaseous phase into a porous network is observed at eight different thermodynamic state points, i.e., for a chemical potential $\mu^* = -7.0$ and for temperatures $T^* = 0.055, 0.7, 0.085, 0.1, 0.115$ as well as for the state points $(\mu^*, T^*) = (-6.0, 0.13)$ and $(-5.0, 0.145)$. For observing the trajectories at these state-points, be referred to appendix B.1.
Here, a closer look at the assembly process shall be taken by taking one particular state point, namely $(\mu^*, T^*) = (-6.0, 0.13)$ as an example.

Trajectories and snapshots. The energy, number of particles and energy per particle trajectories for all twelve independent simulation runs at the state point, $\mu^* = -6.0$ and $T^* = 0.13$ are shown in fig. 4.6. Observing these trajectories, it can be seen that the time for the transition from the gaseous into the condensed phase follows stochastic principles. Further, the energy per particle achieves almost $E_{\text{pot}}/N = -2.0$ in all runs. In fig. 4.7 (a) one of these trajectories is shown. Here we follow the self-assembly process into a porous network in one simulation process, with snapshots shown in fig. 4.8, where the black stars in the energy trajectory indicate the MC-times at which these snapshots are taken. Figure 4.7 (b) shows the order parameter N_d (part I - section 5.4), i.e. the proportions of particles with d number of bonds for the same simulation process. Further fig. 4.7 (c) and (d) show the order parameters, Γ_6 and $C^{\max}$, i.e. the loop order parameter and the largest cluster size, for all 12 simulation runs.

From the snapshots shown in fig. 4.8, the formation of several nuclei (see configuration 400) can be seen, and, within this selection of snapshots, starting from configuration 680, we see that one cluster prevails, increasing in size over the course of the following simulation cycles. It is believed that we observe a nucleation process following the classical nucleation theory [78], as the growth of a single nucleus is observed. However, details on whether the growth happens through attachment of single particles or particle clusters, cannot be answered

within the current analysis. An in-depth study of these nucleation processes is necessary before answering these questions.
Although the nucleus is not growing completely free of defects (see fig. 4.9), the main source for defects in configuration 2400 seems to stem from when the nucleus extends over the whole simulation box and the network is trying to close. As will be shown in section 4.6, networks of patchy ellipses at high enough temperatures, have the capacity to rearrange particles and curing these defects.

Figure 4.9 gives a closer insight into the growth of the nuclei at different state points. In these snapshots, the loop-size color scheme is used to highlight loops of sizes six, five and seven. It is seen that a regular structure of loops with size six is dominant in most nuclei, with loop defects of sizes five and seven eventually occurring. For the lowest temperature, $T^* = 0.055$, the nucleus is the most defective, in the sense that no loops of size six are observed in the nucleus. For higher temperatures, such as $T^* = 0.1$, 0.13 and 0.145, the arrangement of ellipses into loops of size six is more stable.

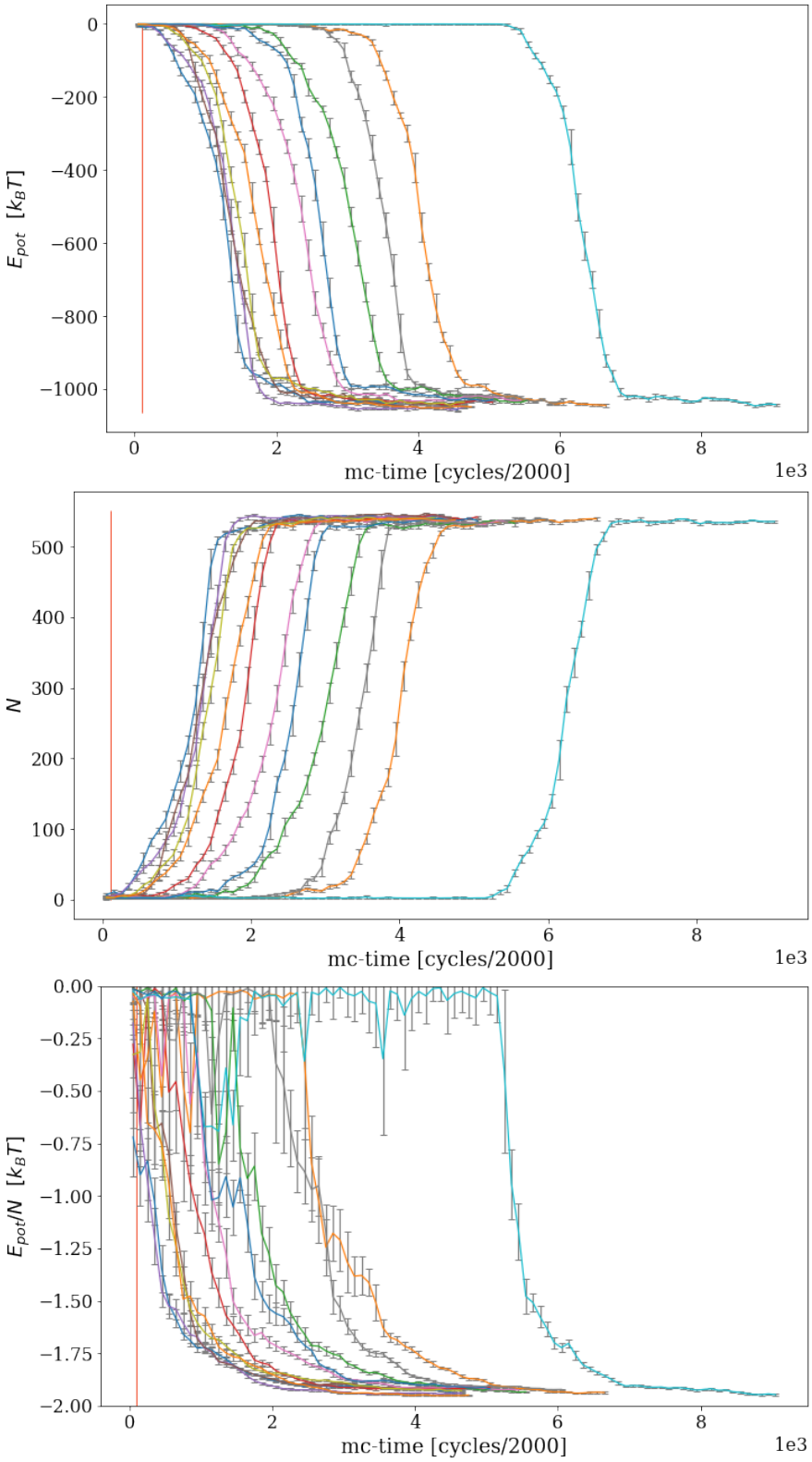


Figure 4.6.: Energy, number of particles and energy per particle trajectories for all twelve independent simulation processes at $T^* = 0.13$ and $\mu^* = -6.0$ as a function of MC-time. Data points and error bars correspond to mean and standard deviation of blocks of size $s_{\text{block}} = 100$.

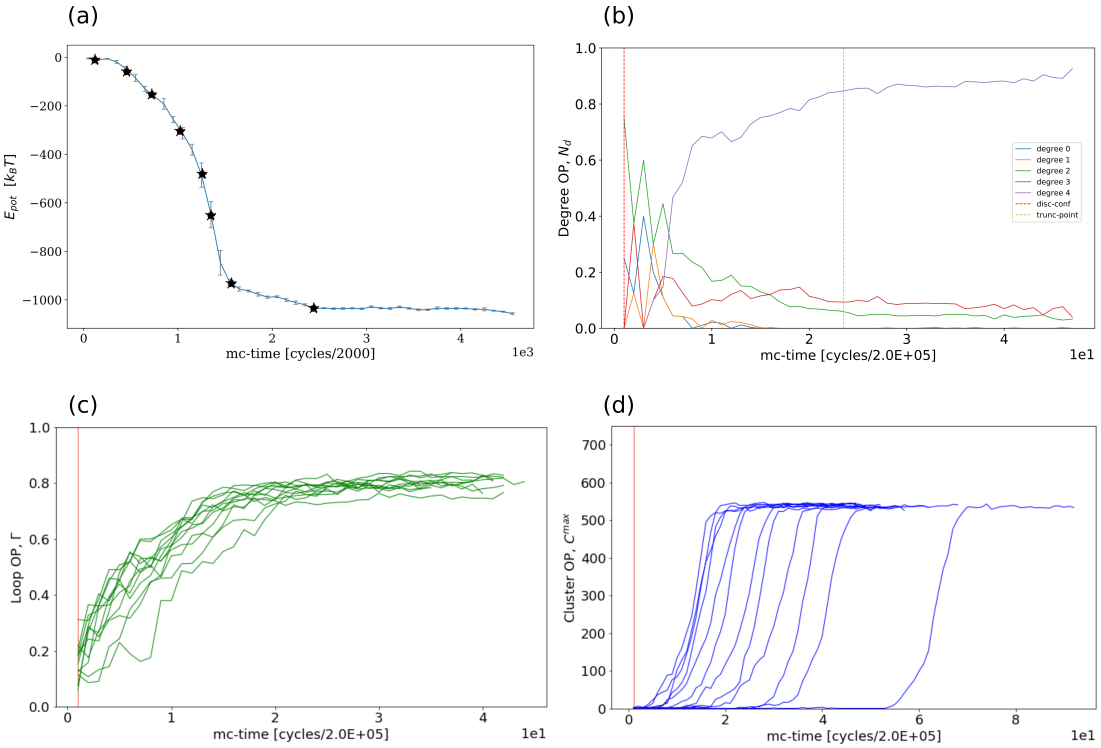


Figure 4.7.: All plots show data of the state-point at $T^* = 0.13$ and $\mu^* = -6.0$.
(a) Energy trajectory for a single simulation process over the MC-time. Data points and error bars correspond to mean and standard deviation of blocks of size $s_{\text{block}} = 100$. Specific configurations indicated by black star symbols are shown as snapshots in fig. 4.8.
(b) Order parameter N_d, showing the proportion of particles with a certain degree, i.e. the number of bonds d.
(c) Loop order parameter for all 12 independent simulation runs over MC-time and finally (d) showing the cluster order parameter $C^{\max}$, i.e. the size of the largest cluster again for all 12 simulation runs.

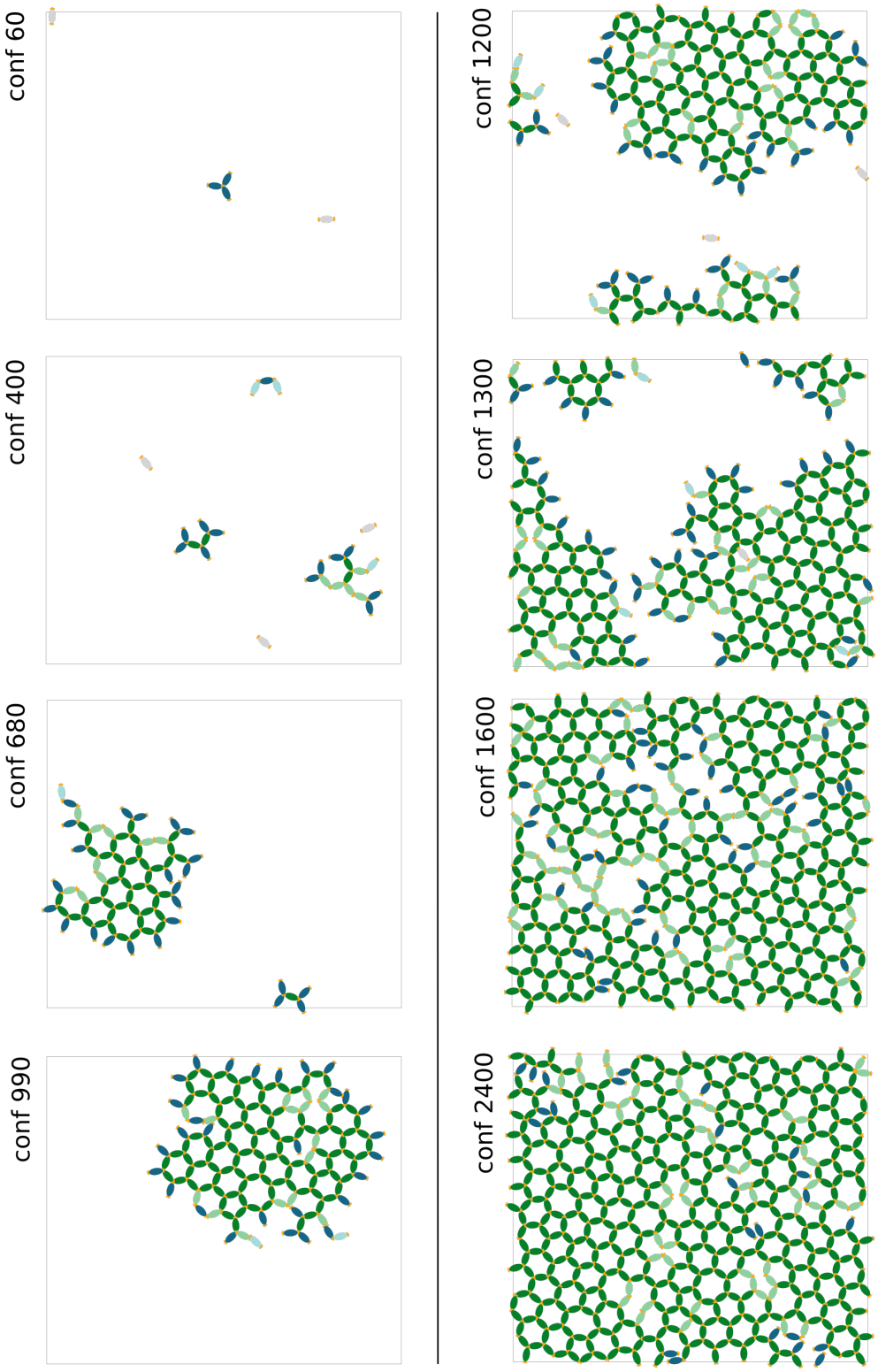


Figure 4.8.: Snapshots for a system at $T^* = 0.13$ and $\mu^* = -6.0$ showing the nucleation process. The colors follow the bonding-energy color scheme defined in fig. 4.1. All snapshots correspond to the marked MC-times in fig. 4.7 (a).

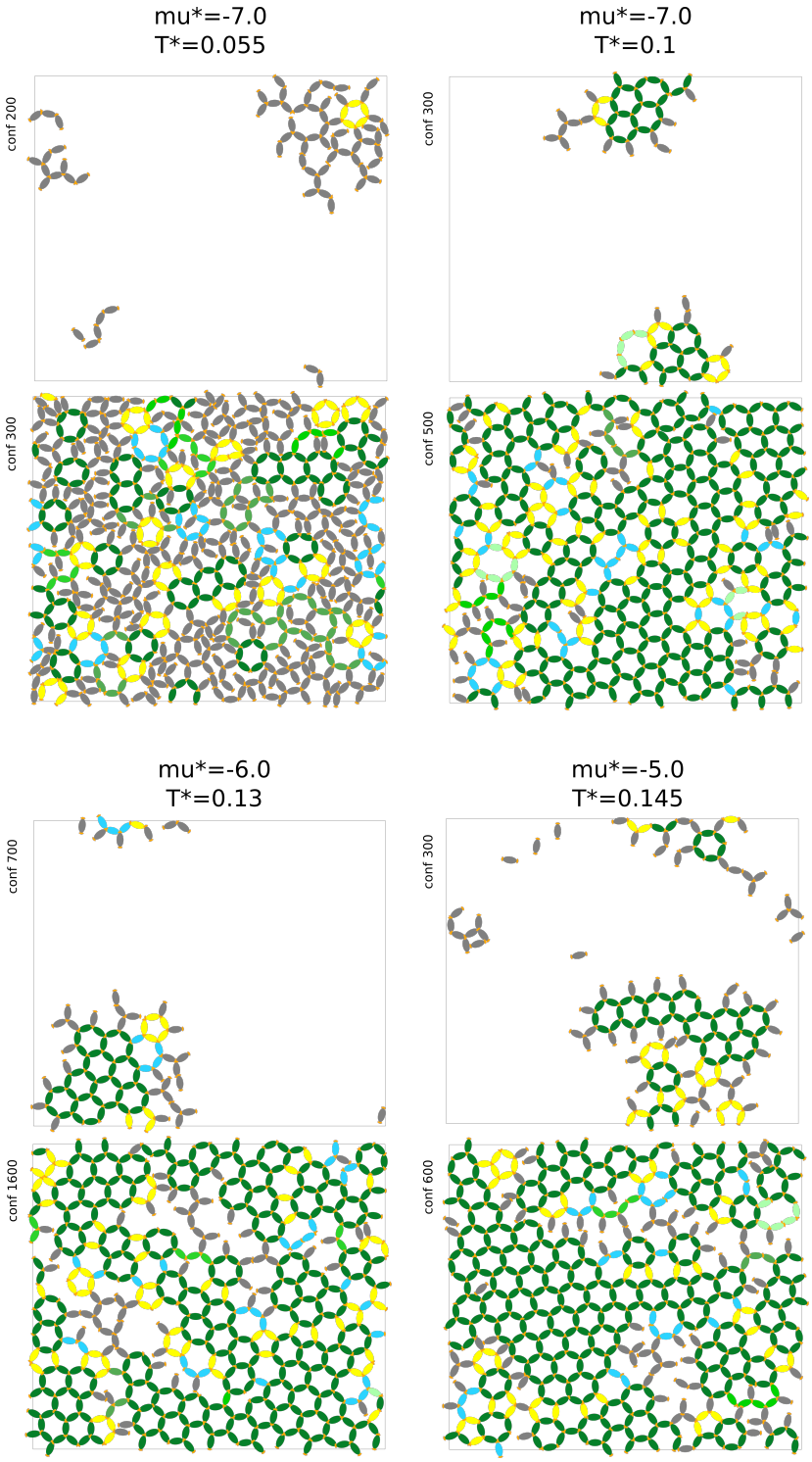


Figure 4.9.: Snapshots of nuclei and the first completely connected configuration at different thermodynamic state-points, highlighting the loops of size six (dark green), five (blue) and seven (yellow). The particles color is chosen using a hierarchical order according to loop size 6 of highest priority followed by loops of size 5, 7, and loops of larger sizes.

4.5. Assembled Porous Networks

Characteristic Lengths. For the interpretation of the radial distribution function (RDF) it is important to know the characteristic distances between particles in the target structure. In a network of loops, the particle positions in each loop form approximately a regular polygon. In a polygon with n edges we have $k_{\max} = \lfloor \frac{k}{2} \rfloor$ such characteristic distances which will be named $l^{(k)}(n)$ of order $k = 1, ..., k_{\max}$. The distance of order $k = 1$ gives the distance to the direct neighbouring particle. Second order with $k = 2$ indicates the distance to the second neighbouring particle and so forth. These characteristic distances $l^{(k)}(n)$, for the k-th distance in a regular n-polygon the following relation has been derived:

$$l^{(k)}(n) = \frac{\sqrt{2}a}{\tan\left(\frac{\pi}{n}\right)} \sqrt{1 - \cos\left(\frac{2\pi k}{n}\right)}, \tag{4.1}$$

where a is the semi-major axis of the ellipse. With the chosen unit of lengths, $2\sqrt{ab} = 1$, this quantity is $a = \frac{\sqrt{\kappa}}{2}$.

In the following analysis of the **radial distribution function**, the characteristic distances of for $n = 6$, i.e. in a loop of size six, are indicated by colored bars, where the width of the bars is chosen as $2\delta_a$, with δ_a being the patch interaction width.
In fig. 4.10 the radial distribution function, RDF is given for thermodynamic states at which the nucleation process is observed. This analysis of the RDF, however, only includes the configurations succeeding the truncation point, and further, each RDF comprises configurations of all 12 independent simulations.

From fig. 4.10 the first three peaks are identified as the characteristic lengths in a loop of six particles. However, the marked regions for the characteristic distances of particles in the hexagon do overlap with the distances in loops of seven or five particles. Comparing the different temperatures, it can be clearly seen that the strongest correlations at the peak positions is observed for $T^* = 0.16$. The correlations at the peaks decrease with lower temperature, as do the long-distance correlations.

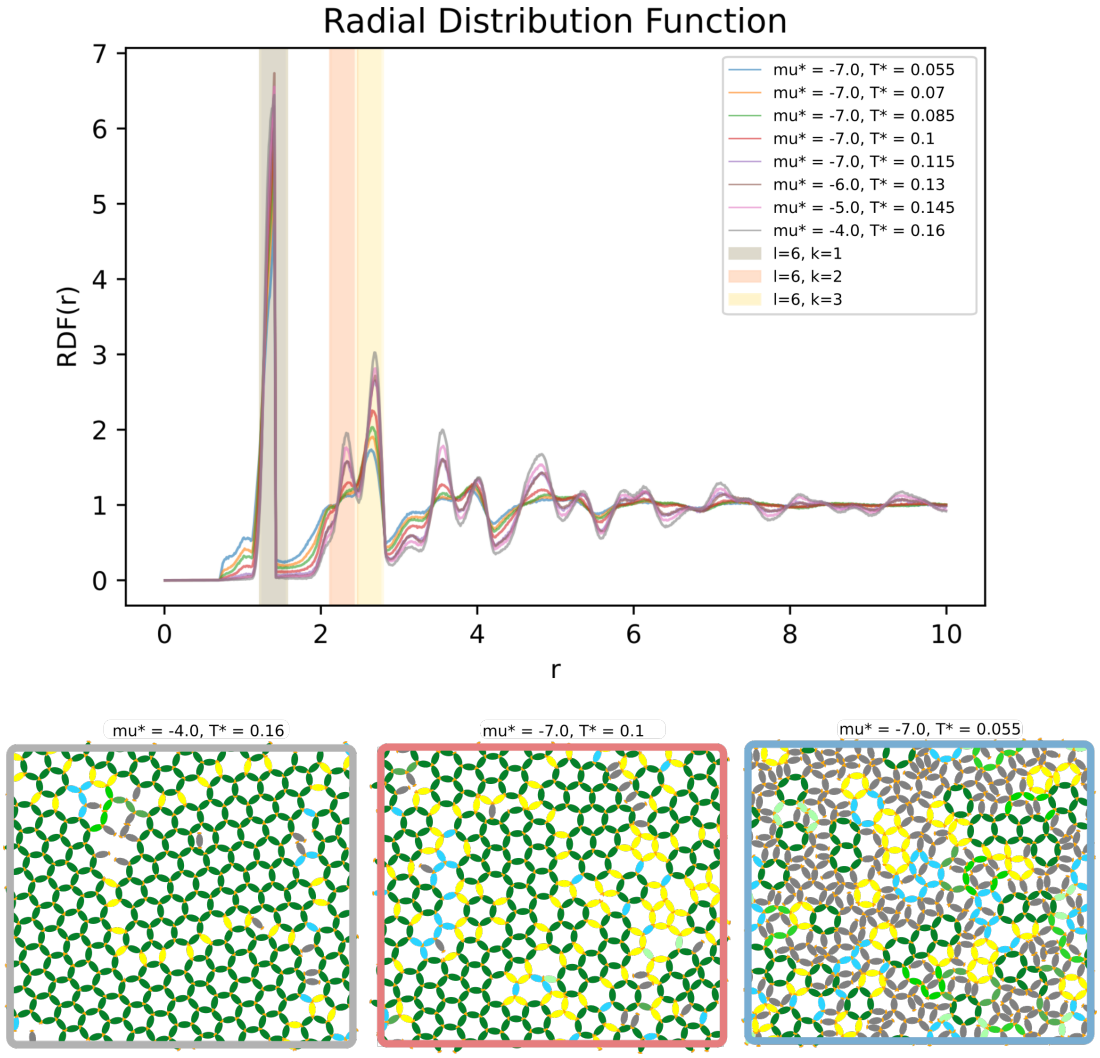


Figure 4.10.: RDF for thermodynamic states μ^*, T^* at which nucleation is observed. Snapshots for three thermodynamic state-points showing the last configurations of the simulation.

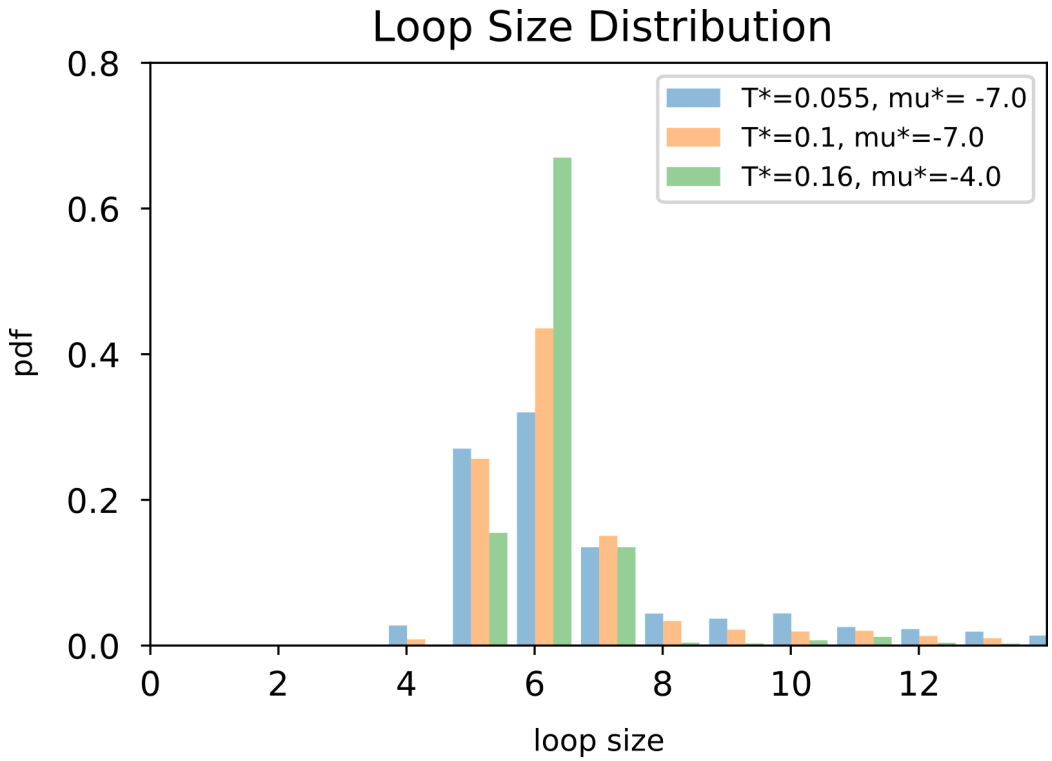


Figure 4.11.: The loop size distribution at different state points (μ^*, T^*) (as labeled) at which nucleation is observed in the simulations. Each distribution represents the average of 12 independent and truncated simulation processes.

The **loop size distributions** for a selection of the state points exhibiting the nucleation process is shown in fig. 4.11. Overall, loops with sizes five, six and seven are most prevalent at all of these state points. It is seen that the highest proportion of loops with size six is obtained at $T^* = 0.16$. As the temperature is decreasing, this proportion drops. Instead, the loops with size five are increasing as the temperatures gets lower. The number of loops of size seven stays relatively constant.

4.6. Self-Healing in Porous Networks

In this section, we follow two simulation trajectories at $T^* = 0.16$. One state point has a negative chemical potential $\mu^* = -2.0$ and the other has a positive chemical potential $\mu^* = 1.0$. At these state points the particles assemble very quickly into a condensed state, with the whole assembly process taking place during the initial and later discarded cycles of the MC-simulation. Already from the beginning of the sampling cycles in the MC simulation, a fully assembled network state is observed. Following the total energy and number of particles along their trajectories (see appendix B.1), it is interesting to see that with the total energy decreases with MC-time, so does the number of particles. This slightly decreasing tendency of the total number of particles with MC-time being observed, indicates that some sort of structural reorganization is going on.
These processes are visualized in fig. 4.12 and fig. 4.13 by showing the loop order parameter Γ_6 (see section 4.3) over MC-time. Respective snapshots for the marked MC-times are shown using the loop color scheme.

In fig. 4.12 we see a porous network evolving with MC-time from a network with substantial loop defects (i.e. loops with size not equal to six) and network defects (i.e., particles with unsaturated bonds). Over the course of the simulation, most of these defects diminish and in the final configuration we see domains of regular porous networks with loops of size six and domain boundaries, with mostly loops of size five and seven. The loop order parameter Γ_6 shows that the rearrangement of particles forming loops of size six is happening in the first half of the simulation run, with the order parameter plateauing at $\Gamma_6 = 0.8$. From here on, the configuration seems to be in a steady state, i.e. no substantial changes in structure or thermodynamic state variables (see trajectories in appendix B.1).

Further fig. 4.13 shows the self-healing process in a more crowded configuration at a positive $\mu^* = 1.0$. From the snapshots, we see that over time the initially disordered and crowded configuration establishes network domains with loops of size six and single particles captured inside the pores. This ordering transition is also seen from the loop order parameter Γ_6, initially at $\Gamma_6 \approx 0.2$ and increasing over MC-time to a final $\Gamma_6 \approx 0.4$. Within this simulation run, judging from the loop order parameter, no steady state was reached. It could be hypothesized that the target structure, which is not yet reached, is a network with loops of size six and single particles occupying the space within the pores.

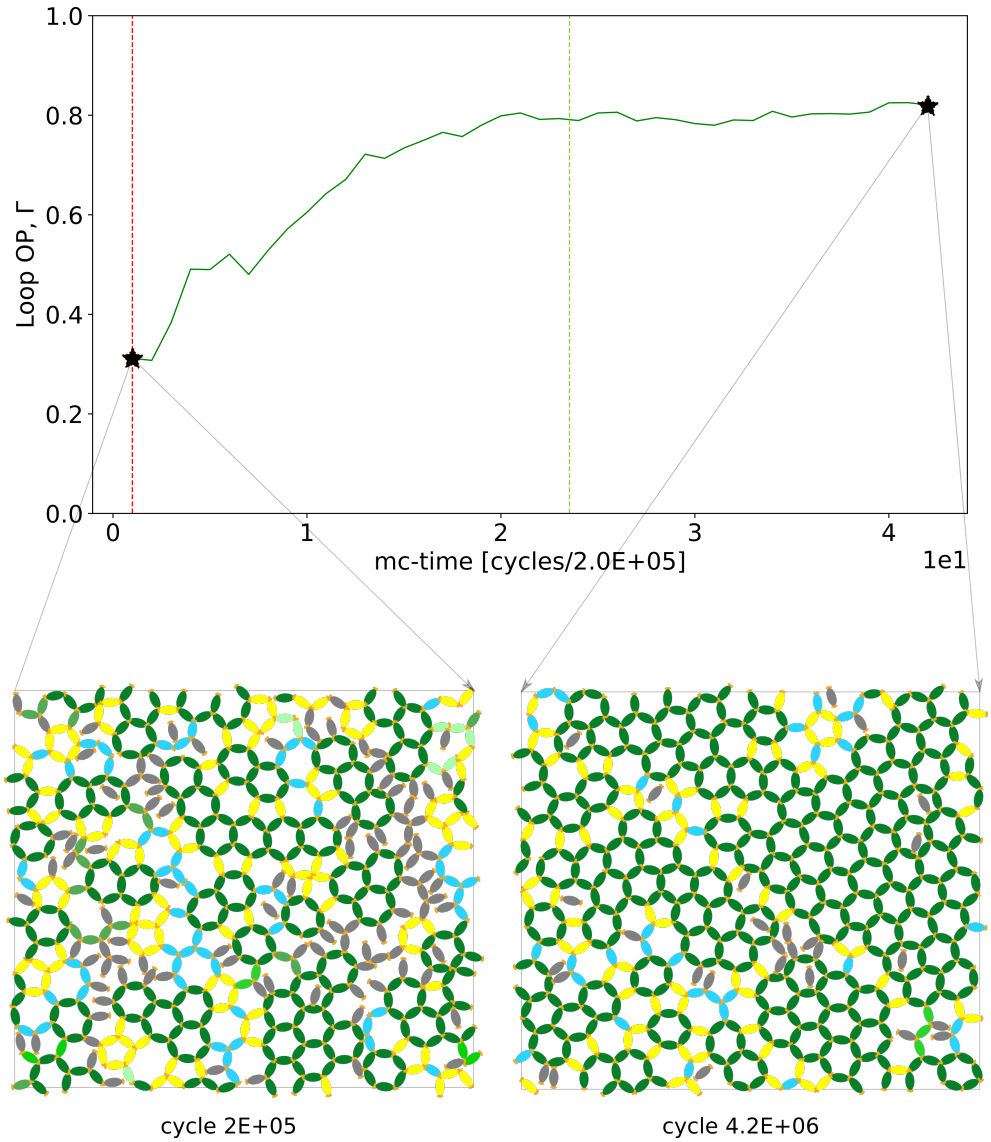


Figure 4.12.: For a single simulation process at $T^* = 0.16$ and $\mu^* = -2.0$ the top panel shows the loop order parameter Γ_6 as a function of the MC-time, while the bottom panels show snapshots at the indicated MC-times. Loops of different sizes (dark green - size 6, blue - size 7, yellow - size 5) are indicated in the snapshots. The sampling starts at the red dashed line. The truncation point is indicated by the green dashed line.

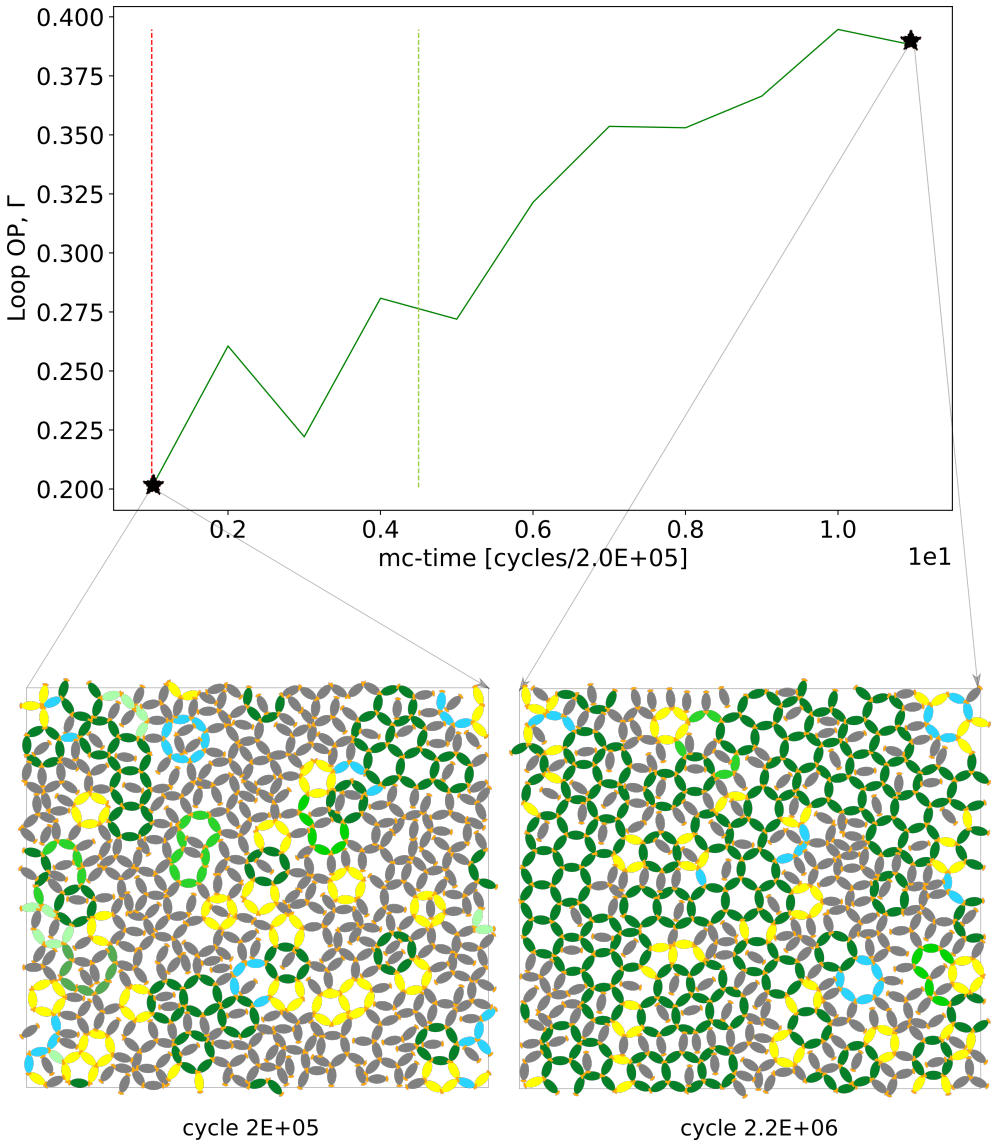


Figure 4.13.: For a single simulation process at $T^* = 0.16$ and $\mu^* = 1.0$ the top panel shows the loop order parameter as function of the MC-time, while the bottom panels show snapshots at the indicated MC-times. Loops of different sizes (dark green - size 6, blue - size 7, yellow - size 5) are indicated in the snapshots. The sampling starts at the red dashed line. The truncation point is indicated by the green dashed line.

5. Results on Self Assembly of Patchy Ellipses with $\theta = 15°$

In the following chapter, the self-assembly behaviour of patchy elliptic particles with a patch opening angle of $\theta_a = 15°$ forming particle chains is examined.
In section 5.1 an overview of assembled structures in the simulations is given. Further, section 5.2 discusses the computed ensemble averages for state variables such as potential energy, number of particles and energy per particle.
In a further analysis, the assembly processes of chains going from negative to positive chemical potentials μ^* are discussed for two temperatures $T^* = 0.07$ and $T^* = 0.115$ separately. These two temperatures are chosen as examples for two temperature regimes where different assembly processes are observed, one at low temperatures $T^* < 0.1$ and one at high temperatures $T^* > 0.1$. The low-temperatures assembly of chains follows a continual growth process of the chain in an otherwise empty simulation box. This type of assembly is looked at in more detail in section 5.3. At higher temperatures, e.g., $T^* = 0.115$, the assembly of chains is happening from a particle gas, i.e., freely moving single particles eventually binding with other particles and forming chains. More details on the assembly process at this temperature is given in section 5.4.

5.1. Snapshots & Simulation Trajectories

The snapshots in fig. 5.1 shown for a selection of thermodynamic state points provide evidence of the rich assembly behaviour of these particles. We distinguish a regime of sparse chains, at $\mu^* - 6.0$ and $T^* = 0.055$ and a regime of packed chains, e.g., at state points (μ^*, T^*) of $(-1.0, 0.055)$ and $(3.0, 0.1)$.
Further we identify disordered, fluid configurations, occurring for example at $T^* = 0.16$ and at $T^* = 0.1$ for $\mu^* = -6.0$ and -1.0. The fluid phases show almost empty configurations for the lowest chemical potentials. At the chemical potential of value $\mu^* = -1.0$, a fluid phase hosting single particles as well as clustered particles forming chains a few particles long, is observed. At the highest temperature $T^* = 0.16$, the observed chains have a considerably shorter length, with the most prevalent chain length being two particles.

Simulation Trajectories. A complete record of simulation trajectories according to potential energy, number of particles and energy per particle is summarized in B.1.

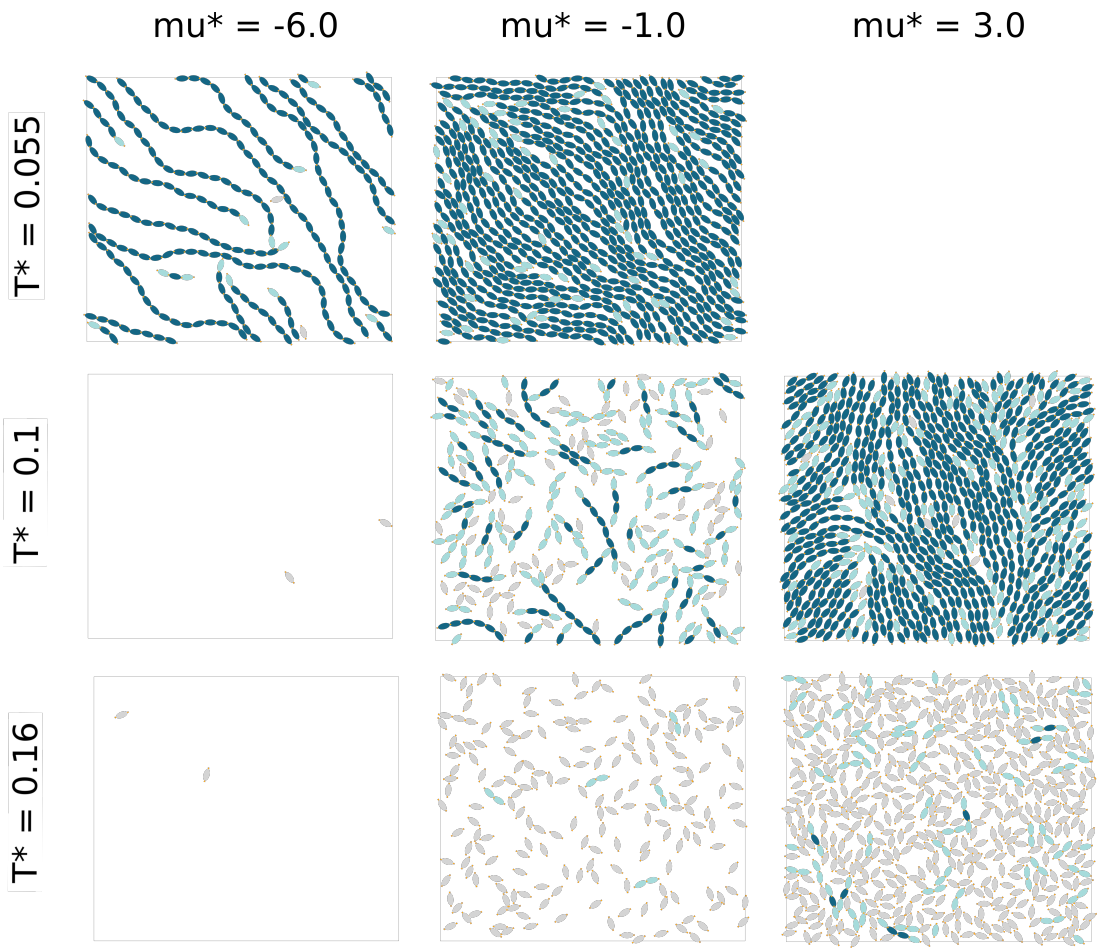


Figure 5.1.: Snapshots at three chemical potentials ($\mu^* = -6.0$, -1.0 and 3.0) and three temperatures ($T^* = 0.055$, 0.1 and 0.16) with particles colored according to their bonding energy - no bonds (grey), one bond (light blue) and two bonds (dark blue). For the state point $\mu^* = 3.0$ and $T^* = 0.055$ no simulation results were obtained due to exceedingly long simulation times.

5.2. Ensemble Averages - State Variables

Here, an overview of all thermodynamic state points included in this study is given. For this overview, the ensemble averages of state variables such as the potential energy, E_{pot}, number of particles N and energy per particle E_{pot}/N are computed over all 12 independent trajectories. The results for these ensemble averages can be seen in fig. 5.2 to fig. 5.4.

The potential energy E_{pot}, presented in fig. 5.2, shows a monotonically increasing behaviour as the temperature grows for each of the chemical potentials. Following a single curve of constant μ^*, a transition from a condensed phase at low energies to a phase with $E_{\text{pot}} \approx 0$ can be seen. A similar transition is seen for the isotherm $T^* = 0.07$ of E_{pot} versus the chemical potential μ^*. The decreasing behaviour of E_{pot} with growing μ^* is following a steep curve, while for an isotherm $T^* = 0.16$, the decrease in energy follows a much flatter curve.

Figure 5.3 shows the total number of particles for the respective state-points. Empty states with $N \approx 0$ can be identified at $\mu^* = -6.0, -5.0$ and -4.0. For each isotherm, a transition occurs from a state with negative chemical potential, characterized by a low number of particles, to a phase with positive chemical potential, characterized by a high number of particles.

The energies per particle shown in fig. 5.4 display a qualitatively very similar behaviour as the potential energy E_{pot} in fig. 5.2. From viewing the isotherms of the energy per particle, it can be seen that the transition from negative to positive chemical potentials goes along with the particles bonding. The bonding probability is higher for lower temperatures, with the lowest temperature $T^* = 0.055$ almost achieving the theoretical minimum of $E_{\text{pot}}/N = -1.0$ at the chemical potential $\mu^* = -6.0$. For the isotherm of $T^* = 0.055$, the energy per particle slightly increases with increasing chemical potential μ^* at $\mu^* > -4$. This behaviour is also observed at isotherms $T^* = 0.07$ and 0.085. However, results at these state points must be treated carefully since, for these low temperatures, the simulations did not reach a steady state.

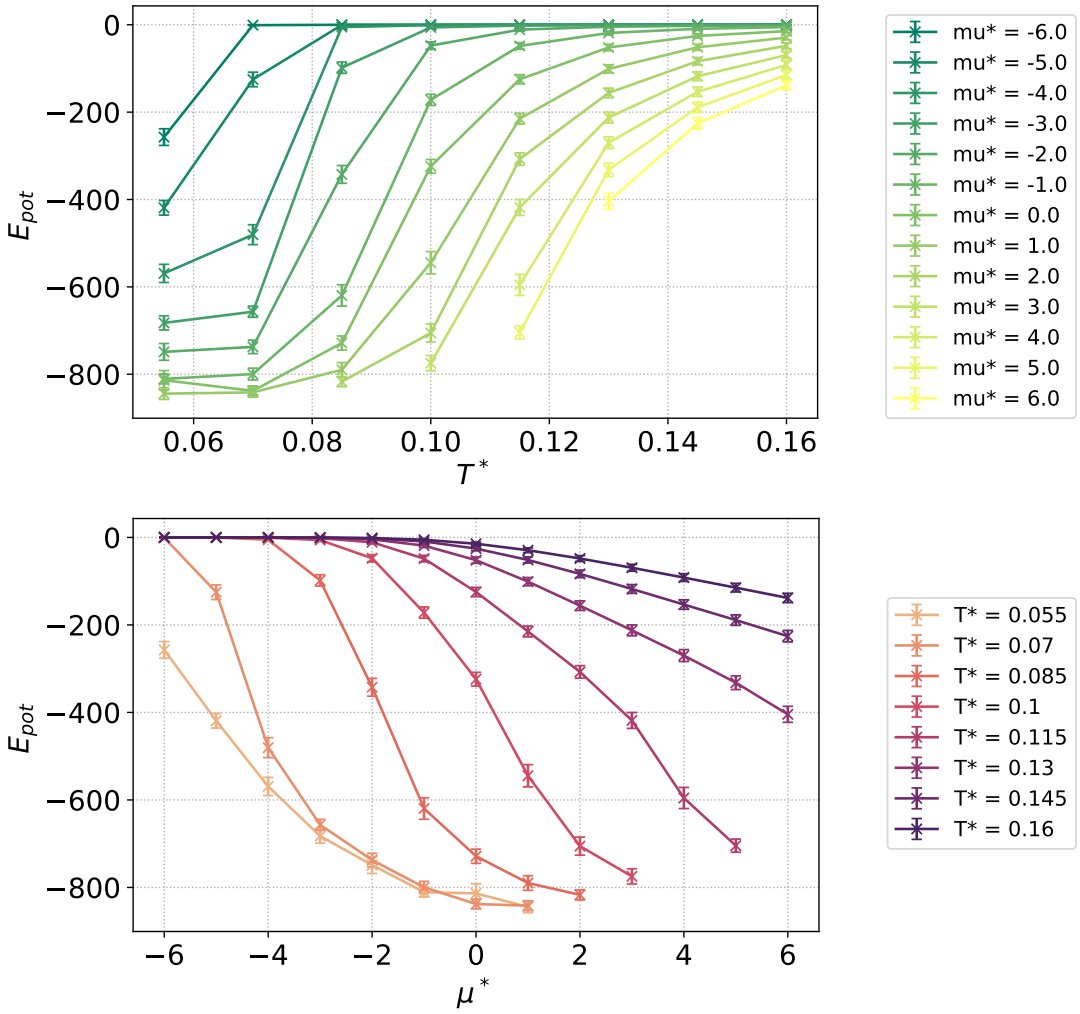


Figure 5.2.: Total energy E_{pot} in units of $k_B T$ for constant μ^* across temperatures T^* (top) and for constant T^* across chemical potentials μ^* (bottom).

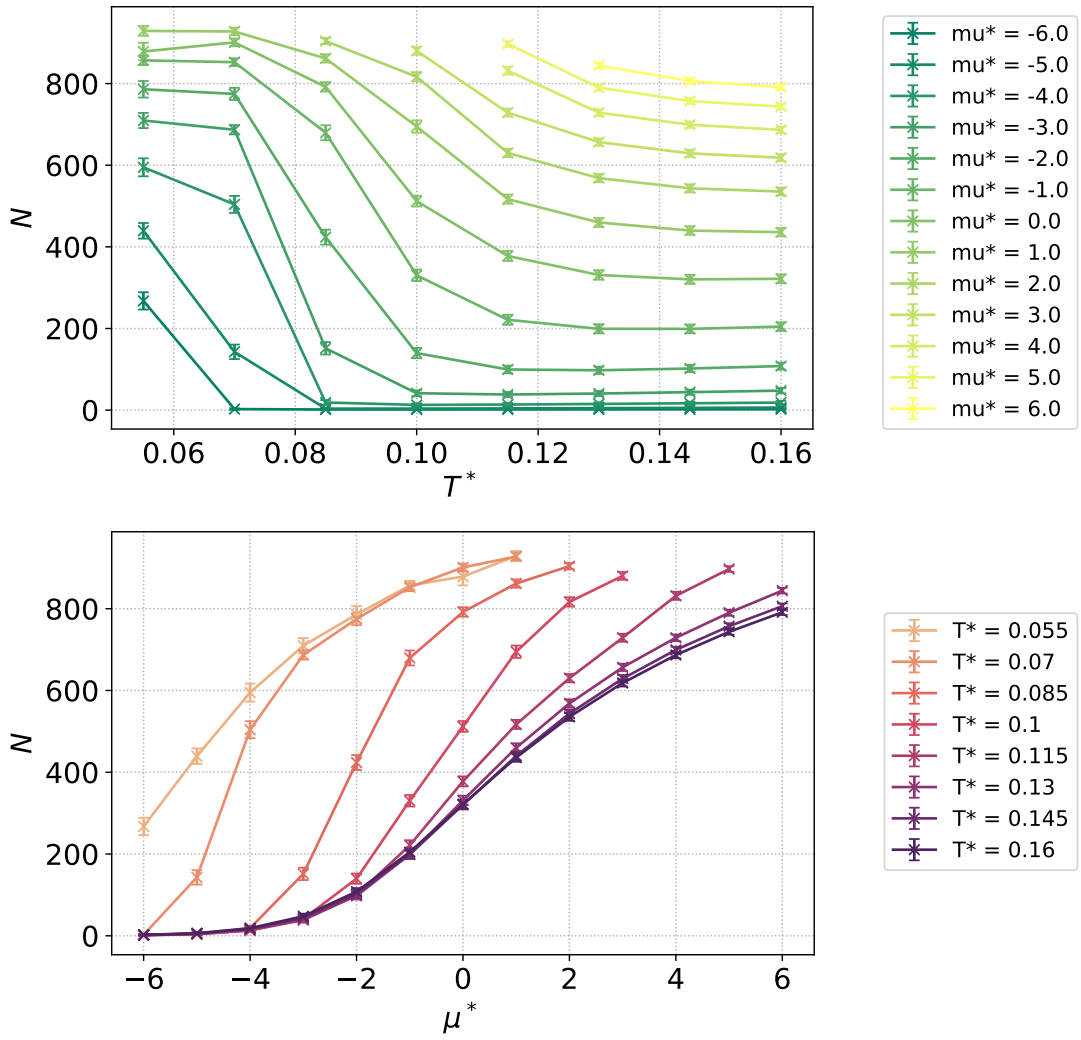


Figure 5.3.: Total number of particles N for constant μ^* across temperatures T^* (top) and for constant T^* across chemical potentials μ^* (bottom).

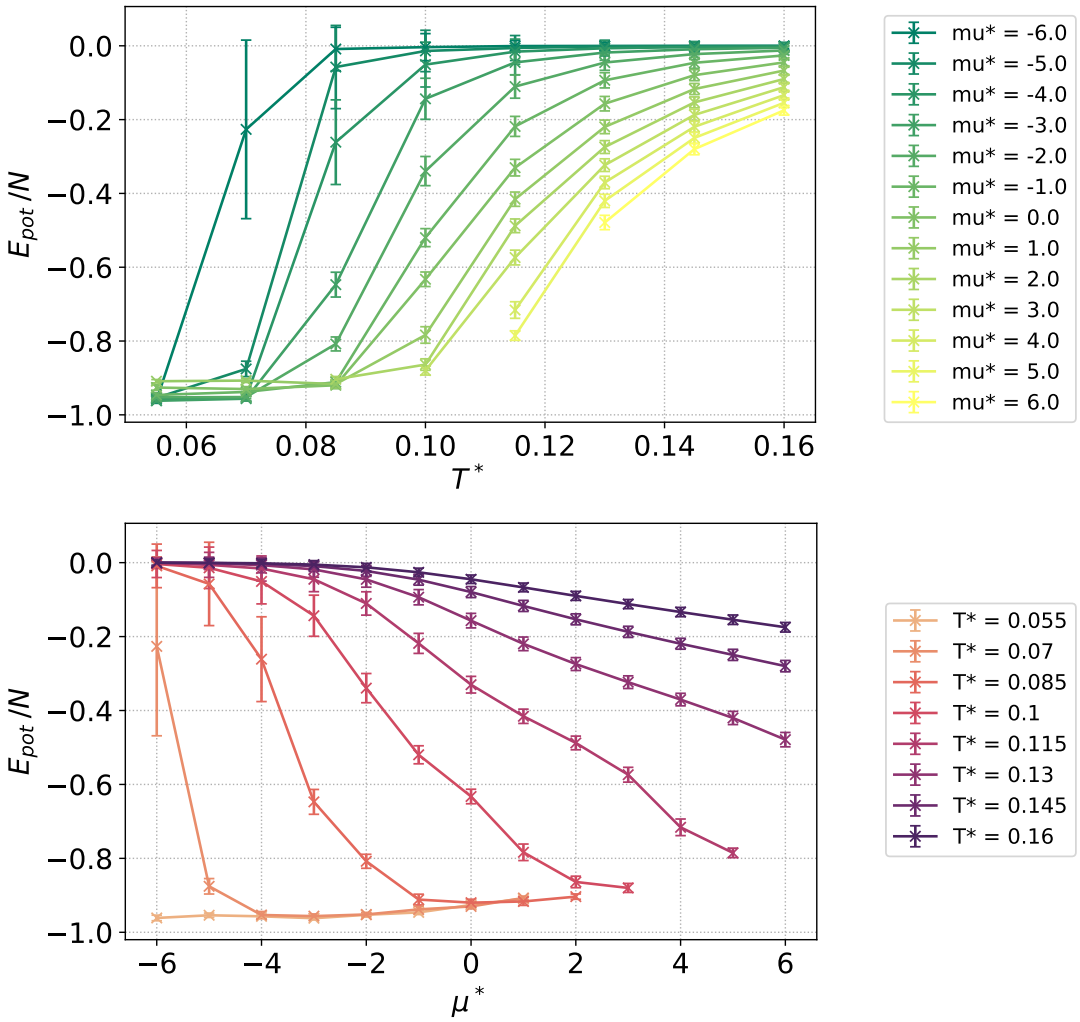


Figure 5.4.: Energy per particle E_{pot}/N in units of k_BT for constant μ^* across temperatures T^* (top) and for constant T^* across chemical potentials μ^* (bottom).

5.3. Chain Formation at Low Temperature - $T^* = 0.07$

Over the course of simulation runs at low temperatures ($T^* = 0.055$ and 0.07) as well as for negative chemical potentials $\mu^* < 0$, very slow self-assembly processes of chains can be observed. Due to this slow MC dynamics at these state points in our simulations, we did not reach a steady-state and hence cannot make conclusive predictions on the target states at each chemical potential. However, we assume that at low enough temperatures, because of the high bonding energy, the particles will mainly appear in chains and the chemical potential will dictate the overall density, i.e., the number of particles in the box.
In fig. 5.5 the snapshots at different stages of this assembly process in one such simulation run at $T^* = 0.07$ and $\mu^* = -4.0$ are shown. The assembly process at this particular state-point is observed being realized in two stages: first, the assembly of particles into chains, and second, the alignment of those chains such that a higher overall density is reached.

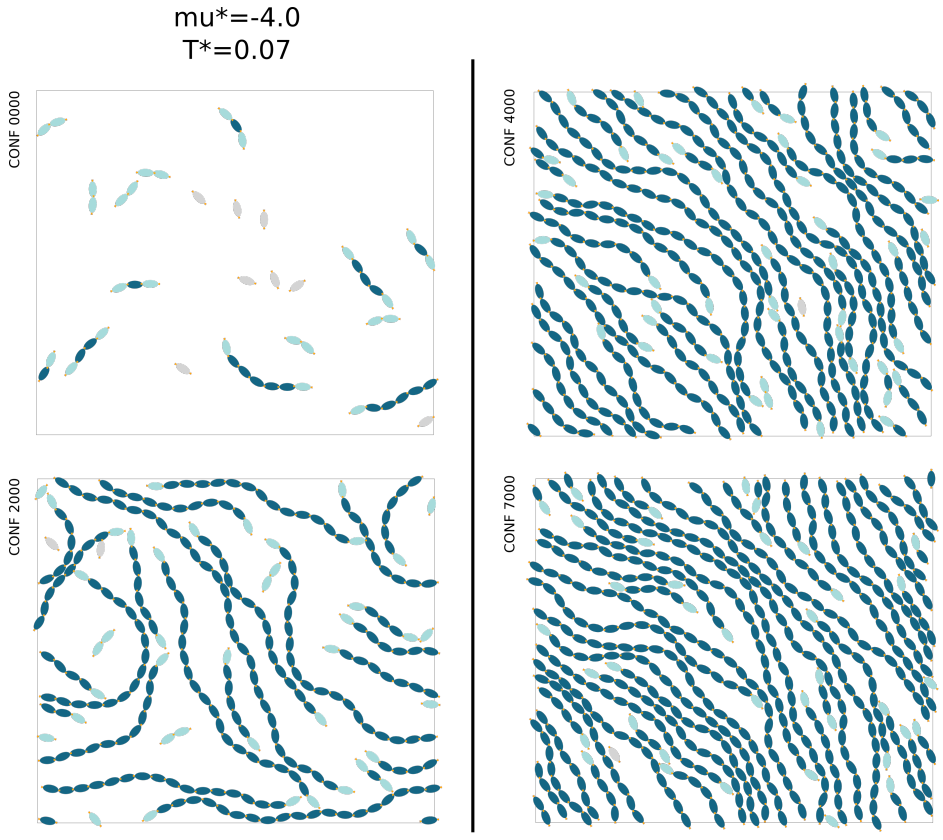


Figure 5.5.: The self assembly process for one simulation run at $^* = 0.07$ and $\mu^* = -4.0$ over the course of MC-time, with two configurations being separated by 2.0E+03 MC-cycles.

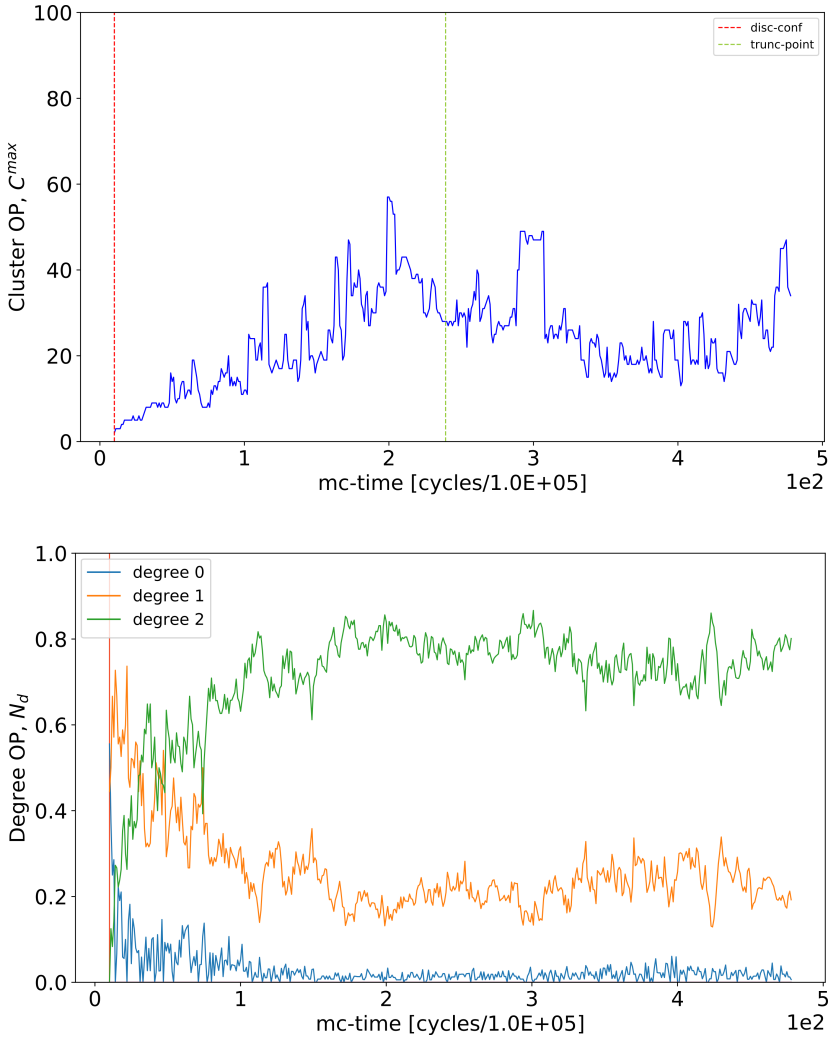


Figure 5.6.: The largest cluster size $C^{\max}$ (top) and the degree order parameter N_{d} (bottom) for one simulation run at $^* = 0.07$ and $\mu^* = -4.0$ over MC-time for the self assembly process shown in fig. 5.5.

The **order parameters** $C^{\mathbf{max}}$, i.e. the largest cluster size and the **proportion of particles** N_d **with degree** d for the same simulation run as shown in fig. 5.5 at $T^* = 0.07$ and $\mu^* = -4.0$ is depicted in fig. 5.6. The slow growth of the chains can be observed during the first half of the simulation run. A maximum chain length of $C^{\max} > 30$ is most likely extending over the periodic boundary conditions. In the same panel the degree order parameter N_d for $d = 0, 1$ and 2 bonds per particle is showing, that the proportion of particles within chains, i.e. $d = 2$, is growing continually. The proportion of single particles ($d = 0$) is almost zero while the proportion of particles at the end of chains ($d = 1$) sinks as the chains grow in size.

Further, the **nematic order parameter** was determined for all simulation runs at state points with temperature $T^* = 0.07$ and at $\mu^* = -4.0$ and $\mu^* = 0.0$. In fig. 5.7 the nematic order parameter for these two states is plotted for each simulation trajectory individually at different MC-times. While the simulation run shown in fig. 5.5 reaches a global nematic order parameter of $\Lambda \approx 0.8$, other simulation runs achieve considerably lower global nematic order. This is understood to be due to different domains of aligned chains with different directors arising in most of the simulations, such as shown in fig. 5.7.

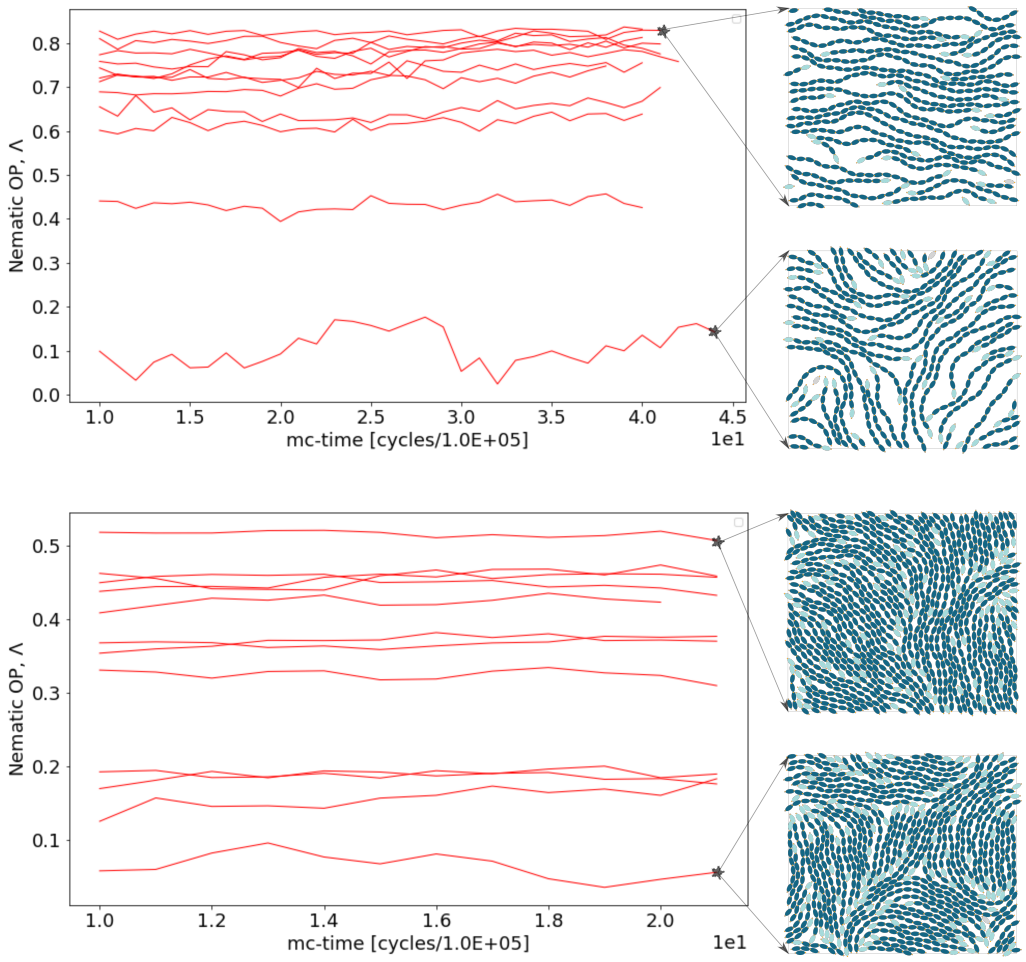


Figure 5.7.: Nematic order parameter with MC-time for twelve independent simulation runs at thermodynamic state points ($\mu^* = -4.0, T^* = 0.07$ (top) and ($\mu^* = 0.0, T^* = 0.07$ (bottom). Snapshots of last configurations at maximum and minimum global nematic order parameter at each state-point shown.

In the **radial distribution function (RDF)**, shown in fig. 5.8 of different thermodynamic states with $T^* = 0.07$, high correlations occur at integer multiples of the distance corresponding to the major axis of the ellipse, $2a = \sqrt{2}$. These correlations occur due to chain formation. Also, a bump in the RDF at a distance just above the distance for the length of the minor axis $2b$ is observed, growing in size as the chemical potential increases. This marks the decreasing distance between chains as the overall density in the simulation box grows.

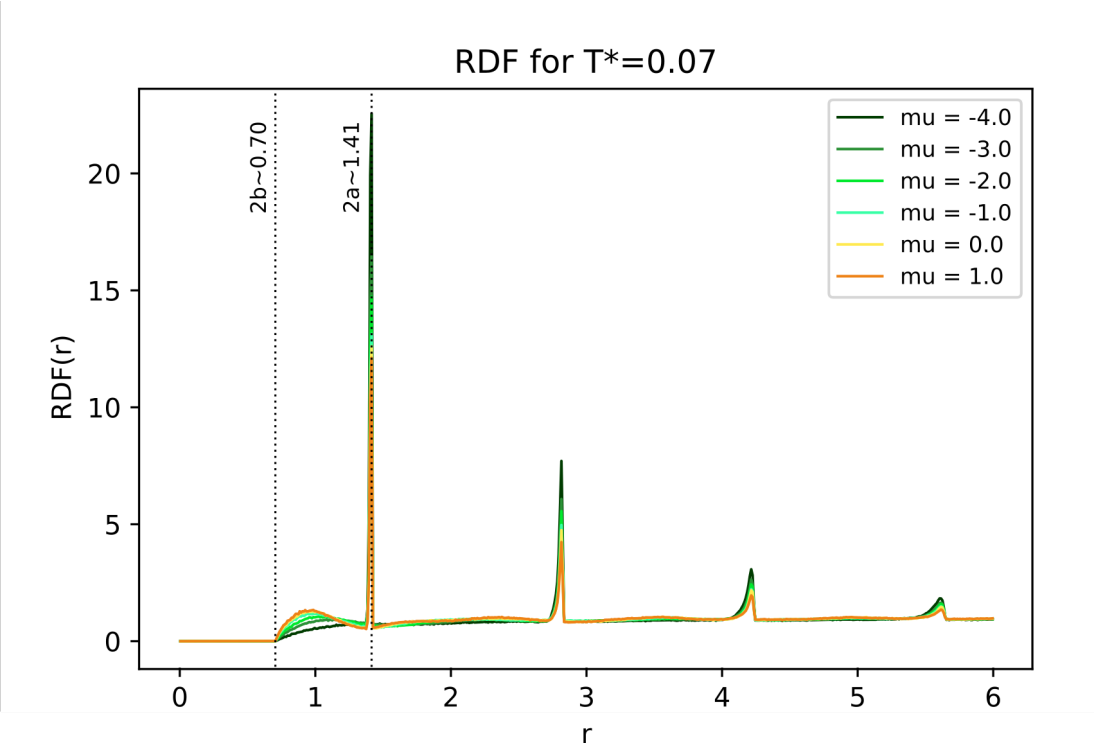


Figure 5.8.: RDF for different chemical potentials μ^* at temperature $T^* = 0.07$. High correlation occur at integer multiples of the distance corresponding to the major axis of the ellipse $2a = \sqrt{2}$ due to chain formation.

In fig. 5.9 the **chain length distribution** at $T^* = 0.07$ is shown for three different chemical potentials. At the chemical potential of negative value $\mu^* = -5.0$ an exponential decay is observed - a particle without bonds, i.e. a cluster size equal to one, shows the highest probability, while the probability of clusters, i.e. chains, of larger sizes follows an exponential distribution, as can be determined from the log-plot (see fig. 5.10). For chemical potentials $\mu^* = 0.0$ and 1.0 the shape of the distribution changes. A typical chain length of eight and ten particles for the chemical potentials $\mu^* = 0.0$ and 1.0 can be determined respectively. The typical chain length is increasing as the chemical potential is growing.
An analysis of the tails of these distributions as well as of the maximum lengths at different state points is difficult to describe and therefore not presented here. This is because the chains extend over the periodic boundaries of the simulation box because at low temperatures $T^* < 0.1$.

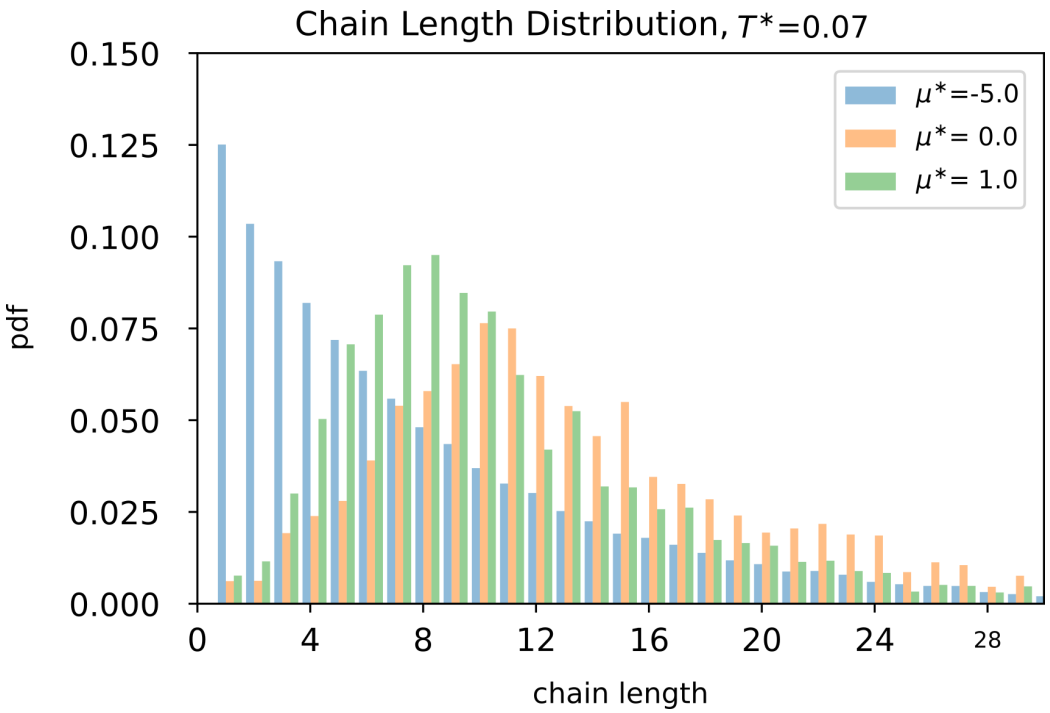


Figure 5.9.: Chain length distribution at $T^* = 0.07$ for three different chemical potentials μ^* on a linear scale. Each distribution represents the average of 12 independent and truncated simulation processes.

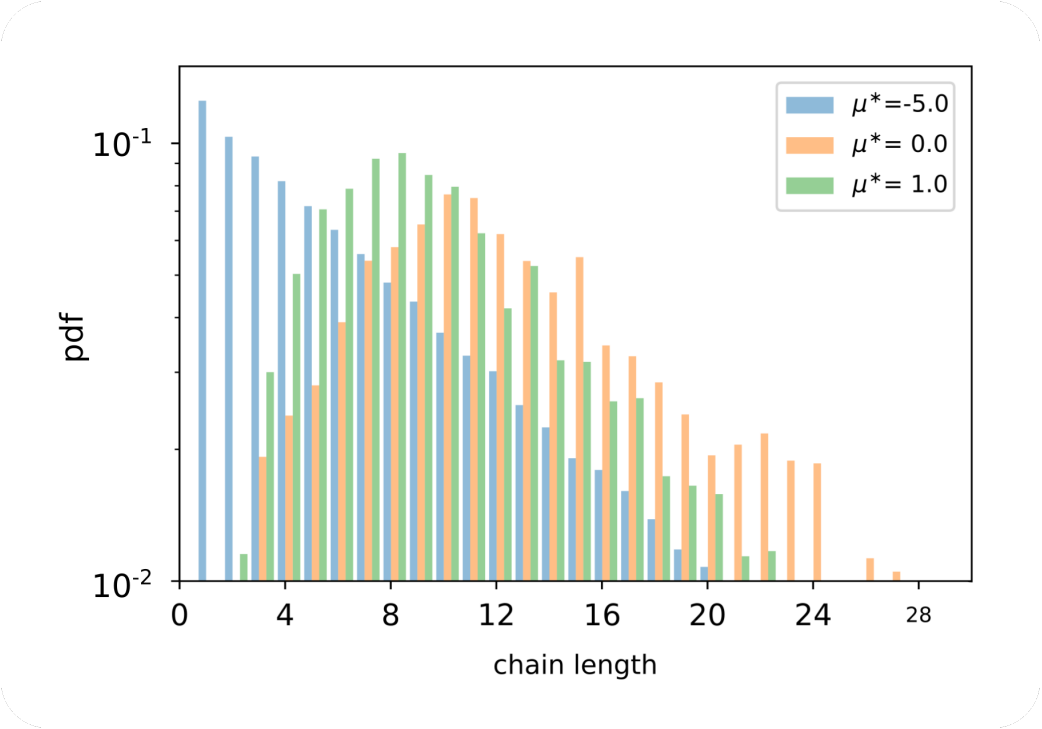


Figure 5.10.: Chain length distribution at $T^* = 0.07$ for three different chemical potentials μ^* on a log-scale. Each distribution represents the average of 12 independent and truncated simulation processes.

5.4. Chain Formation at High Temperature - $T^* = 0.115$

At a higher temperature, $T^* > 0.1$ and specifically for $T^* = 0.115$, the assembly of chains at lower chemical potentials μ^* starts from a particle gas, i.e., freely moving single particles eventually binding with other particles and forming chains. At higher chemical potentials, these chains undergo a second transition, from freely moving chains at lower μ^* to domains of packed chains at higher chemical potentials. The orientation and distribution of these domains are still changing over the course of the MC-time, as can be seen in fig. 5.11.

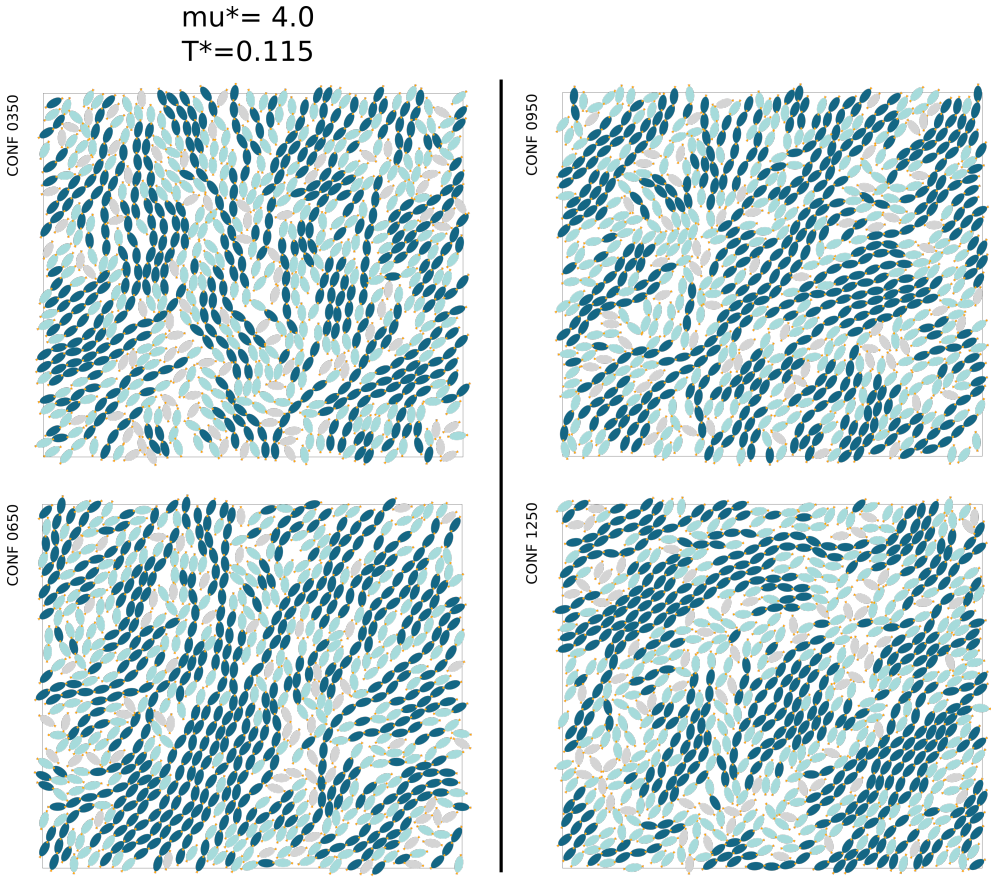


Figure 5.11.: Self assembled states at $T^* = 0.115$ and $\mu^* = 4.0$.

The overall chain-formation behaviour can be captured by determining the size of the longest chain, given by the **order parameter** $C^{\mathbf{max}}$, at each thermodynamic state point, as shown in fig. 5.12. This analysis shows that the chain length is critically dependent on the temperature T^* with lower temperatures stabilizing longer chains and higher temperatures stabilizing shorter chains. Also, the rate of growth of the longest chain with increasing chemical potential μ^* is larger for higher temperatures.

Figure 5.13 shows **chain length distributions** at temperature $T^* = 0.115$ and for three chemical potentials, $\mu^* = -5.0$, 0.0 and 4.0. At the chemical potential $\mu^* = -5.0$, almost no chain formation is visible, only a few particle pairs, i.e. chains of length two, form. For higher chemical potentials, exponential distributions over chain lengths are observed, however, still with single particles most common. With increasing chemical potential, also the maximum chain length is increasing.

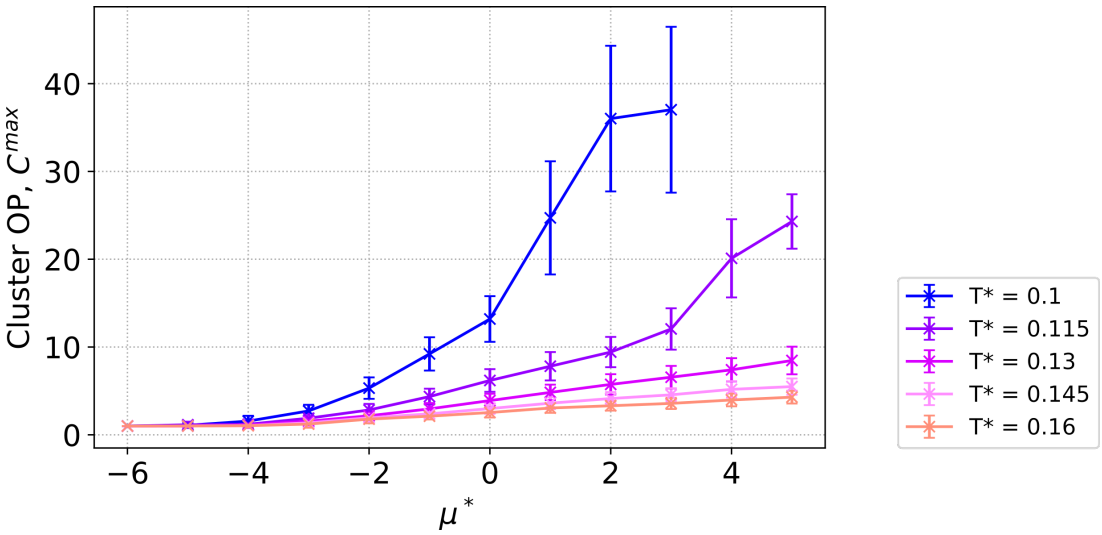


Figure 5.12.: Largest cluster size $C^{\max}$ for high temperatures $T^* > 0.1$.

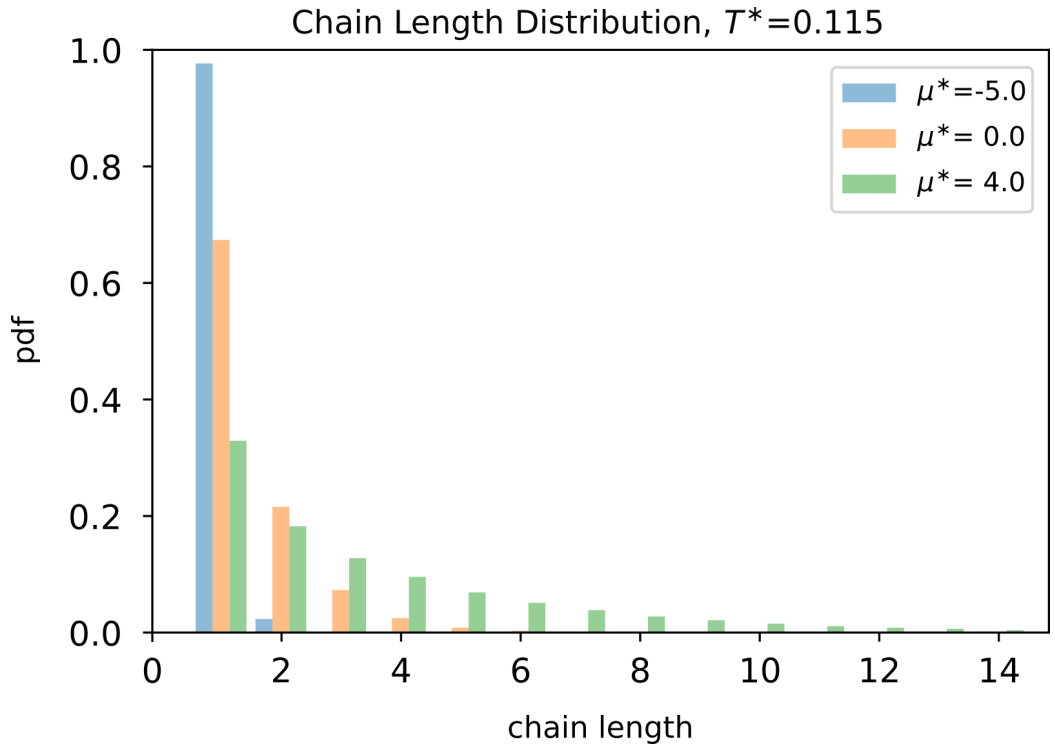


Figure 5.13.: Chain length distribution at $T^* = 0.115$ for three chemical potentials $\mu^* = -5.0, 0.0$ and 4.0. Each distribution represents the average of 12 independent and truncated simulation processes.

6. Summary & Outlook

In the preceding chapters, the self-assembly properties of patchy ellipses with two opposing patches of Kern-Frenkel type located on the vertices of the hard core ellipse of aspect ratio $\kappa = 2$ have been studied via MC-simulations in the grand-canonical ensemble. The self-assembly of two types of these patchy ellipses, differing in the patch opening angle, have been studied, where strikingly different types of assemblies are observed.

For patchy ellipses with a patch opening angle $\theta_a = 45°$ allowing for up to two bonds per patch, the particles assemble mainly into porous networks. At positive chemical potentials, crowded and disordered states are observed, but also crowded states with a network structure and single particles occupying the pores in the network. The structural properties of these networks have been studied using a newly proposed loop order parameter, defined specifically for this purpose. Also, dynamic properties such as the assembly process into these networks, as well as self-healing properties of assembled lattices with defects were highlighted. The networks were shown to assemble via a nucleation process at negative chemical potentials. Further, self-healing properties of the lattices at high temperatures were shown to exist. At the highest temperature, $T^* = 0.16$, the assemblies were observed to yield the best lattices, in the sense that they have the least defects. We believe that this is the case due to the fact that the bonding energy is lower compared to temperatures strong and rearrangement of particles is possible because the breaking of bonds once formed is more likely. This way the dynamic pathway to the energetic ground state is not hindered.

Over the investigated thermodynamic state points it was shown that patchy ellipses with $\theta_a = 45°$ assemble into porous networks with a loop-size distribution centering around size six and depending on the specific thermodynamic state. The target structure is believed to be a network of purely size six loops for certain negative chemical potentials. However, at much lower or much higher chemical potentials, it is believed that other target structures could be uncovered.
We believe that as one goes to very low chemical potentials, networks with very low particle density could emerge. These networks could either be of an irregular structure or of a regular one, with a network based on the truncated square tiling as the only candidate structure. This network type exhibits a lower number density than the network based on the hexagonal tiling and still fulfills the condition

of two bonds per patch, meaning it is an energetic ground state.
The theoretical limit at very high chemical potentials, when the minimization of the grand-canonical potential is dominated by the maximization of the number of particles, is the closest packed lattice configuration. As proposed in part II two entropically favoured lattice types exist, the parallel and the diagonal lattice. As ellipses with patch-bonds are hindered in their rotational degree of freedom, it is still unclear which of the two lattice types would be entropically favoured. For lower but still high chemical potentials, we can only speculate on the emerging phases, which could be network structures with the pores completely filled, or even stacked chains with dangling particles at each joint.

For patchy ellipses with a patch opening angle allowing for only one bond per patch, the formation of chains was observed. Overall thermodynamic properties have been reported, and the assembly process of such chains at high and low temperatures was demonstrated. An in-depth qualitative and quantitative analysis is pending. Parts of the analysis presented here comprised the chain-length distributions as well as insight into the largest cluster analysis for high temperatures ($T^* > 0.1$). At low temperatures, the radial distribution function revealed very located correlations, reflecting the inter-particle distances within chains. For high temperatures, the analysis of the radial distribution function is pending and hence no quantitative statements on the structural order is possible yet.

For the system of $\theta_a = 15°$ patchy particles, numerous questions are still open for further analysis and discussion. The assembly of chains at high chemical potentials could not yet be observed in the present simulations due to the computational time limit. A different MC design should, however, make it possible to achieve at least a small number of configurations at these thermodynamic state points. In order to gain a good overall understanding of this system at different thermodynamic state points, analysis of high-density states is needed.

Concluding, by two examples of the patchy ellipse model, a very versatile phase behaviour of the patchy ellipse was shown. The phase behaviour of other types of patchy ellipses, with different geometric patch arrangements, interactions and also ellipse aspect ratios, will most likely uncover an even richer phase behaviour. From showing that patchy ellipses with $\theta_a = 45°$ and aspect ratio $\kappa = 2$ prefer open lattice configurations, we assume that these types of patchy ellipses, including ellipses with higher aspect ratios, are candidate models for versatile open networks with tuneable pore-sizes.

A. Appendix - Lattices of Hard Ellipses

A.1. Derivation of Free energy term A_0

In the Einstein Molecule method, the reference system is the Einstein Molecule, defined in section 4.4.1 with free energy A_0, which will be named A_{mol-id} here. A derivation of this quantity is provided here for a two dimensional Einstein Molecule with in-plane rotations of particles. The derivation was done following [22].

For the ideal Einstein Molecule the free energy of the system is given by:

$$\frac{A_{mol-id}}{Nk_BT} = -\frac{1}{N}\ln\left(Z_{mol-id}\right) \tag{A.1}$$

with the canonical partition function Z_{mol-id} defined via the Hamiltonian of the Einstein molecule H_{mol-id}:

$$H_{mol-id} = H_{mol-id}(\vec{p}^N, \vec{r}^N, \vec{\psi}^N) = K_{mol-id}(\vec{p}^N) + V_{mol-id}(\vec{r}^N, \vec{\psi}^N) \tag{A.2}$$

$$K_{mol-id}(\vec{p}^N) = \sum_{i=1}^{N} \frac{\vec{p}_i^2}{2m} \tag{A.3}$$

$$V_{mol-id}(\vec{r}^N, \vec{\psi}^N) = \Lambda_{trans}\sum_{i=2}^{N}(\vec{r}_i - \vec{r}_{i,0})^2 + \Lambda_{rot}\sum_{i=1}^{N}(mod(\Psi_i - \Psi_{i,0}, \pi))^2 \tag{A.4}$$

From the Hamiltonian it can be seen, that the Einstein molecule is the Einstein crystal with one particle acting as a carrier of the lattice. Therefor the carrier particle (particle with index 1) is not allowed to vibrate around it's lattice position, but still it is allowed to move freely through the simulation box.
For the canonical partition function due to distinguishable particles we do not need the factor $\frac{1}{N!}$. In the Einstein molecule we choose one particle (with index 1) to be the carrier of the ideal lattice. Because we could choose any of the N particles as the carrier we have to to account for our specific choice with a factor

$\frac{1}{N}$. Therefore, the canonical partition function gives:

$$Z_{mol-id} = \frac{1}{h^{DN}}\Big[\frac{1}{N}\frac{1}{(2\pi)^N}\int\int\int d\vec{p}^N d\vec{r}^N d\vec{\psi}^N \exp\Big(-\beta H_{mol-id}\Big)\Big] \tag{A.5}$$

$$= \frac{1}{N}\frac{1}{\Lambda_c^{D_t N}}\int d\vec{r}_1 \int d\vec{r}_2 ... d\vec{r}_N \exp\Big(-\beta\Lambda_{trans}\sum_{i=2}^{N}(\vec{r}_i - \vec{r}_{i,0})^2\Big)\times \tag{A.6}$$

$$\times\frac{1}{(2\pi)^N}\int d\psi_1 ... d\psi_N \exp\Big(-\beta\Lambda_{rot}\sum_{i=1}^{N}(mod(\Psi_i - \Psi_{i,0}, \pi))^2\Big) \tag{A.7}$$

In the second line the Compton wavelength (or de Broglie wavelegth) $\Lambda_c = h(2mk_BT\pi)^{-\frac{1}{2}}$ stems from the integral over the momenta of N particles that are allowed to move in D_t dimensions in the unconstraint Einstein-molecule.
In the next step a transformation to relative coordinates,

$$\vec{r}_1 \rightarrow \vec{r}_1' = 0 \tag{A.8}$$

$$\vec{r}_2 \rightarrow \vec{r}_2' = \vec{r}_2 - \vec{r}_1 \tag{A.9}$$

$$\vec{r}_3 \rightarrow \vec{r}_3' = \vec{r}_3 - \vec{r}_1 \tag{A.10}$$

is performed. This change of coordinates is only necessary to show explicitly that the second integral is in fact independent of the position of particle 1. The integral over the position of particle 1 then gives the volume of the simulation box (or area in 2 dimensions). By renaming the remaining integrals, the partition function gives:

$$Z_{mol-id} = \frac{V}{N\Lambda_c^{D_t N}} \times \kappa_{trans} \times \kappa_{rot} \tag{A.11}$$

All in all, the free energy then becomes:

$$\frac{A_{mol\_id}}{Nk_BT} = \frac{1}{N}\ln(\rho) + D_t\ln(\Lambda_c) - \frac{1}{N}\ln(\kappa_{trans}) - \frac{1}{N}\ln(\kappa_{rot}) \tag{A.12}$$

Here, $D_t\ln(\Lambda_c)$ is also called the ideal gas contribution.

The remaining parts κ_{trans} and κ_{rot} give:

$$\kappa_{trans} = \Big(\frac{\pi}{\beta\Lambda_{trans}}\Big)^{\frac{D_t(N-1)}{2}}\Big(\mathrm{erf}\,(\frac{L_x}{2}\sqrt{\beta\Lambda_t})\Big)^{(N-1)}\Big(\mathrm{erf}\,(\frac{L_y}{2}\sqrt{\beta\Lambda_t})\Big)^{(N-1)} \approx \Big(\frac{\pi}{\beta\Lambda_{trans}}\Big)^{\frac{D_t(N-1)}{2}} \tag{A.13}$$

$$\kappa_{rot} = \frac{1}{(2\pi)^N}\Big(\frac{\pi}{\beta\Lambda_{rot}}\Big)^{\frac{D_r N}{2}}\Big(\mathrm{erf}\,(\pi\sqrt{\beta\Lambda_r})\Big)^{D_r N} \approx \frac{1}{(2\pi)^N}\Big(\frac{\pi}{\beta\Lambda_{rot}}\Big)^{\frac{D_r N}{2}} \tag{A.14}$$

where the error functions can be approximated by 1 if the argument is large

enough.
In simulations Λ_c is usually set to 1 such that the term doesn't contribute. Finally, for $D_t = 2$ and $D_r = 1$ the free energy term, also named A_0 becomes:

$$\frac{A_0}{Nk_BT} = \frac{1}{N}\ln(\rho) + (1 - \frac{1}{N})\ln(\frac{\beta\Lambda_{trans}}{\pi}) + \frac{1}{2}\ln(\frac{\beta\Lambda_{rot}}{\pi}) + \ln(2\pi) \tag{A.15}$$

For more details concerning this derivation see [22].

A.1.1. Limiting cases for A_0

View limiting cases of $A_0(\Lambda_E)$:

- $\Lambda_E \to 0$: ideal gas limit, expecting free energy of the ideal gas $A_0 = A_{id-gas}$
- $\Lambda_E \to \infty$: all positions are fixed, one configuration left, therefore expecting $A_0 = 0$

Testing this gives:

- $\Lambda_E \to 0$: - limit for κ_{trans}

$$\begin{aligned}
\lim_{\Lambda_t\to 0}\kappa_{trans} &= \lim_{\Lambda_t\to 0}\Big(\frac{\pi}{\beta\Lambda_t}\Big)^{(N-1)}\Big(\operatorname{erf}(\frac{L_x}{2}\sqrt{\beta\Lambda_t})\Big)^{(N-1)}\Big(\operatorname{erf}(\frac{L_y}{2}\sqrt{\beta\Lambda_t})\Big)^{(N-1)} \\
&= \lim_{\Lambda_t\to 0}\Big(\frac{\pi}{\beta\Lambda_t}\Big)^{(N-1)}\Big(\frac{2}{\sqrt{\pi}}\frac{L_x}{2}\sqrt{\beta\Lambda_t} + \mathcal{O}(\Lambda^{\frac{3}{2}})\Big)^{(N-1)} \\
&\quad \Big(\frac{2}{\sqrt{\pi}}\frac{L_y}{2}\sqrt{\beta\Lambda_t} + \mathcal{O}(\Lambda^{\frac{3}{2}})\Big)^{(N-1)} \\
&= \lim_{\Lambda_t\to 0}\Big(\frac{\pi}{\beta\Lambda_t}\Big)^{(N-1)}\Big(\frac{L_xL_y}{\pi}\beta\Lambda_t + \mathcal{O}(\Lambda^4)\Big)^{(N-1)} \\
&= (L_xL_y)^{(N-1)}
\end{aligned}$$

  - limit for κ_{rot}

$$\begin{aligned}
\lim_{\Lambda_r\to 0}\kappa_{rot} &= \lim_{\Lambda_r\to 0}\frac{1}{2\pi}\Big(\frac{\pi}{\beta\Lambda_r}\Big)^{\frac{N}{2}}\Big(\operatorname{erf}(\pi\sqrt{\beta\Lambda_r})\Big)^{N} \\
&= \lim_{\Lambda_r\to 0}\frac{1}{2\pi}\Big(\frac{\pi}{\beta\Lambda_r}\Big)^{\frac{N}{2}}\Big(\frac{2}{\sqrt{\pi}}(\pi\sqrt{\beta\Lambda_r}) + \mathcal{O}(\Lambda_r^{\frac{3}{2}})\Big)^{N} \\
&= \frac{1}{2\pi}(2\pi)^N = (2\pi)^{N-1}
\end{aligned}$$

A.2. Complete Autocorrelation Analysis

Complete results on the autocorrelation-study described in section 4.1.2 are collected here in table A.1 and table A.2.

N	ρ^*	$\omega[^\circ]$	$\tau_{a.c.}^{trans}$[cycles] $\lambda_{ID} = 1$	$\lambda_{ID} = 50$	$\lambda_{ID} = 100$
32	0.95	0	1.32E+05	5.00E+03	3.00E+03
32	0.95	30	1.27E+05	8.00E+03	3.00E+03
32	0.98	0	1.15E+05	8.00E+03	3.00E+03
32	0.98	30	2.28E+05	5.00E+03	1.00E+03
72	0.95	0	5.10E+05	6.00E+03	1.00E+03
72	0.95	30	1.78E+05	8.00E+03	1.00E+03
72	0.98	0	8.10E+05	6.00E+03	5.00E+03
72	0.98	30	3.30E+05	1.00E+04	1.00E+04
128	0.95	0	3.57E+06	1.04E+05	4.00E+03
128	0.95	30	2.09E+06	3.60E+04	1.20E+04
128	0.98	0	3.31E+06	3.80E+04	1.00E+04
128	0.98	30	3.41E+06	4.00E+04	1.60E+04
200	0.95	0	1.27E+06	1.72E+05	1.00E+04
200	0.95	30	3.63E+06	6.00E+04	8.00E+03
200	0.98	0	3.36E+06	5.20E+04	1.00E+04
200	0.98	30	4.45E+06	4.40E+04	1.40E+04
288	0.95	0	1.62E+06	1.62E+05	6.00E+03
288	0.95	30	3.74E+06	5.20E+04	8.00E+03
288	0.98	0	3.49E+06	5.80E+04	1.40E+04
288	0.98	30	6.12E+06	8.00E+04	2.20E+04

N	ρ^*	$\omega[^\circ]$	$\lambda_{ID} = 1$	$\lambda_{ID} = 51$	$\lambda_{ID} = 96$
392	0.95	0	5.568E+06	5.40E+04	3.0E+03
392	0.95	30	3.657E+06	5.40E+04	3.0E+03
392	0.98	0	6.855E+06	6.30E+04	3.0E+03
392	0.98	30	5.403E+06	3.60E+04	6.0E+03

Table A.1.: Auto-correlation times for translations and rotations $\tau_{a.c.}$ for different λ-states (given by the ID for each lambda) for system sizes $N = 32$, $N = 72$, $N = 128$, $N = 200$, $N = 288$ and $N = 392$, densities ρ^* and lattice states ω.

			$\tau_{a.c.}^{trans}$[cycles]		
N	ρ^*	$\omega[^\circ]$	$\lambda_{ID} = 1$	$\lambda_{ID} = 50$	$\lambda_{ID} = 100$
32	0.95	0	1.80E+04	6.00E+03	1.00E+03
32	0.95	30	1.50E+04	5.00E+03	4.00E+03
32	0.98	0	4.50E+04	4.00E+03	5.00E+03
32	0.98	30	2.50E+04	2.00E+03	1.00E+03
72	0.95	0	1.33E+05	4.00E+03	2.00E+03
72	0.95	30	2.70E+04	2.00E+03	1.00E+03
72	0.98	0	3.02E+05	4.00E+03	3.00E+03
72	0.98	30	9.80E+04	5.00E+03	2.00E+03
128	0.95	0	1.15E+06	2.40E+04	4.00E+03
128	0.95	30	1.08E+06	2.00E+04	6.00E+03
128	0.98	0	1.43E+06	2.40E+04	1.00E+04
128	0.98	30	6.70E+05	3.60E+04	4.00E+03
200	0.95	0	8.58E+05	2.20E+04	4.00E+03
200	0.95	30	4.72E+05	1.80E+04	4.00E+03
200	0.98	0	2.73E+06	3.60E+04	6.00E+03
200	0.98	30	2.75E+06	5.20E+04	2.00E+03
288	0.95	0	2.86E+06	2.60E+04	2.00E+03
288	0.95	30	1.73E+06	3.20E+04	4.00E+03
288	0.98	0	2.27E+06	1.60E+04	4.00E+03
288	0.98	30	9.98E+05	3.80E+04	4.00E+03

N	ρ^*	$\omega[^\circ]$	$\lambda_{ID} = 1$	$\lambda_{ID} = 51$	$\lambda_{ID} = 96$
392	0.95	0	2.196E+06	1.50E+04	6.00E+03
392	0.95	30	4.200E+05	1.80E+04	3.00E+03
392	0.98	0	1.950E+06	2.40E+04	6.00E+03
392	0.98	30	1.110E+06	2.10E+04	3.00E+03

Table A.2.: Auto-correlation times for rotational Einstein Energies $\tau_{a.c.}^{rot}$ for different λ-states (given by the ID for each lambda) for system sizes $N = 32$, $N = 72$, $N = 128$, $N = 200$, $N = 288$ and $N = 392$, densities ρ^* and lattice states ω.

A.3. Truncation Analysis on Free Energies

The truncation analysis on the total free energies $A_{tot}^{(p)}$ and $A_{tot}^{(d)}$ for densities $\rho^* = 0.95$ and 0.98 was performed for system sizes $N = 128$ and larger. All results of this analysis are presented here:

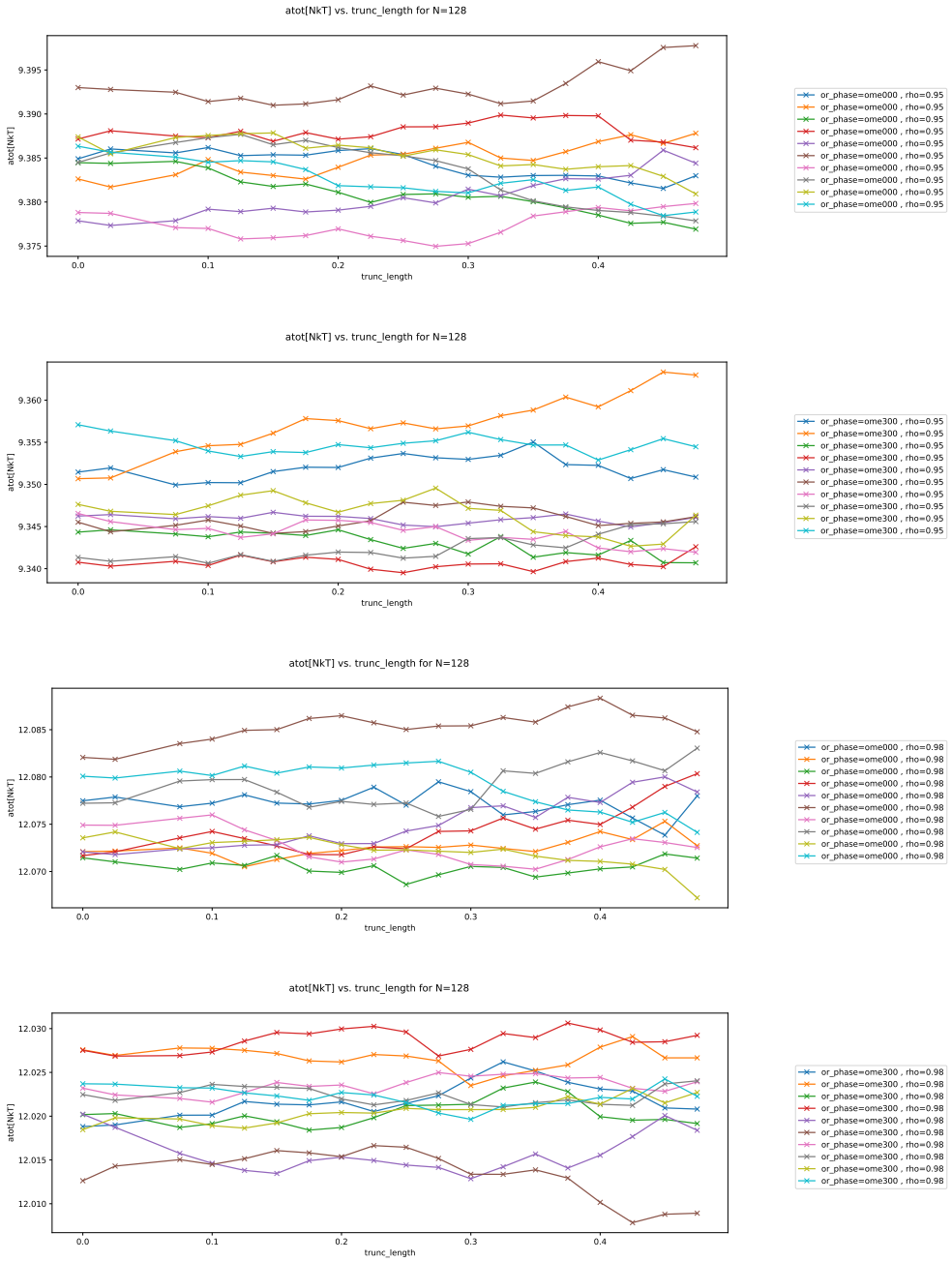


Figure A.1.: Truncation analysis for system size $N = 128$ for individual simulation trajectories.

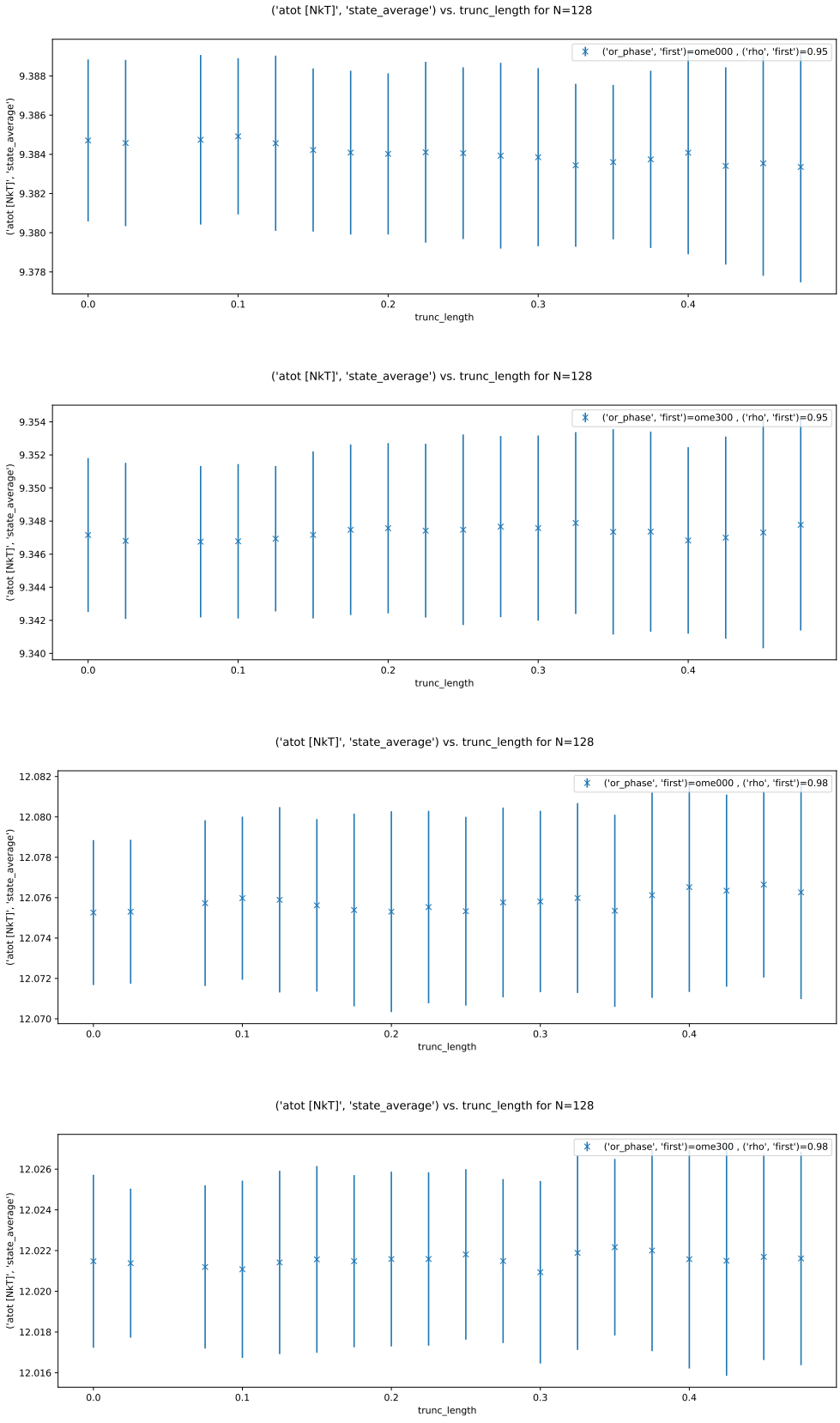


Figure A.2.: Truncation analysis for system size $N = 128$ average taken over all simulation trajectories

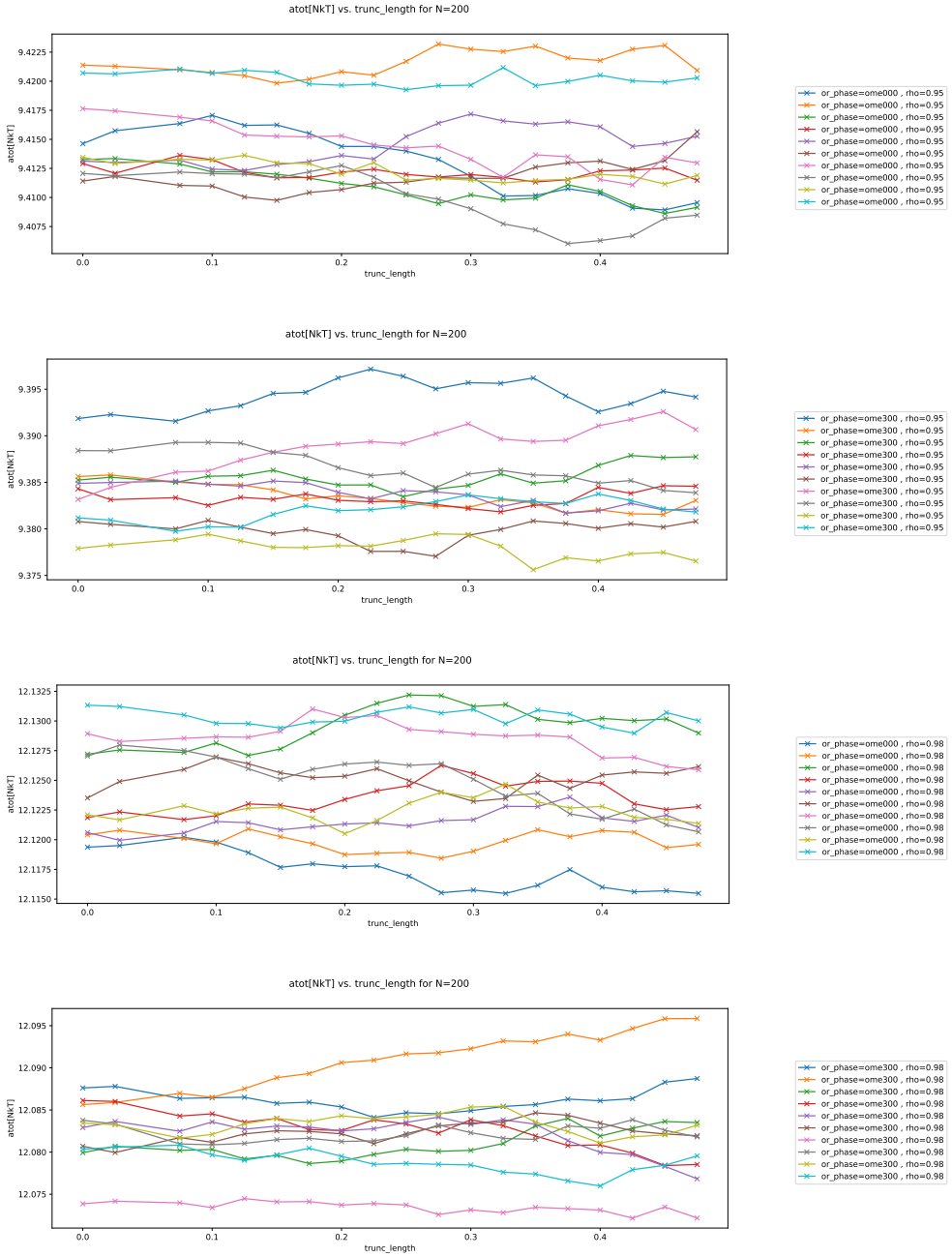


Figure A.3.: Truncation analysis for system size $N = 200$ for individual simulation trajectories.

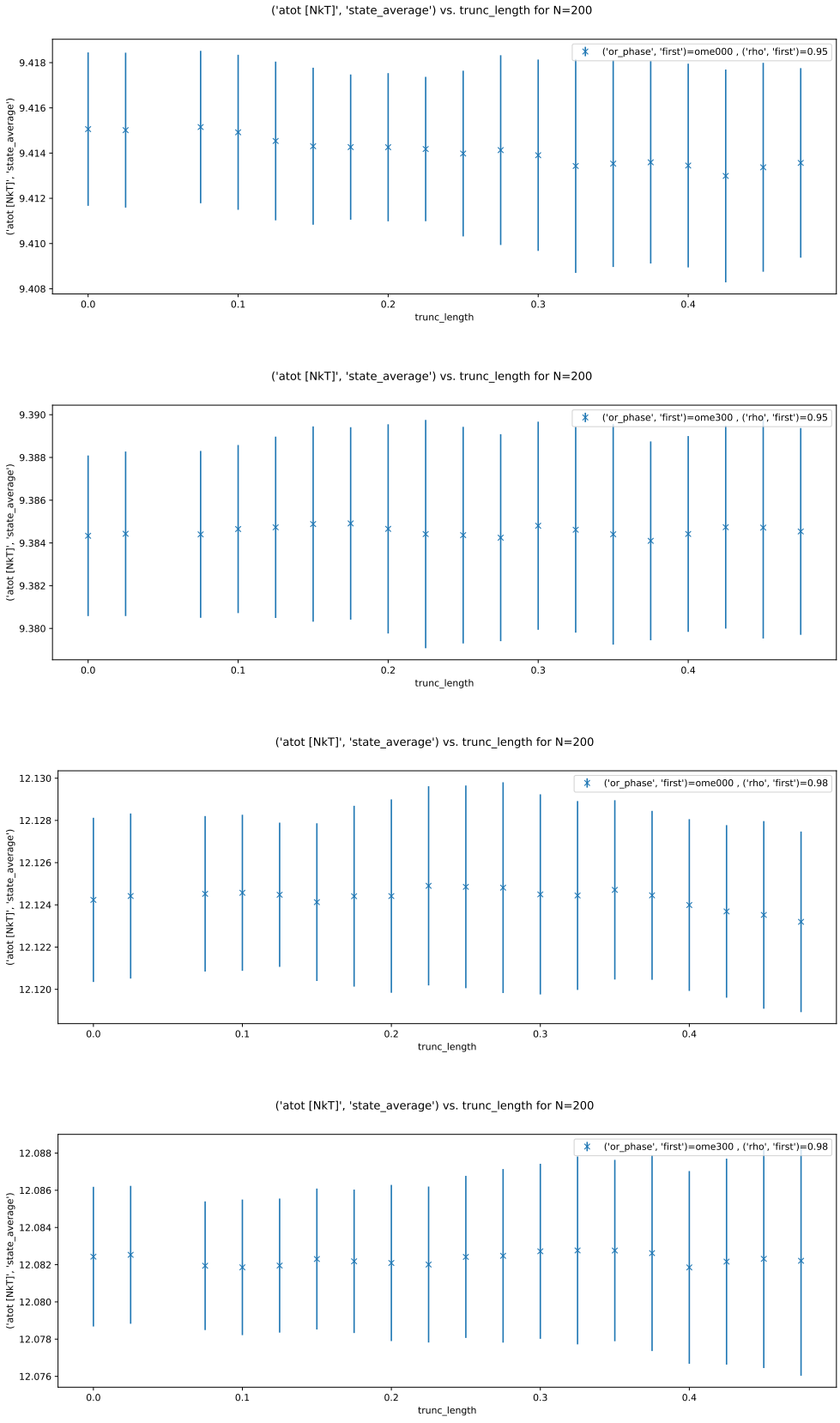


Figure A.4.: Truncation analysis for system size $N = 200$ average taken over all simulation trajectories

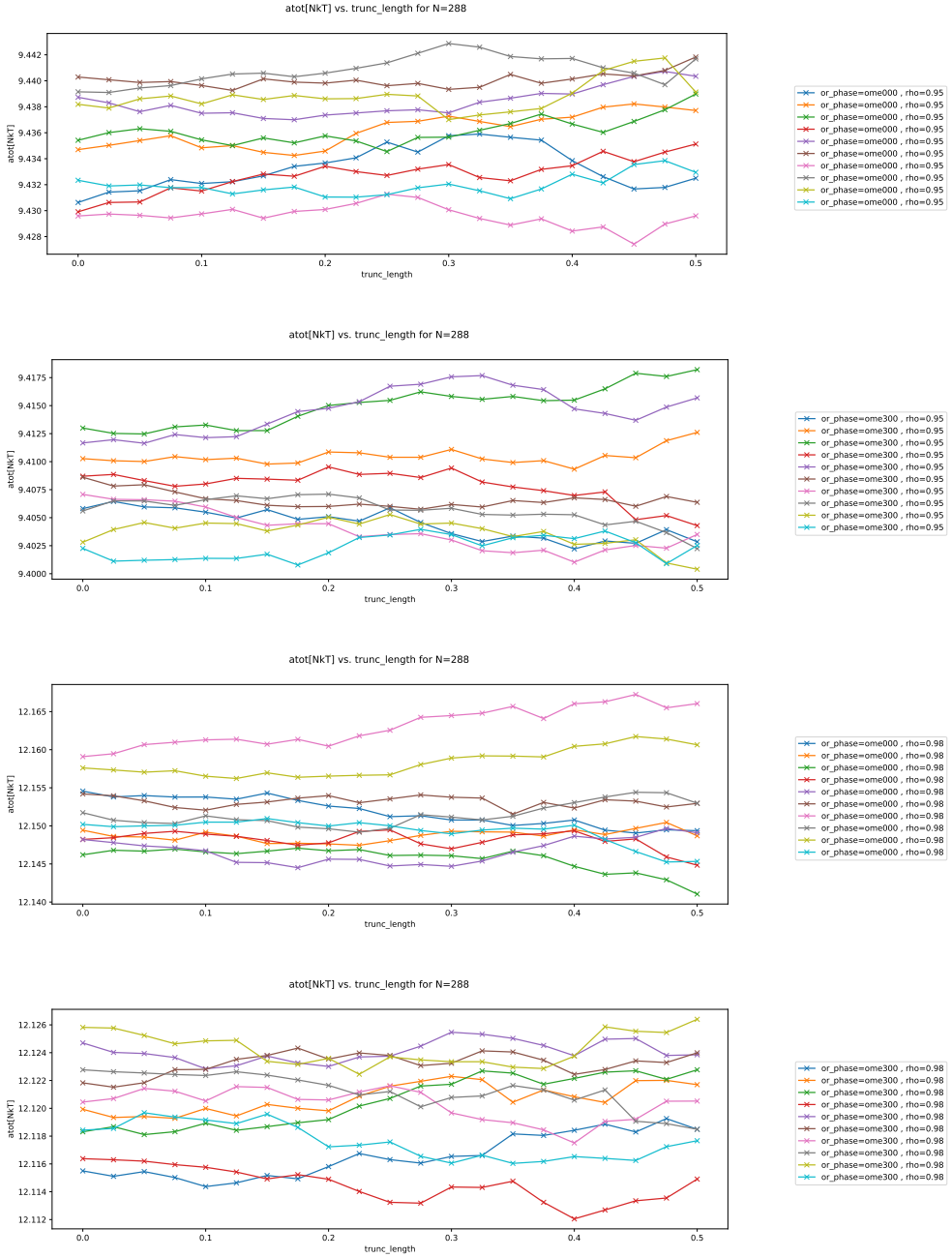


Figure A.5.: Truncation analysis for system size $N = 288$ for individual simulation trajectories.

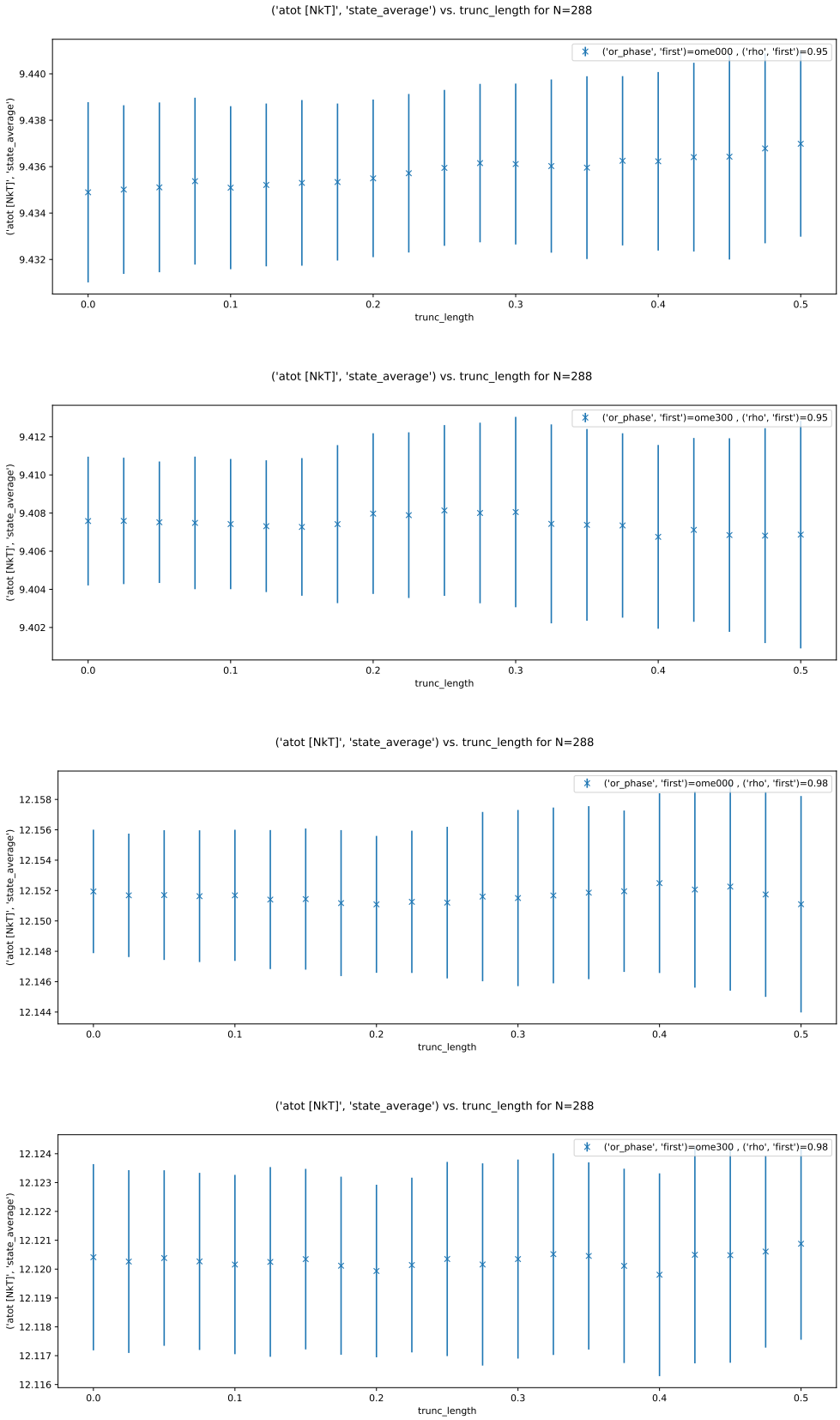


Figure A.6.: Truncation analysis for system size $N = 288$ average taken over all simulation trajectories

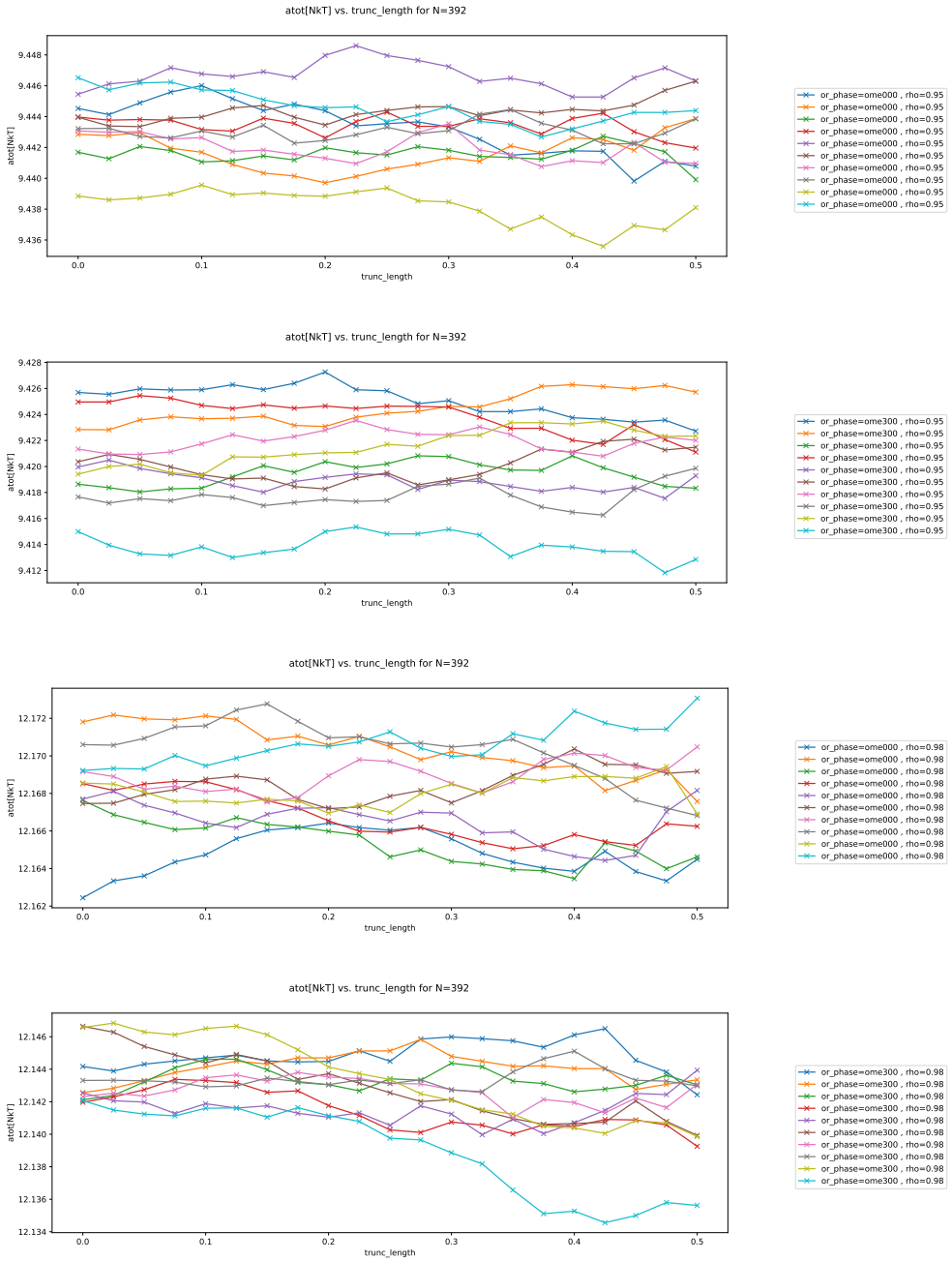


Figure A.7.: Truncation analysis for system size $N = 392$ for individual simulation trajectories.

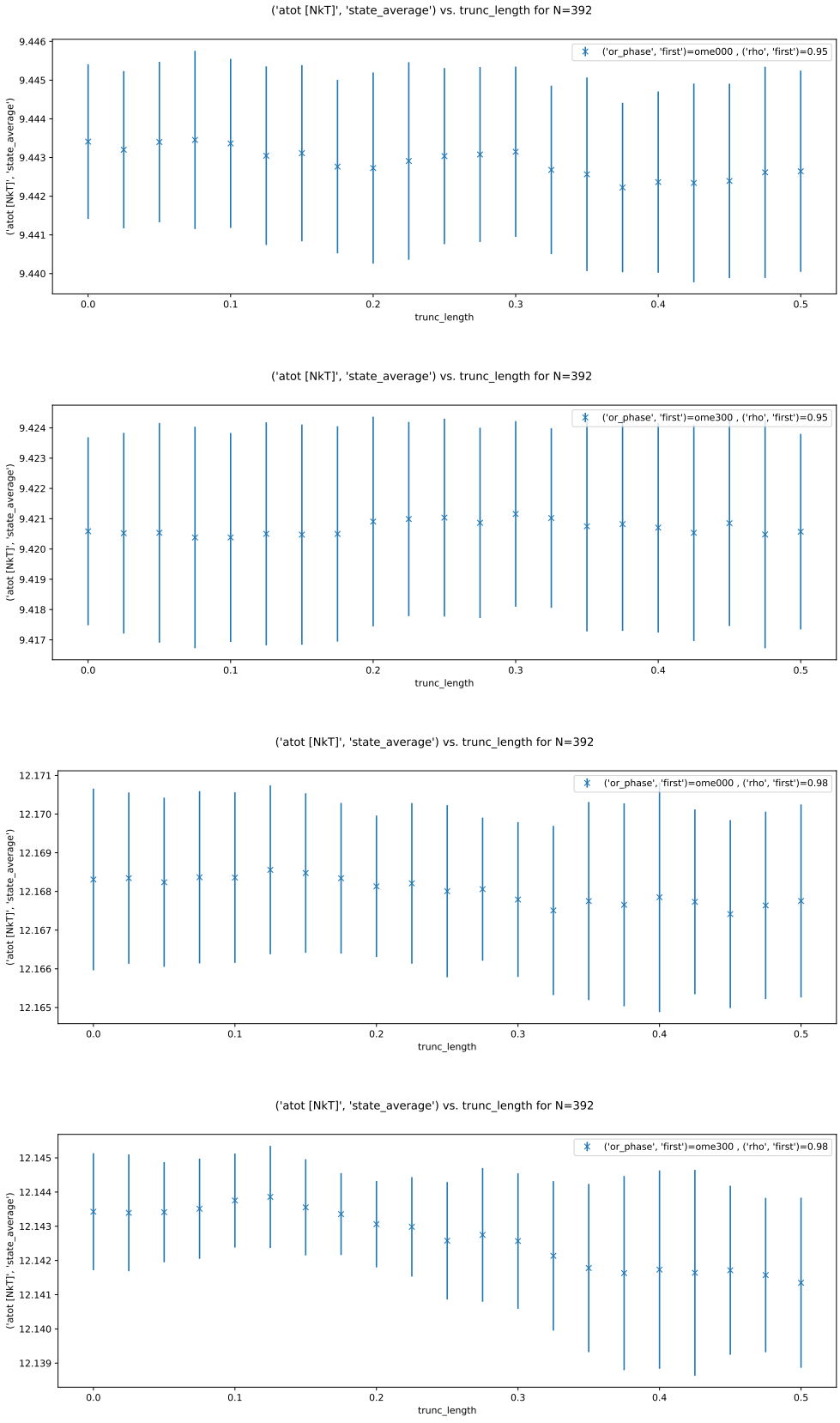


Figure A.8.: Truncation analysis for system size $N = 392$ average taken over all simulation trajectories

A.4. Simulation Trajectories

The following figures show the simulation trajectories for different states of λ ($\lambda_{\mathrm{ID}} = 1$, 50 and 100) in a system of $N = 288$ particle. The trajectories are shown for the lattice's translational Einstein energy as well as the rotational Einstein energy. The x-axes are given in units of the sampling step size (=8E+6, 2E+5 and 2E+5 cycles from left to right). The block-analysis is performed with a block size of $s_{\mathrm{block}} = 20$. Unit of the y-axes is $k_{\mathrm{B}}T$.

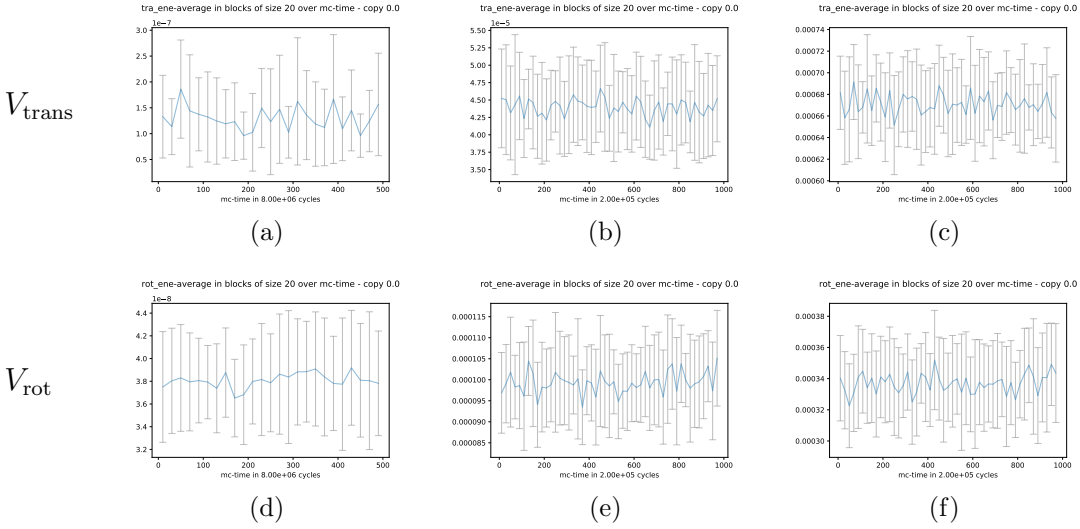


Figure A.10.: Parallel lattice state ($\omega = 30°$) at $\rho^* = 0.95$. Top row: trajectories of the translational Einstein energy for $\lambda_{\text{ID}} = 1$, 50 and 100 (from left to right). Bottom row: trajectories of the rotational Einstein energy.

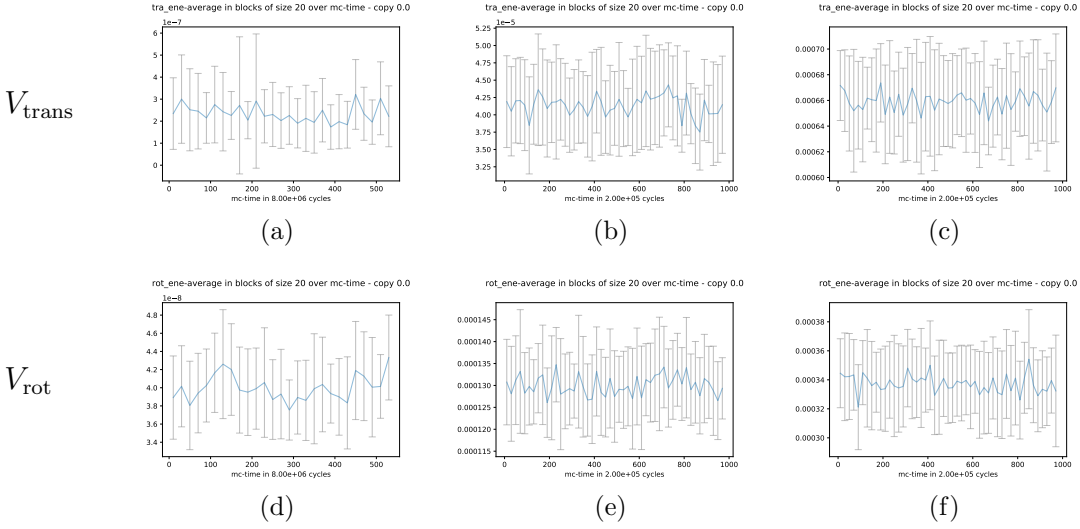


Figure A.9.: Diagonal lattice state ($\omega = 0°$) at $\rho^* = 0.95$. Top row: trajectories of the translational Einstein energy for $\lambda_{\text{ID}} = 1$, 50 and 100 (from left to right). Bottom row: trajectories of the rotational Einstein energy.

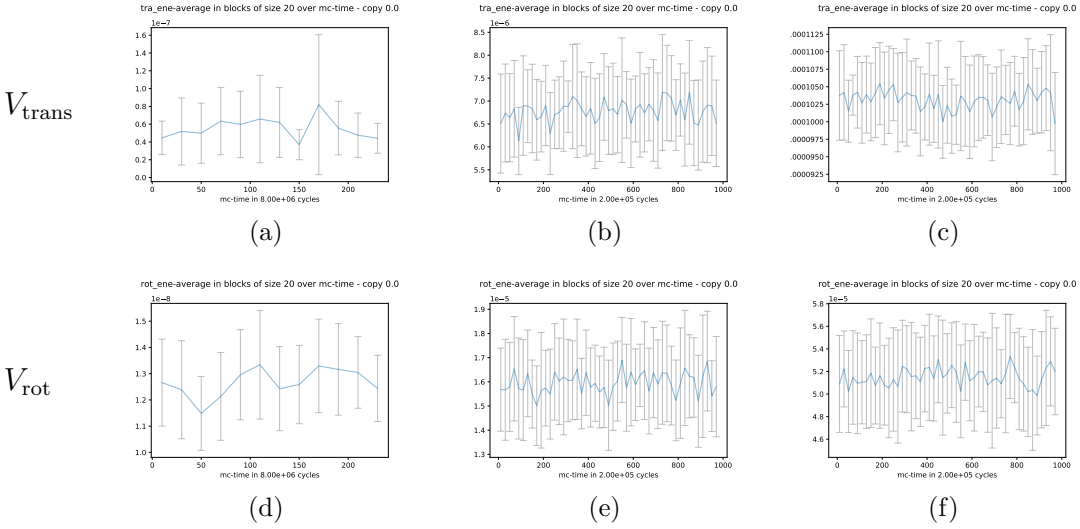


Figure A.12.: Parallel lattice state ($\omega = 30°$) at $\rho^* = 0.98$. Top row: trajectories of the translational Einstein energy for $\lambda_{\text{ID}} = 1$, 50 and 100 (from left to right). Bottom row: trajectories of the rotational Einstein energy.

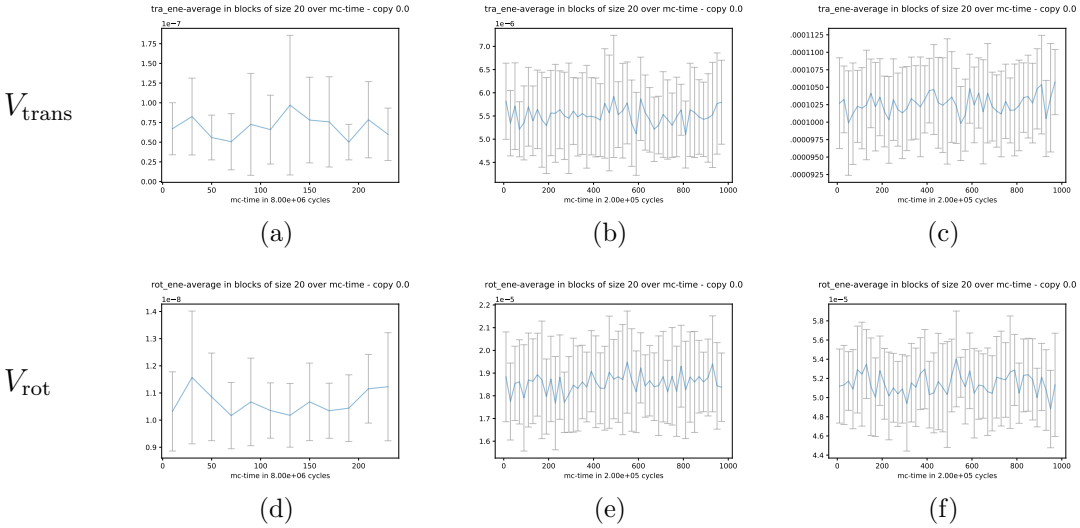


Figure A.11.: Diagonal lattice state ($\omega = 0°$) at $\rho^* = 0.98$. Top row: trajectories of the translational Einstein energy for $\lambda_{\text{ID}} = 1$, 50 and 100 (from left to right). Bottom row: trajectories of the rotational Einstein energy.

B. Appendix - Patchy Elliptic Particles

B.1. Simulation Trajectories $\theta = 45°$ and $\theta = 15°$

Complete record of energy, number of particle and energy per particle trajectories of all simulations performed, underlying the self assembly study in part III.

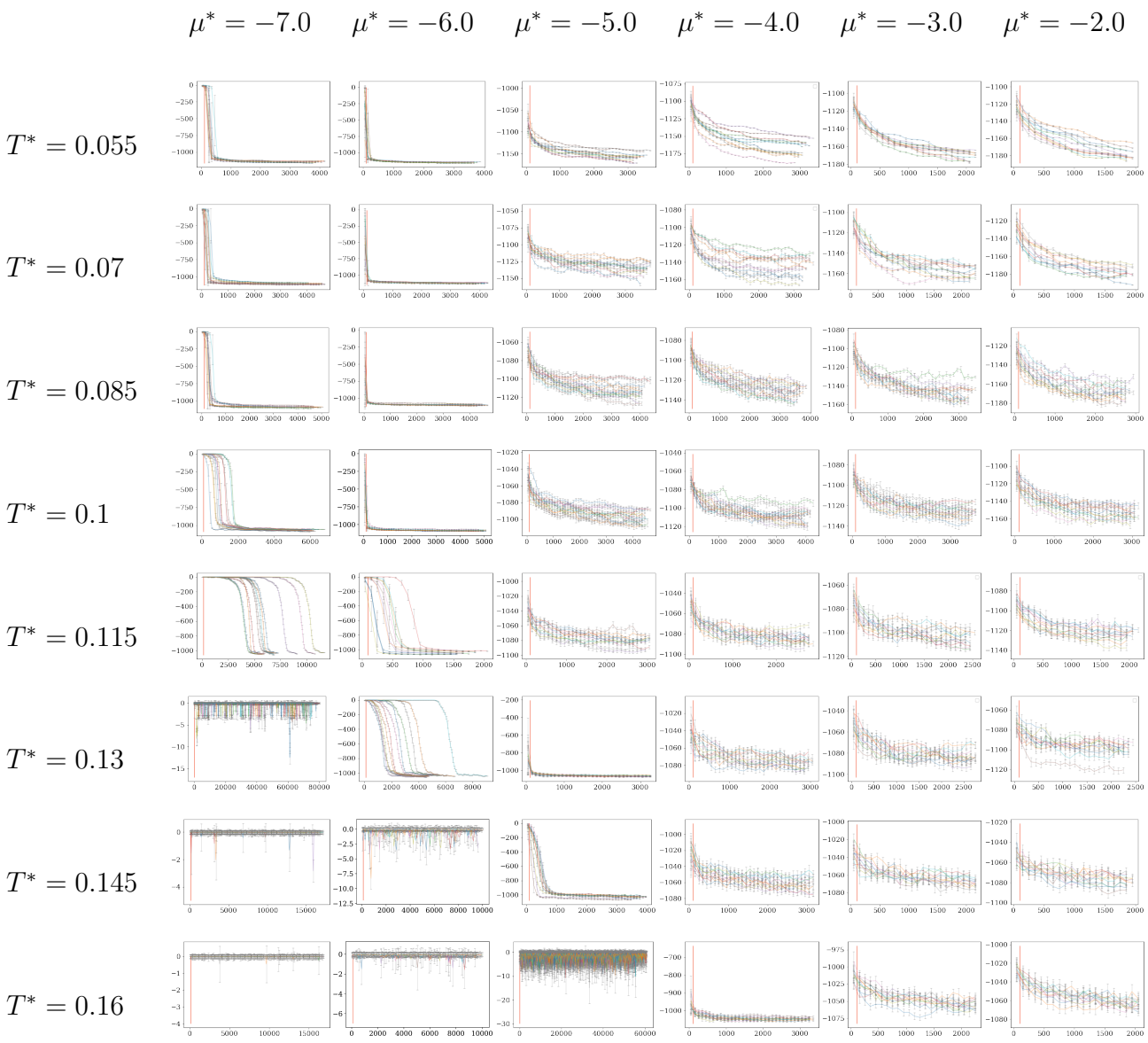


Figure B.1.: Energy trajectories for EPP with $\theta = 45°$. For temperatures (rows) and chemical potential (columns). Monte-Carlo time on the x-axes is given in units of 2000 cycles. The energy on the y-axes is given in units of the square-well depth ϵ of the patch potential.

$\mu^* = -1.0$ $\mu^* = 0.0$ $\mu^* = 1.0$ $\mu^* = 2.0$ $\mu^* = 3.0$ $\mu^* = 4.0$ $\mu^* = 5.0$

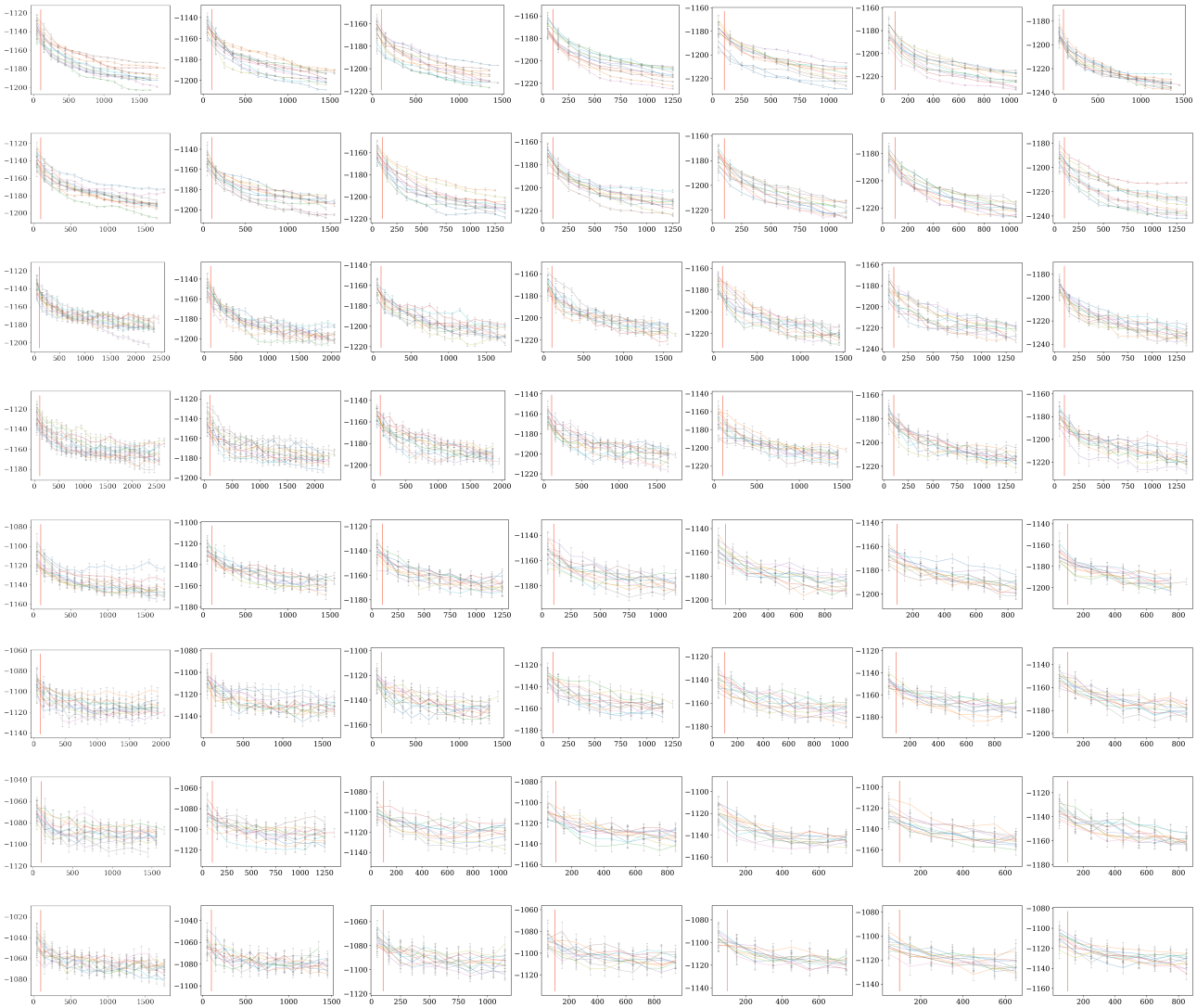


Figure B.2.: Continuation of energy trajectories for EPP with $\theta = 45°$ in appendix B.1.

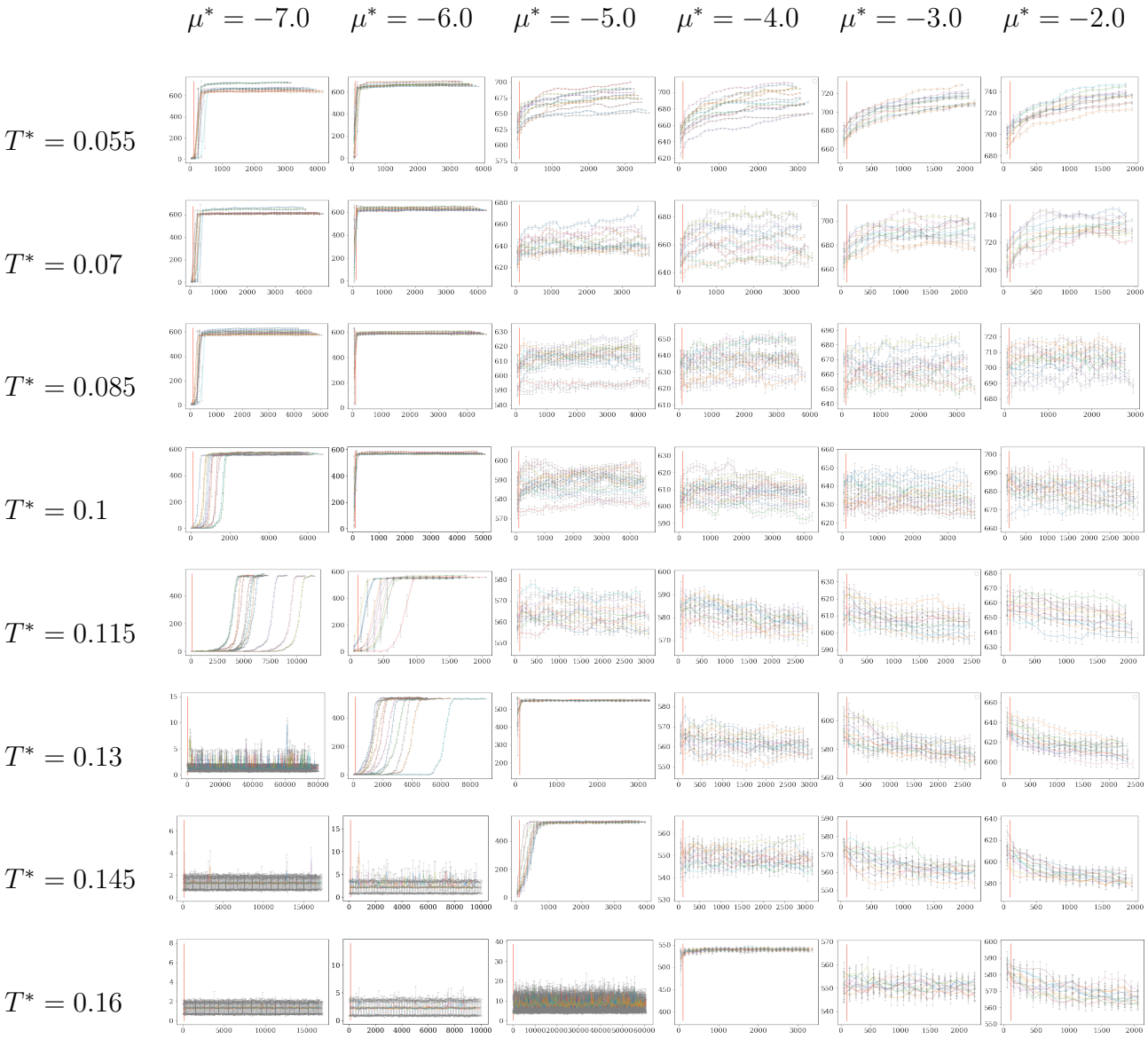


Figure B.3.: Number of Particles trajectories for EPP with $\theta = 45°$. For temperatures (rows) and chemical potential (columns). Monte-Carlo time on the x-axes is given in units of 2000 cycles.

$\mu^* = -1.0$ $\mu^* = 0.0$ $\mu^* = 1.0$ $\mu^* = 2.0$ $\mu^* = 3.0$ $\mu^* = 4.0$ $\mu^* = 5.0$

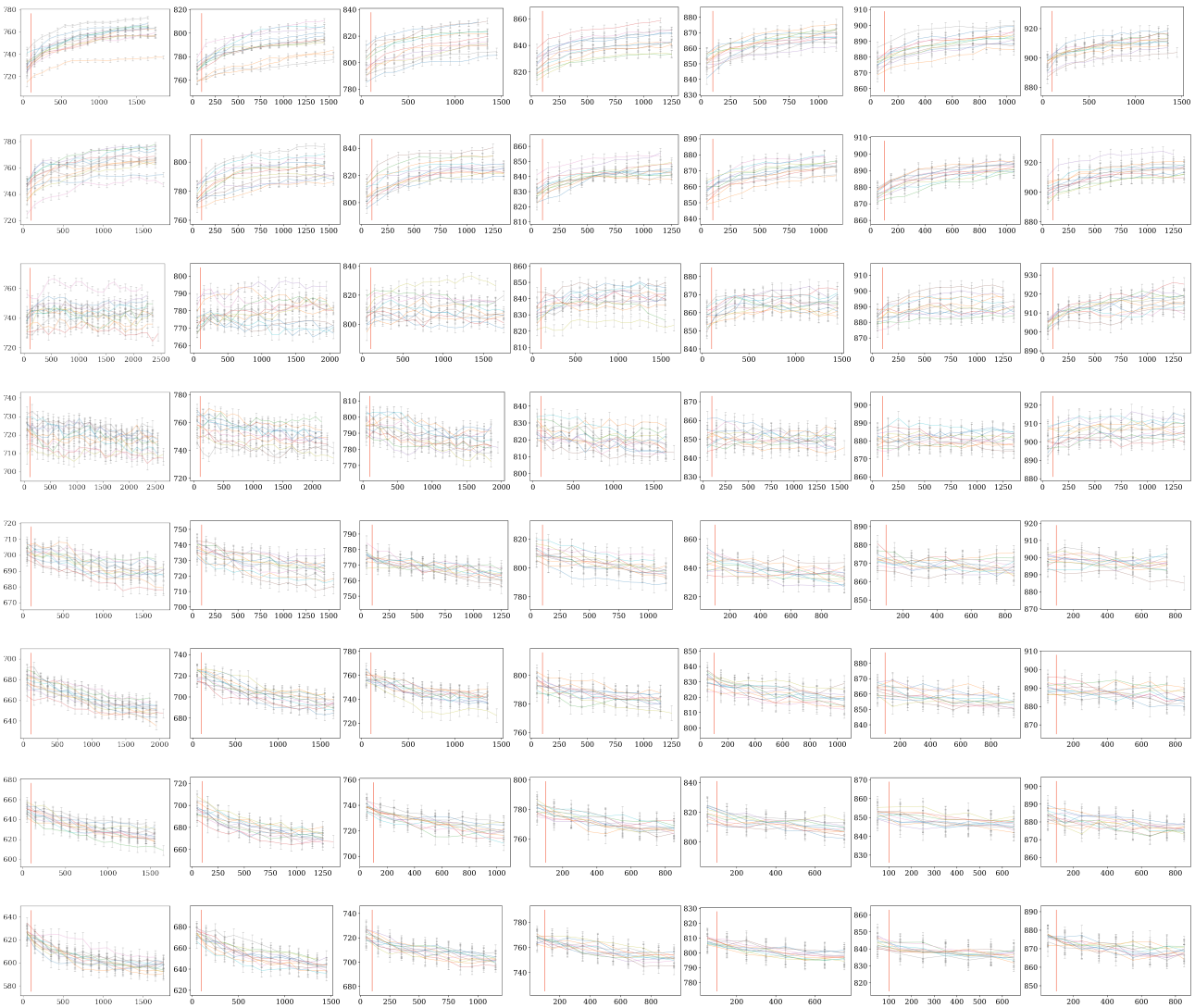


Figure B.4.: Continuation of Number of Particles trajectories for EPP with $\theta = 45°$ in appendix B.1.

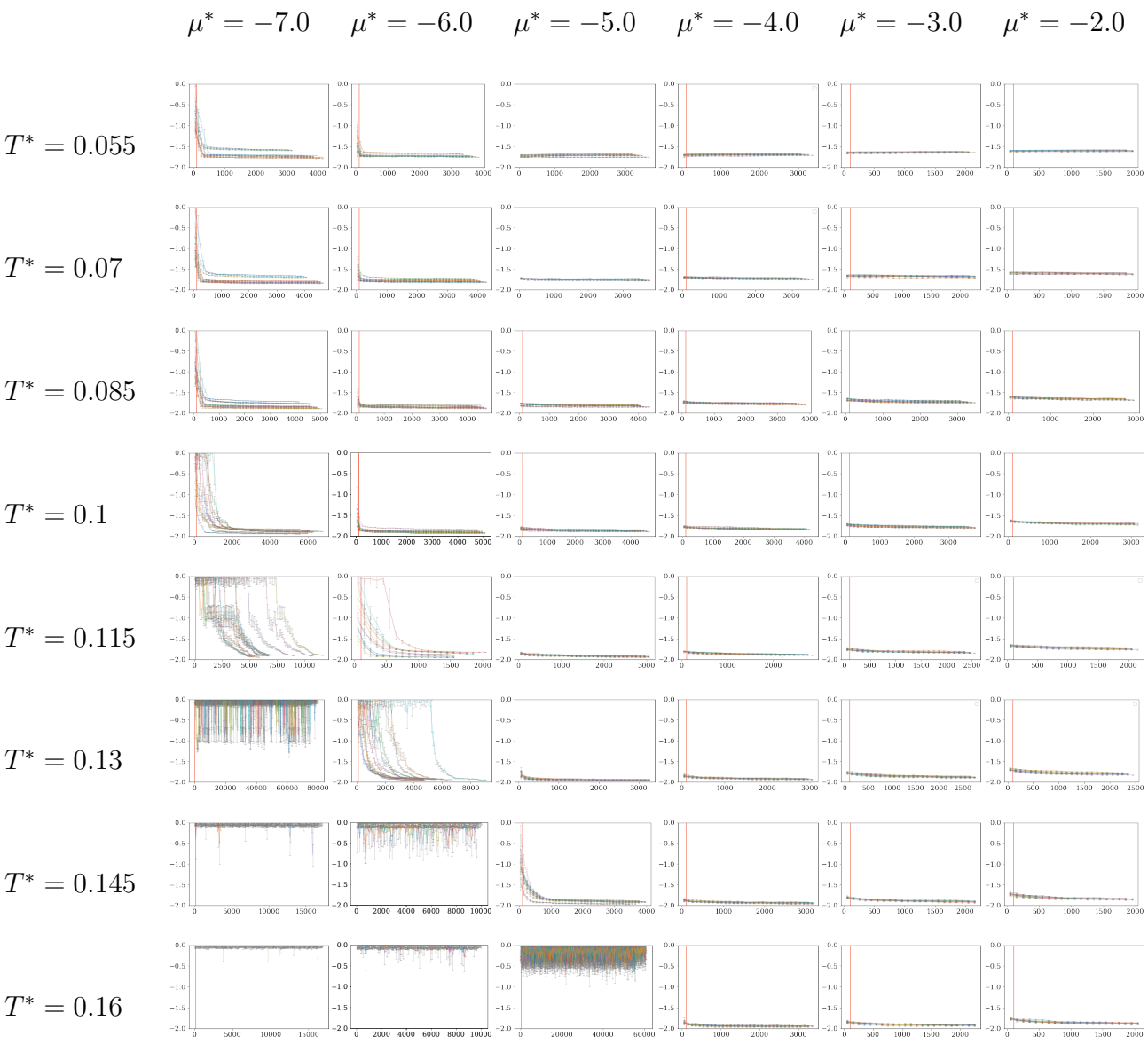


Figure B.5.: Energy per Particle trajectories for EPP with $\theta = 45°$. For temperatures (rows) and chemical potential (columns). Monte-Carlo time on the x-axes is given in units of 2000 cycles. The energy on the y-axes is given in units of the square-well depth ϵ of the patch potential and the particle number N.

$\mu^* = -1.0$ $\mu^* = 0.0$ $\mu^* = 1.0$ $\mu^* = 2.0$ $\mu^* = 3.0$ $\mu^* = 4.0$ $\mu^* = 5.0$

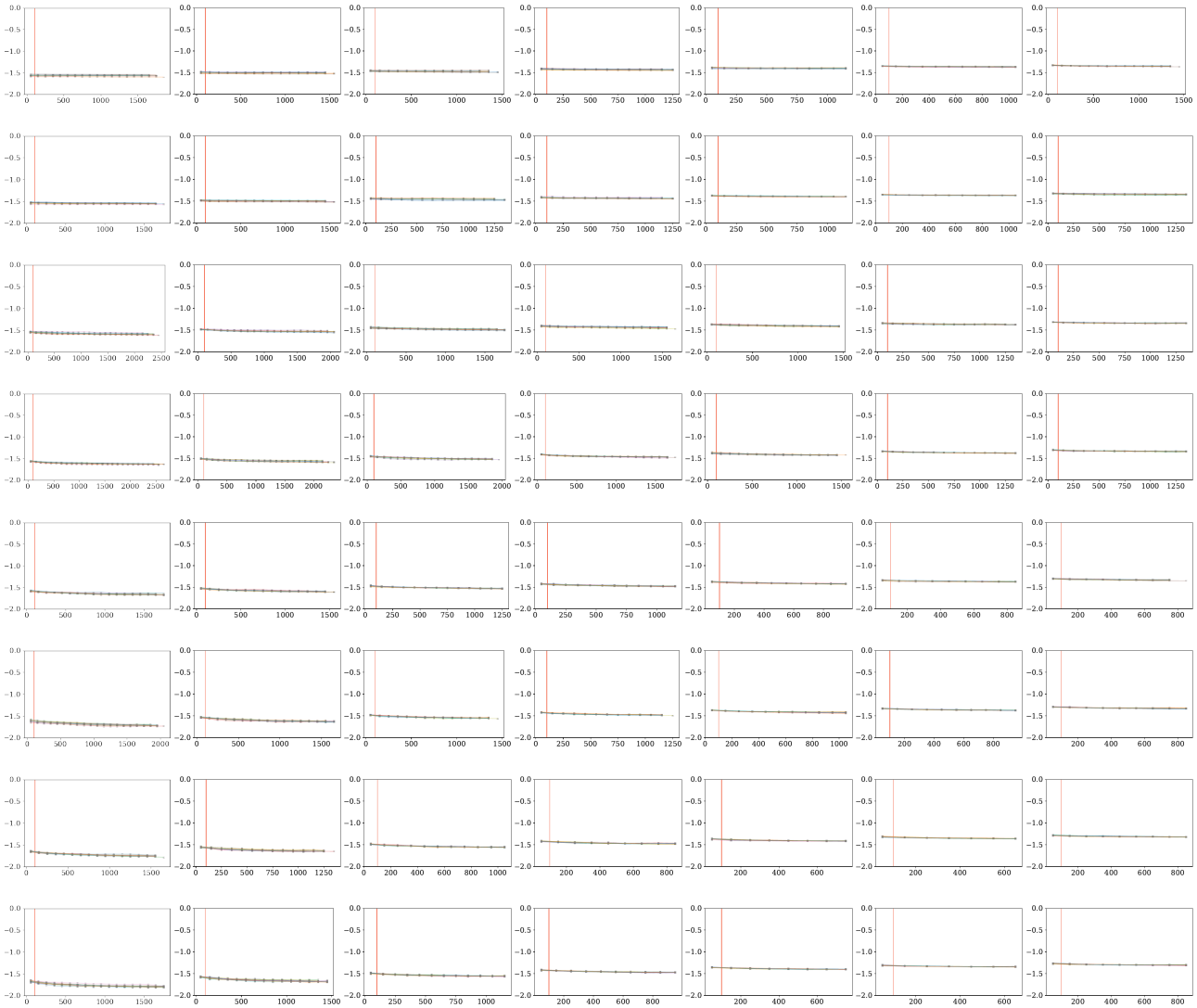


Figure B.6.: Continuation of energy per particle trajectories for EPP with $\theta = 45°$ in appendix B.1.

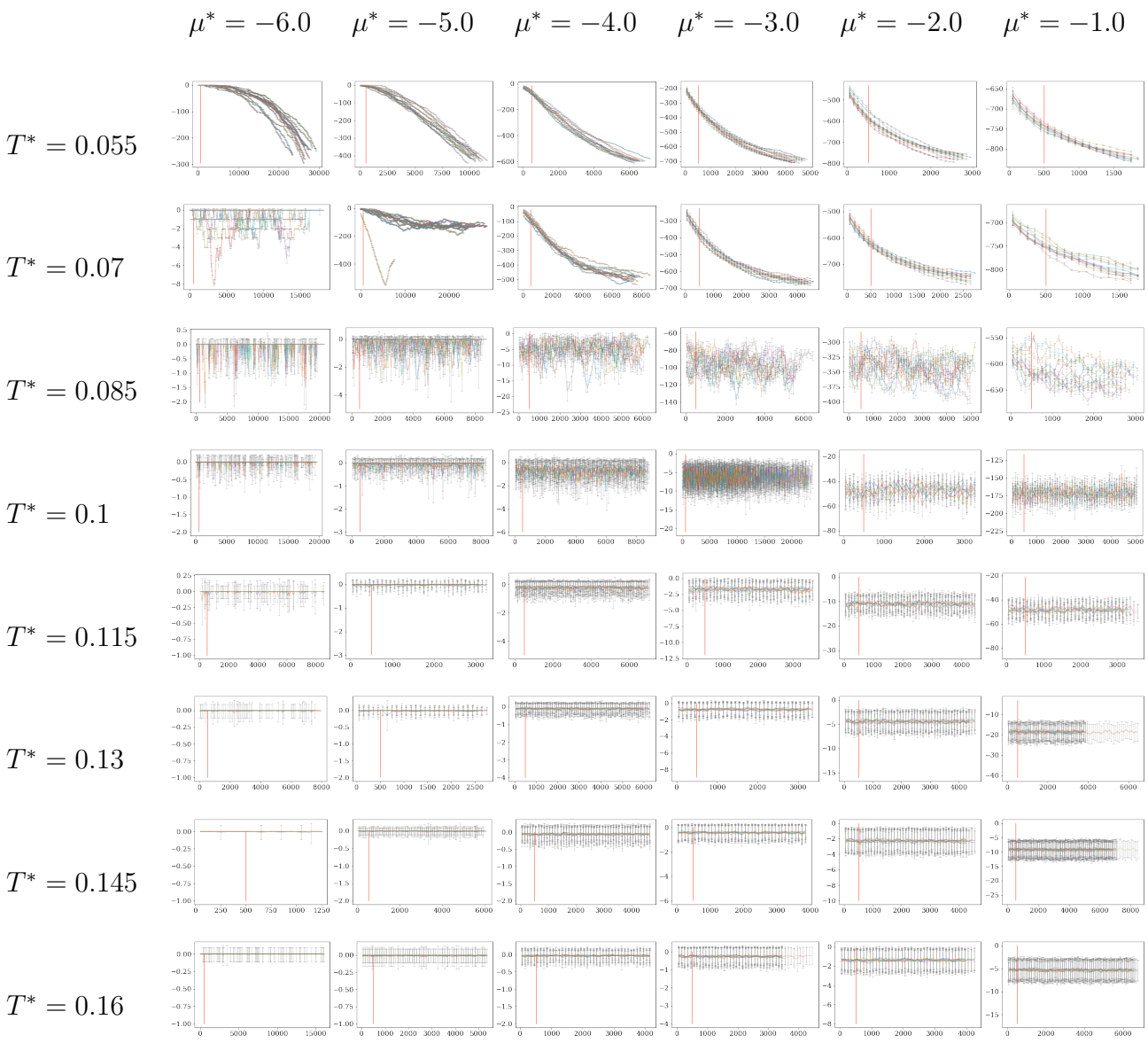


Figure B.7.: Energy trajectories for EPP with $\theta = 15°$. For temperatures (rows) and chemical potential (columns). Monte-Carlo time on the x-axes is given in units of 2000 cycles. The energy on the y-axes is given in units of the square-well depth ϵ of the patch potential.

$\mu^* = 0.0$ $\mu^* = 1.0$ $\mu^* = 2.0$ $\mu^* = 3.0$ $\mu^* = 4.0$ $\mu^* = 5.0$ $\mu^* = 6.0$

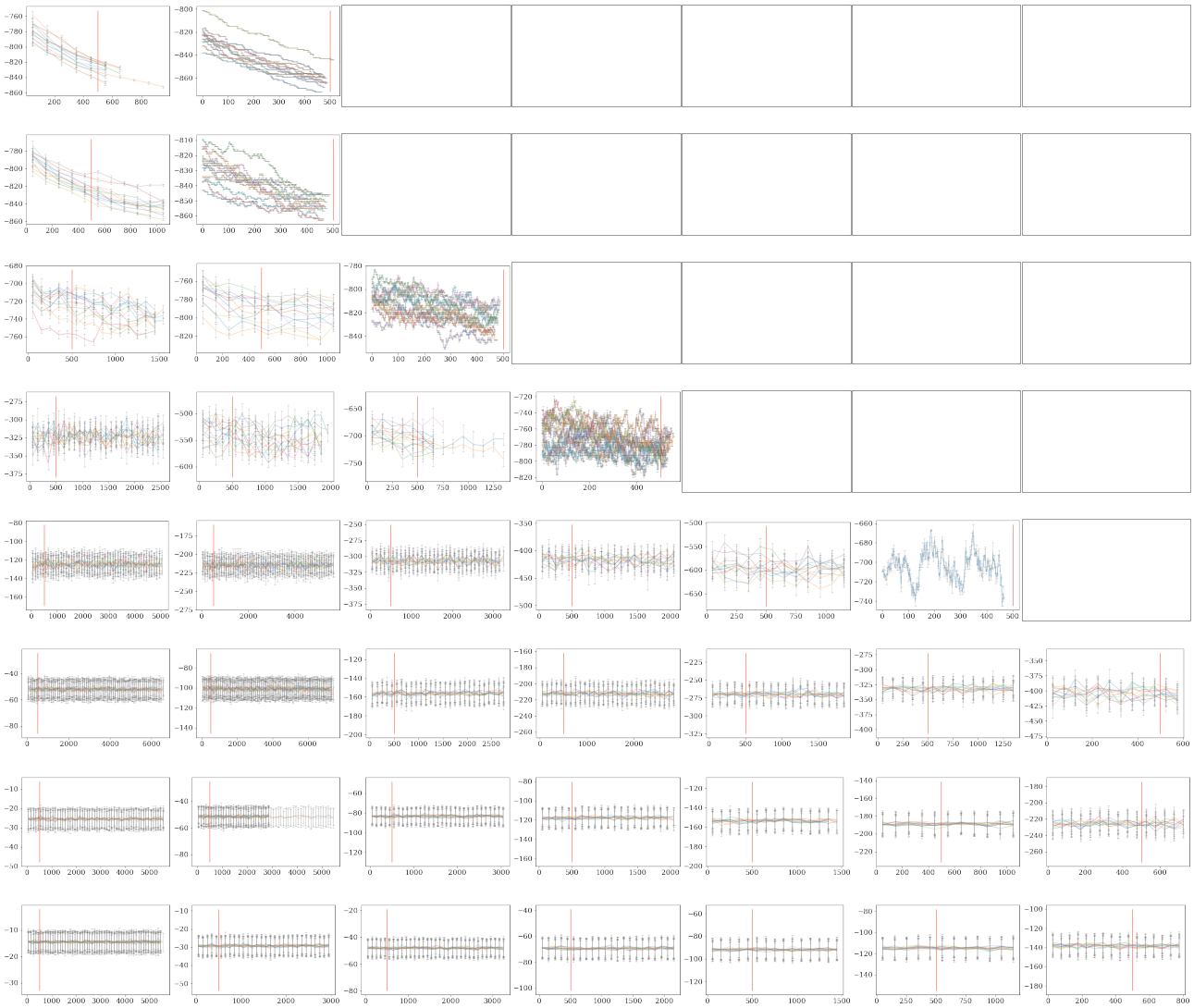


Figure B.8.: Continuation of energy trajectories for EPP with $\theta = 15°$ in appendix B.1.

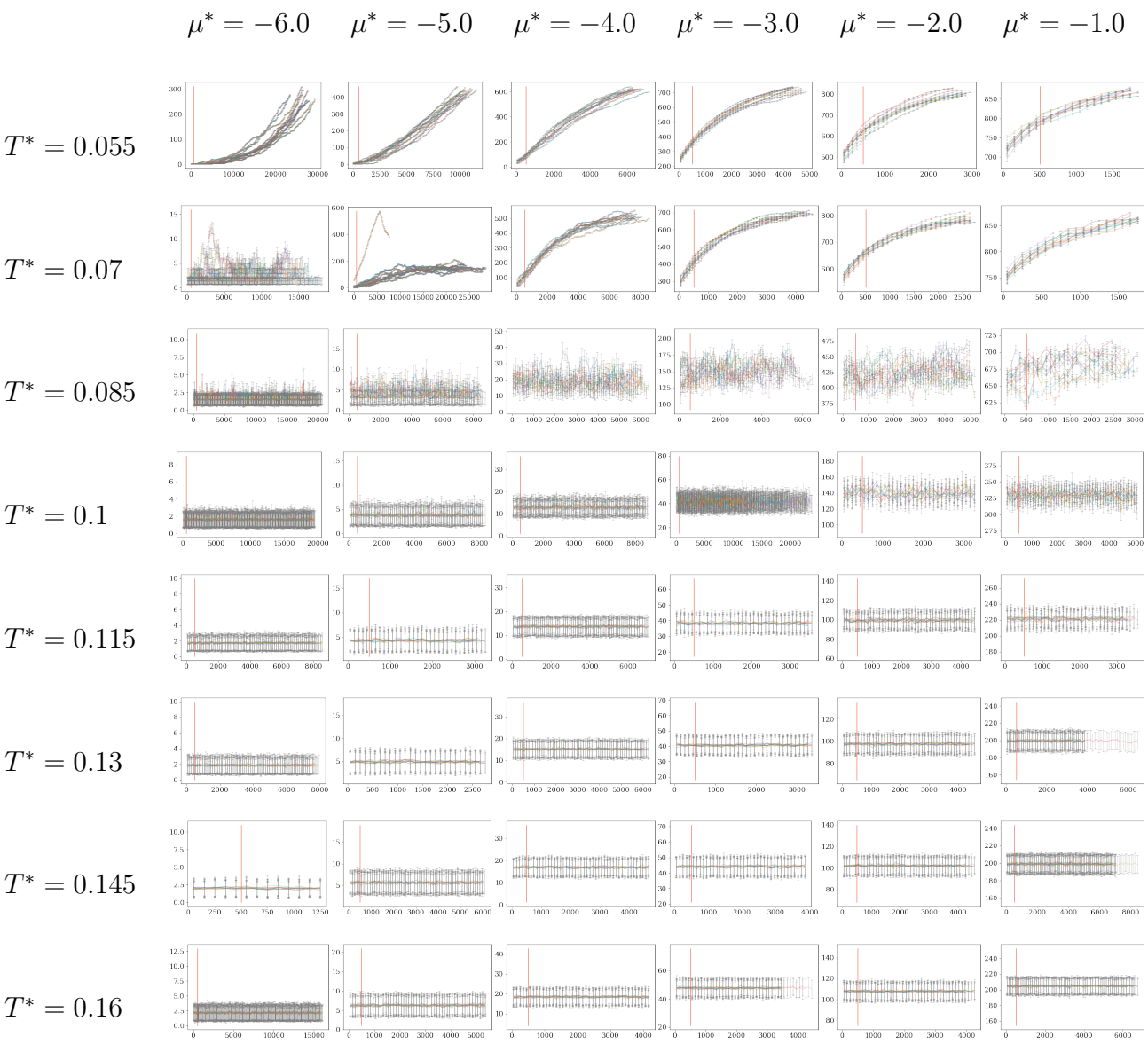


Figure B.9.: Number of Particles trajectories for EPP with $\theta = 15°$. For temperatures (rows) and chemical potential (columns). Monte-Carlo time on the x-axes is given in units of 2000 cycles.

$\mu^* = 0.0$ $\mu^* = 1.0$ $\mu^* = 2.0$ $\mu^* = 3.0$ $\mu^* = 4.0$ $\mu^* = 5.0$ $\mu^* = 6.0$

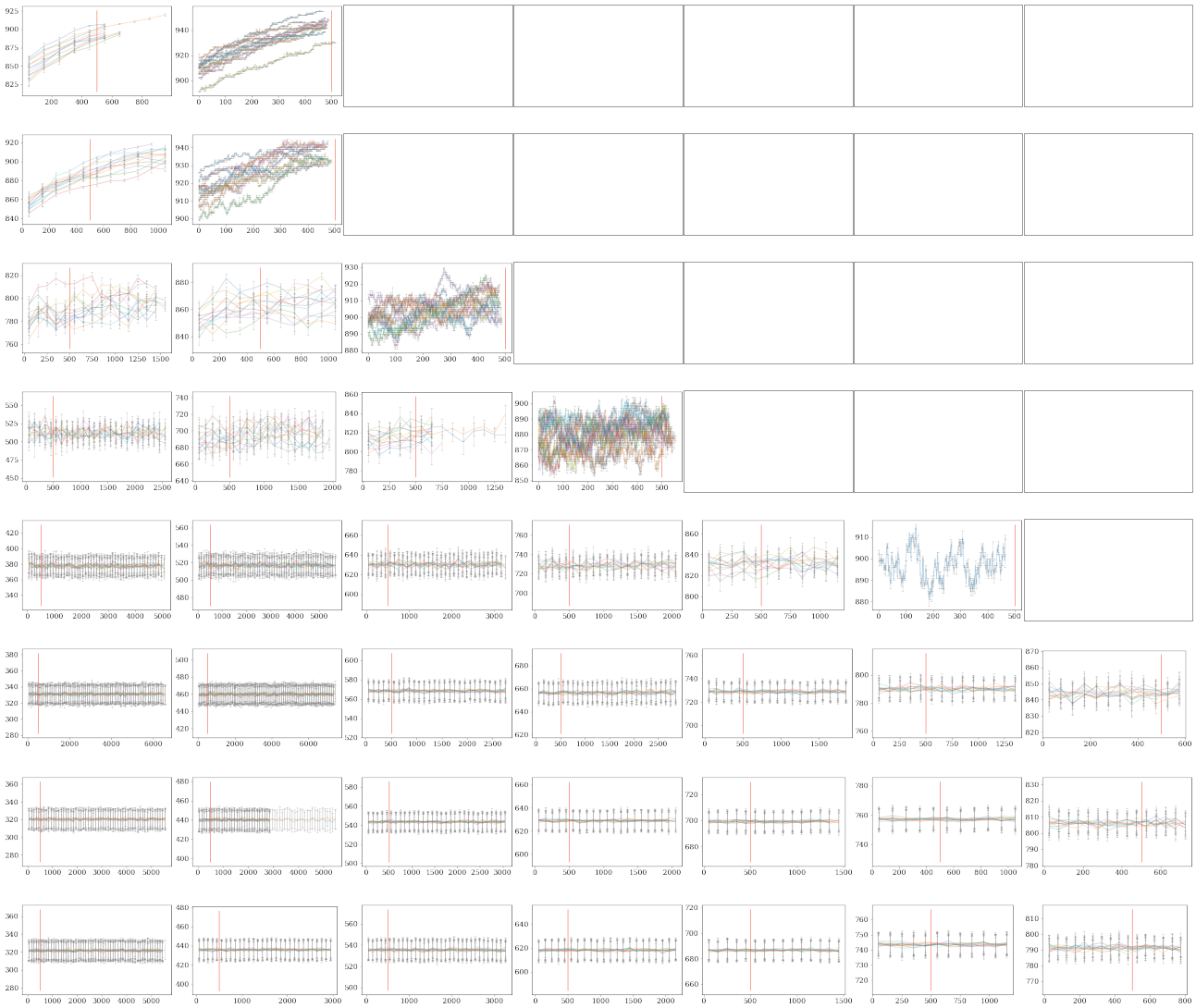


Figure B.10.: Continuation of Number of Particles trajectories for EPP with $\theta = 15°$ in appendix B.1.

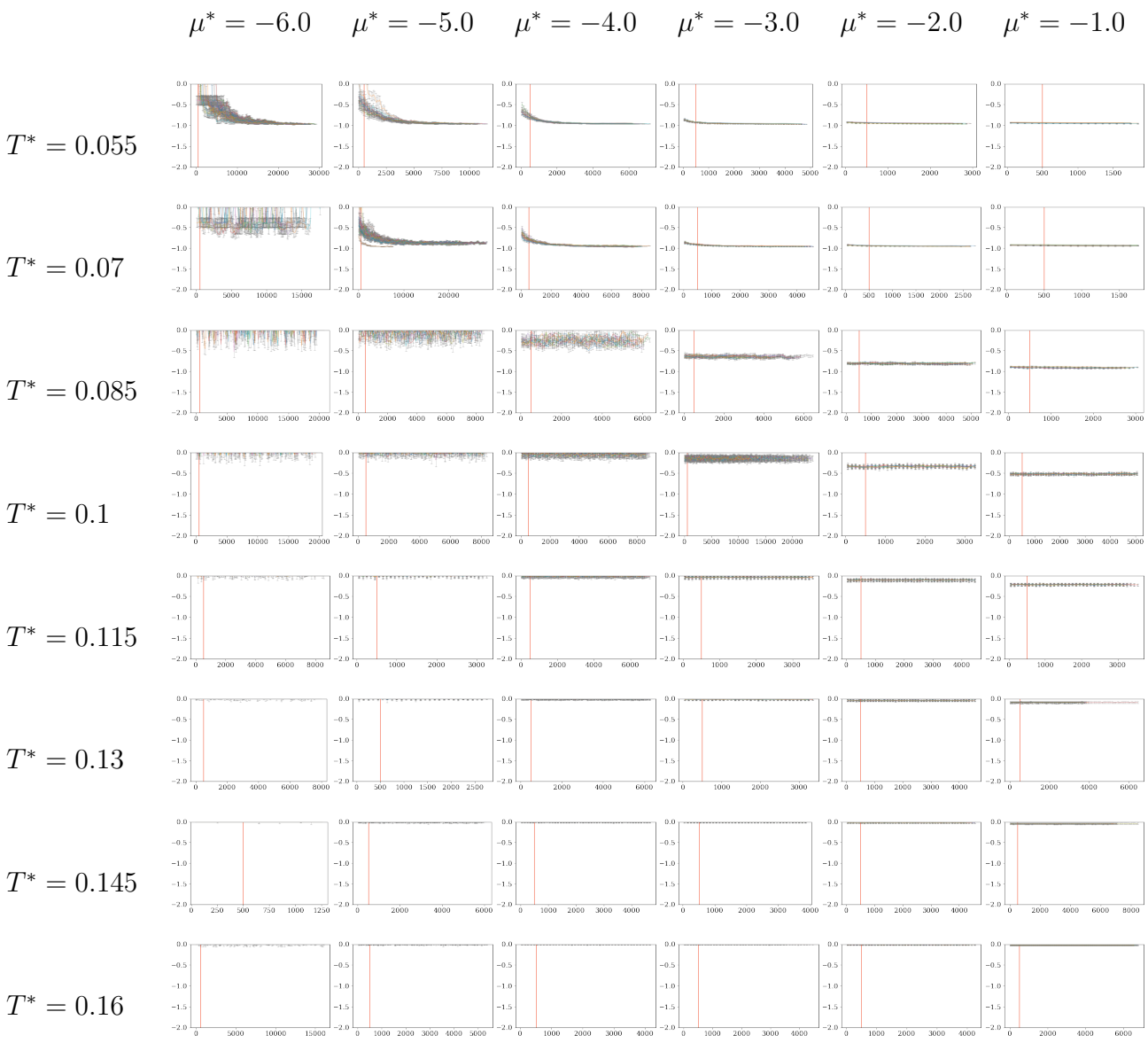


Figure B.11.: Energy per Particle trajectories for EPP with $\theta = 15°$. For temperatures (rows) and chemical potential (columns). Monte-Carlo time on the x-axes is given in units of 2000 cycles. The energy on the y-axes is given in units of the square-well depth ϵ of the patch potential and the particle number N.

$\mu^* = 0.0$ $\mu^* = 1.0$ $\mu^* = 2.0$ $\mu^* = 3.0$ $\mu^* = 4.0$ $\mu^* = 5.0$ $\mu^* = 6.0$

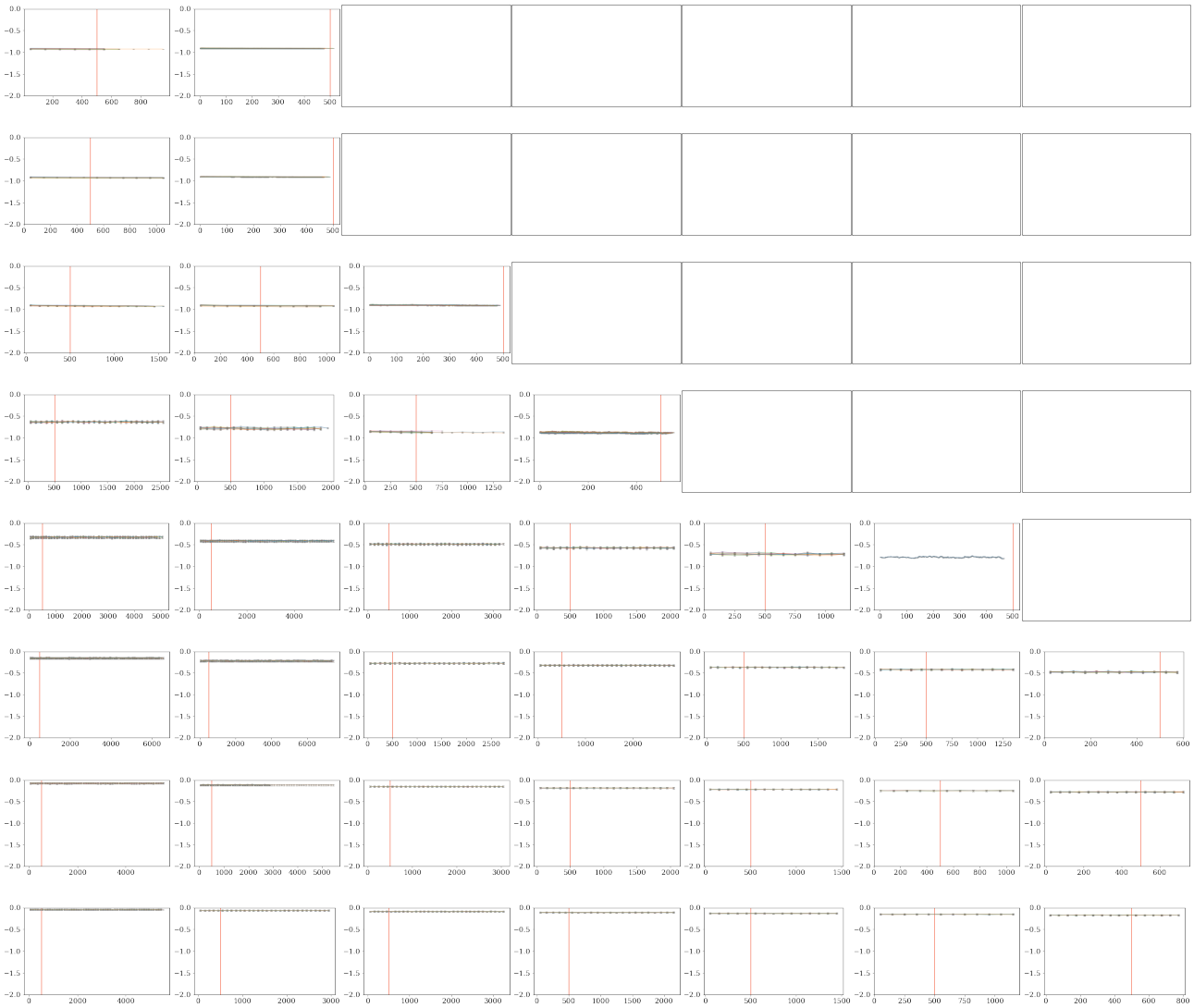


Figure B.12.: Continuation of energy trajectories for EPP with $\theta = 15°$ in appendix B.1.

Bibliography

[1] Daan Frenkel and Berend Smit. *Understanding Molecular Simulation*. Academic Press, 2001, p. 664.

[2] M. P. Allen and D. J. Tildesley. *Computer Simulation of Liquids*. Oxford University Press, 2017, p. 640.

[3] D. P. Landau and K. Binder. *A Guide to Monte Carlo Simulations in Statistical Physics*. Cambridge University Press, 2021, p. 564.

[4] Nicholas Metropolis et al. „Equation of State Calculations by Fast Computing Machines“. In: *The Journal of Chemical Physics* 21.6 (June 1953), pp. 1087–1092.

[5] Guido Van Rossum and Fred L Drake. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009.

[6] Wes McKinney et al. „Data structures for statistical computing in python“. In: *Proceedings of the 9th Python in Science Conference*. Vol. 445. Austin, TX. 2010, pp. 51–56.

[7] Charles R Harris et al. „Array programming with NumPy“. In: *Nature* 585 (2020), pp. 357–362.

[8] Aric Hagberg, Pieter Swart, and Daniel S Chult. *Exploring network structure, dynamics, and function using NetworkX*. Tech. rep. Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.

[9] Stephen Whitelam and Phillip L. Geissler. „Avoiding unphysical kinetic traps in Monte Carlo simulations of strongly attractive particles“. In: *The Journal of Chemical Physics* 127.15 (Oct. 2007), p. 154101.

[10] A. D. Bruce, N. B. Wilding, and A. J. Ackland. „Free energy of crystalline solids: A lattice-switch monte carlo method“. In: *Physical Review Letters* 79.16 (1997), pp. 3002–3005.

[11] A. D. Bruce et al. „Lattice-switch Monte Carlo method“. In: *Phys. Rev. E* 61.1 (2000), pp. 906–919.

[12] N. B. Wilding and A. D. Bruce. „Freezing by Monte Carlo Phase Switch“. In: *Physical Review Letters* 85.24 (Dec. 2000), p. 5138.

[13] Stephen Whitelam. „Approximating the dynamical evolution of systems of strongly interacting overdamped particles“. In: *Molecular Simulation* 37.7 (June 2011), pp. 606–612.

[14] Linge Bai and David Breen. „Calculating Center of Mass in an Unbounded 2D Environment“. In: *Journal of Graphics Tools* 13.4 (2008), pp. 53–60.

[15] Robin J. Wilson. *Introduction to Graph Theory*. Fourth Edi. Longman Group Ltd, 1996.

[16] H. Flyvbjerg and H. G. Petersen. „Error estimates on averages of correlated data“. In: *The Journal of Chemical Physics* 91.1 (1989), pp. 461–466.

[17] John Gurland and Ram C Tripathi. „A Simple Approximation for Unbiased Estimation of the Standard Deviation“. In: *The American Statistician* 25.4 (1971), pp. 30–32.

[18] Carlos Vega and Eva G. Noya. „Revisiting the Frenkel-Ladd method to compute the free energy of solids: The Einstein molecule approach“. In: *The Journal of Chemical Physics* 127.15 (Oct. 2007), p. 154113.

[19] Eva G. Noya, Carlos Vega, and Enrique de Miguel. „Determination of the melting point of hard spheres from direct coexistence simulation methods“. In: *The Journal of Chemical Physics* 128.15 (Apr. 2008), p. 154507.

[20] Daan Frenkel and Anthony J. C. Ladd. „New Monte Carlo method to compute the free energy of arbitrary solids. Application to the fcc and hcp phases of hard spheres“. In: *J. Chem. Phys* 81.August 1998 (1984).

[21] E.M. Landau, L.D. and Lifshitz. *Statistical physics*. Pergamon Press, Oxford., 1958.

[22] E G Noya, M M Conde, and C Vega. „Computing the free energy of molecular solids by the Einstein molecule approach: Ices XIII and XIV, hard-dumbbells and a patchy model of proteins Polymorphic transitions in single crystals: A new molecular dynamics method Computing the free energy of mole“. In: *The Journal of Chemical Physics* 1291.10 (2008).

[23] Eva G. Noya and Noé G. Almarza. „Entropy of hard spheres in the close-packing limit“. In: *Molecular Physics* 113.9-10 (May 2015), pp. 1061–1068.

[24] W. G. et al. *Numerical Recipes in Pascal–The Art of Scientific Computing.* Vol. 56. 193. 1991, p. 396.

[25] Herbert H.H. Homeier and E. Otto Steinborn. „Numerical integration of functions with a sharp peak at or near one boundary using möbius transformations“. In: *Journal of Computational Physics* 87.1 (1990), pp. 61–72.

[26] Alastair D. Bruce and Nigel B. Wilding. „Computational strategies for mapping equilibrium phase diagrams“. In: *Advances in Chemical Physics* 127.1 (2004), pp. 1–64.

[27] J. M. Polson et al. „Finite-size corrections to the free energies of crystalline solids“. In: *Journal of Chemical Physics* 112.12 (Mar. 2000), pp. 5339–5342.

[28] „Theory of Simple Liquids“. In: *Theory of Simple Liquids (Fourth Edition)*. Ed. by Jean-Pierre Hansen and Ian R. McDonald. Fourth Edition. Oxford: Academic Press, 2013, p. i.

[29] A Saupe. „Recent Results in the Field of Liquid Crystals“. In: *Angewandte Chemie International Edition in English* 7.2 (1968), pp. 97–112.

[30] R. Eppenga and D. Frenkel. „Monte Carlo study of the isotropic and nematic phases of infinitely thin hard platelets“. In: *Molecular Physics* 52.6 (1984), pp. 1303–1334.

[31] Eric W. Weisstein. *Log Normal Distribution*.

[32] Carina Karner, Christoph Dellago, and Emanuela Bianchi. „Hierarchical self-assembly of patchy colloidal platelets“. In: *Soft Matter* 16.11 (2020), pp. 2774–2785.

[33] B. J. Alder and T. E. Wainwright. „Phase Transition for a Hard Sphere System“. In: *The Journal of Chemical Physics* 27.5 (Aug. 2004), p. 1208.

[34] William G. Hoover and Francis H. Ree. „Melting Transition and Communal Entropy for Hard Spheres“. In: *The Journal of Chemical Physics* 49.8 (1968), pp. 3609–3617.

[35] Lars Onsager. „the Effects of Shape on the Interaction of Colloidal Particles“. In: *Annals of the New York Academy of Sciences* 51.4 (1949), pp. 627–659.

[36] Daan Frenkel. „Order through entropy“. In: *Nature* 14 (2015), pp. 9–12.

[37] Jacques Vieillard Baron. „Phase Transitions of the Classical Hard-Ellipse System“. In: *The Journal of Chemical Physics* 56.10 (1972), pp. 24501–10034.

[38] JA Cuesta and D. Frenkel. „Monte Carlo simulation of two-dimensional hard ellipses“. In: *Physical Review A* 42.4 (1990), p. 2126.

[39] D. A. Ward and F. Lado. „Structure, thermodynamics, and orientational correlations of the nematogenic hard ellipse fluid from the percus-yevick equation“. In: *Molecular Physics* 63.4 (1988), pp. 623–638.

[40] S. Davatolhagh and S. Foroozan. „Structural origin of enhanced translational diffusion in two-dimensional hard-ellipse fluids“. In: *Physical Review E* 85.6 (June 2012), p. 061707.

[41] Alan M. Luo, Leonard M. C. Sagis, and Patrick Ilg. „The Landau free energy of hard ellipses obtained from microscopic simulations“. In: *The Journal of Chemical Physics* 140.12 (Mar. 2014), p. 124901.

[42] Gustavo Bautista-Carbajal and Gerardo Odriozola. „Phase diagram of two-dimensional hard ellipses“. In: *The Journal of Chemical Physics* 140.20 (May 2014), p. 204502.

[43] Robert H. Swendsen and Jian-Sheng Wang. „Replica Monte Carlo Simulation of Spin-Glasses“. In: *Phys. Rev. Lett.* 57 (21 Nov. 1986), pp. 2607–2609.

[44] Jacques Vieillard-Baron. „Les Transitions de Phase du Systeme des Ellipses Dures Classique“. PhD thesis. 1970.

[45] Xiaoyu Zheng and Peter Palffy-Muhoray. „Distance of closest approach of two arbitrary hard ellipses in two dimensions“. In: *Physical Review E* 75.6 (June 2007), p. 061709.

[46] Wagner S.; Kahl G.; Baumketner A. „to be published“. In: (2022).

[47] Sebastian Schön, Susanne Wagner, and Gerhard Kahl. „to be published“. In: (2022).

[48] W Bol. „Monte Carlo simulations of fluid systems of waterlike molecules“. In: *Molecular Physics* 45.3 (1982), pp. 605–616.

[49] Jiří Kolafa and Ivo Nezbeda. „Monte carlo simulations on primitive models of water and methanol“. In: *Molecular Physics* 61.1 (1987), pp. 161–175.

[50] William R. Smith and Ivo Nezbeda. „A simple model for associated fluids“. In: *The Journal of Chemical Physics* 81.8 (1984), pp. 3694–3699.

[51] Larry W. Dahl and Hans C. Andersen. „Cluster expansions for hydrogen-bonded fluids. III. Water“. In: *The Journal of Chemical Physics* 78.4 (1983), pp. 1962–1979.

[52] C. Casagrande et al. „«Janus beads»: Realization and behaviour at water/oil interfaces“. In: *Epl* 9.3 (1989), pp. 251–255.

[53] H Takei and N Shimizu. „Gradient Sensitive Microscopic Probes Prepared by Gold Evaporation and Chemisorption on Latex Spheres“. In: *Langmuir* 13.7 (1997), pp. 1865–1868.

[54] Amar B. Pawar and Ilona Kretzschmar. „Fabrication, assembly, and application of patchy particles“. In: *Macromolecular Rapid Communications* 31.2 (2010), pp. 150–168.

[55] Gi-Ra Yi, David J Pine, and Stefano Sacanna. „Recent progress on patchy colloids and their self-assembly“. In: *J. Phys.: Condens. Matter* 25.25 (2013), pp. 193101–12.

[56] Young Sang Cho et al. „Particles with coordinated patches or windows from oil-in-water emulsions“. In: *Chemistry of Materials* 19.13 (2007), pp. 3183–3193.

[57] Daniela J. Kraft, Jan Groenewold, and Willem K. Kegel. „Colloidal molecules with well-controlled bond angles“. In: *Soft Matter* 5.20 (Oct. 2009), p. 3823.

[58] Hailin Cong et al. „Current status and future developments in preparation and application of colloidal crystals“. In: *Chemical Society Reviews* 42.19 (Sept. 2013), p. 7774.

[59] Michael A. Boles, Michael Engel, and Dmitri V. Talapin. „Self-Assembly of Colloidal Nanocrystals: From Intricate Structures to Functional Materials“. In: *Chemical Reviews* 116.18 (Sept. 2016), pp. 11220–11289.

[60] F. Meseguer. „Colloidal crystals as photonic crystals“. In: *Colloids and Surfaces A: Physicochemical and Engineering Aspects* 270-271 (Dec. 2005), pp. 1–7.

[61] Hanbin Zheng and Serge Ravaine. „Bottom-Up Assembly and Applications of Photonic Materials“. In: *Crystals* 6.5 (May 2016), p. 54.

[62] Xi Wang et al. „Recent advances in solution-processed inorganic nanofilm photodetectors“. In: *Chemical Society Reviews* 43.5 (2014), pp. 1400–1422.

[63] Johannes Loehr et al. „Colloidal topological insulators“. In: *Communications Physics* 1.1 (Dec. 2018), p. 4.

[64] Marjolein Dijkstra and Erik Luijten. „From predictive modelling to machine learning and reverse engineering of colloidal self-assembly“. In: *Nature Materials* 20.6 (2021), pp. 762–773.

[65] Sharon C Glotzer and Michael J Solomon. „Anisotropy of building blocks and their assembly into complex structures“. In: *Nature Materials* 6 (2007), pp. 557–562.

[66] Emanuela Bianchi, Ronald Blaak, and Christos N. Likos. „Patchy colloids: State of the art and perspectives“. In: *Physical Chemistry Chemical Physics* 13.14 (2011), pp. 6397–6410.

[67] Bo Li, Di Zhou, and Yilong Han. *Assembly and phase transitions of colloidal crystals*. 2016.

[68] Emanuela Bianchi et al. „Inverse patchy colloids: Synthesis, modeling and self-organization“. In: *Current Opinion in Colloid and Interface Science* 30 (2017), pp. 8–15.

[69] Carina Karner, Christoph Dellago, and Emanuela Bianchi. „Design of Patchy Rhombi: From Close-Packed Tilings to Open Lattices“. In: *Nano Letters* 19.11 (Nov. 2019), pp. 7806–7815.

[70] Jaime A Millan, Daniel Ortiz, and Sharon C Glotzer. „Effect of shape on the self-assembly of faceted patchy nanoplates with irregular shape into tiling patterns“. In: *Soft Matter* 11 (2015), pp. 1386–1396.

[71] Qian Chen, Sung Chul Bae, and Steve Granick. „Directed self-assembly of a colloidal kagome lattice“. In: *Nature Letter* 469 (2011), pp. 381–385.

[72] Flavio Romano and Francesco Sciortino. „Two dimensional assembly of triblock Janus particles into crystal phases in the two bond per patch limit“. In: *Soft Matter* 7 (2011), pp. 5799–5804.

[73] Basavaraj Madivala, Jan Fransaer, and Jan Vermant. „Self-assembly and rheology of ellipsoidal particles at interfaces“. In: *Langmuir* 25.5 (2009), pp. 2718–2728.

[74] Lei Tian et al. „Large-scale fabrication of polymer ellipsoids with controllable patches via the viscosity-induced deformation of spherical particles“. In: *Polymer Chemistry* 8.25 (June 2017), pp. 3774–3777.

[75] Susanne Wagner and Gerhard Kahl. „Structure and equation-of-state of a disordered system of shape anisotropic patchy particles“. In: *Molecular Physics* 117.20 (Feb. 2019), pp. 2873–2880.

[76] Norbert Kern and Daan Frenkel. „Fluid–fluid coexistence in colloidal systems with short-ranged strongly directional attraction“. In: *The Journal of Chemical Physics* 118.21 (June 2003), pp. 9882–9889.

[77] H.M. Cundy and A.P. Rollett. *Mathematical Models*. Oxford Clarendon Press, 1961.

[78] Richard P. Sear. „Nucleation: Theory and applications to protein solutions and colloidal suspensions“. In: *Journal of Physics Condensed Matter* 19.3 (2007).