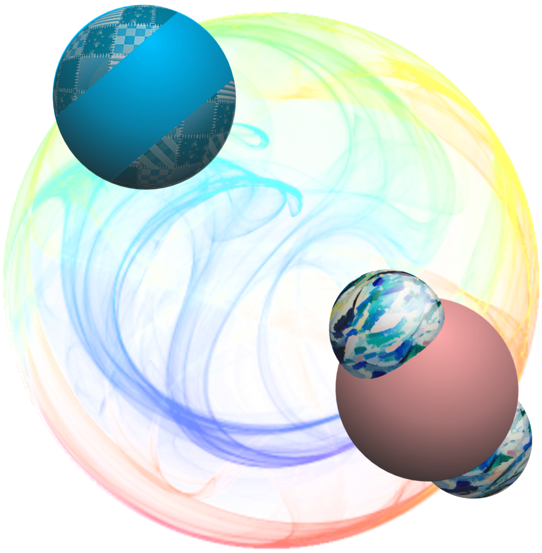
Predicting crystal structures of patchy colloids

Jeroen Schot

July 26, 2012



Supervisors:
Dr. Teun Vissers
Prof. dr. ir. Marjolein Dijkstra

Soft Condensed Matter Group
Debye Institute for Nanomaterials Science
Faculty of Science
Utrecht University

Universiteit Utrecht

Abstract

Recently, much progress has been made on the synthesis of colloidal particles with anisotropic interactions, also called patchy colloids. These particles can self-assemble in interesting structures, but it is difficult to predict the structures that form for a given particle shape and interaction. This challenges us to develop tools to predict the structures of patchy colloids, such as crystals. In this thesis we describe a method to generate likely crystal structures for patchy colloids and the methods we use to examine these crystal structures. We present results of applying this method on two types of patchy colloids. We show that changing the size and shape of the attractive patches results in crystal structures with different density and particle orientation.

Contents

Chapter 1

Introduction

1.1 Colloids

The modern scientific history of colloids starts with the discovery of Brownian motion: the random drifting of colloidal particles. This behaviour was first observed by Robert Brown in 1827 while studying pollen particles under a microscope [1]. Solvent molecules collide with the suspended particles. Because of the small size of the colloids these random collisions do not average out, but exercise a small net force in random directions, causing the diffusive motion.

It was Thomas Graham who in 1861 first used the term 'colloid' [2]. He derived the word from the Greek $\kappa o\lambda\lambda\alpha$ meaning glue, due to the glue-like properties of some colloidal systems. The first physical understanding of this behaviour in terms of statistical fluctuations was presented in 1905 by Albert Einstein [3] and parallel by William Sutherland [4]. These theoretical studies were later verified by the experiments of Jean Baptiste Perrin [5] studying the sedimentation equilibrium of monodisperse systems.

Colloidal particles have sizes between a few nanometers and several micrometers, which means they are larger than atoms and molecules, but they are still microscopic objects. They are typically found dispersed in a continuous molecular medium. The colloidal suspensions and medium can be in different phases. A solid medium with a liquid dispersion is called a gel. Other examples include liquid-liquid (emulsions such as milk), liquid-solid (sols such as blood and paint) and gas-liquid (aerosol such as mist).

Their size puts colloids in an interesting niche. On one side, they are big enough to be easily studied with conventional bright-field light microscopy (like Robert Brown) or the more advanced confocal laser microscopy. On the other side, they are still so small that their thermodynamic properties are similar to that of atoms and molecules. Moreover, the larger size of colloids allows also for manipulation by external fields such as gravity and electric fields.

Early studies of colloids focussed on of systems consisting of hard spheres, and systems with isotropic interactions such as screened-Coulomb interactions or depletion interactions. Crystallisation of systems of spheres was demonstrated theoretically in 1957 [6, 7]. Experimental results with colloidal spheres of polymethylmethacrylate where realised in 1986 [8]. The phase behaviour of systems with electrostatic interactions can be controlled by tuning the interaction strength [9, 10].

The recent years have shown many advances in the synthesis of colloids with a variety of shapes, which promise to be de building blocks for novel materials [11]. We have seen the controlled synthesis of both convex particles such as colloidal rods [12], cubes [13], superballs [14] and tetrahedra [15] as well as octapods [16] and caps [17].

These developments have spurred theoretical and simulation studies of the phase behaviour for systems of anisotropic hard particles. These systems are only driven by entropy, and particle shape has shown to be a important role in determining equilibrium structures.

For example, cubes are found to form simple cubatic crystals, but with a high vacancy concentrations [18]. Cubes with rounded corners can give preference to rhomic crystals and rotator phases [19]. Octapods show hierarchical self-assembly by forming interlocked chains that self-assemble into a three-dimensional superstructure [16].

1.2 Patchy colloids

In recent years methods have been developed to chemically and physically pattern surfaces of colloidal particles in a controlled way. Usually this patterning is in the form of distinct patches, leading to the name *patchy colloids*. The addition of patches typically results in interactions that are very isotropic and directional in nature and offers unique possibilities for self-organisation and assembly [20].

Theoretical models of anisotropic interactions were first developed in the 1980's to describe associating fluids [21]. These models use spherical particles with attractive sites. An analytical theory for the thermodynamic properties of these systems was developed by Wertheim [22]. Two important conditions in Wertheim theory are the single bonding condition and the absence of bonding loops.

Later other models for directional interactions where developed with colloidal systems in mind. The sticky spot model [23] also focusses on single bonds. The Kern-Frenkel model [24] is a general model for patches with varying size on the surface of a sphere. Patch-antipatch models [25] are used to represent specific protein interactions.

There are many different ways to get patchy interactions. In the earliest work patchy colloids were created by clustering of particles of different materials [26, 27]. Directional bonding can be introduced by selection of the materials and the size ratio of the particles. It is also possible to create site-specific bonds between particles by functionalizing patches with lock and key groups, by for example strands of DNA [28, 29].

Another method that has been used successfully is the coating of polymer spheres with metallic patches. Here techniques such as glancing angle deposition [30] are used to create patches with fine control on the size and relative position.

As a specific case, spheres with a single patch that covers an entire hemisphere are known as Janus spheres, after the two-faced Roman god [31]. Simulations have shown interesting phase behaviour with cluster phases of micelles [32, 33]. Spheres with two patches on opposite sides are called (confusingly) triblock Janus spheres or simply triblock spheres. For a specific patch size, assembly triblock spheres into a kagome lattice was observed [34]. These results were reproduced in simulations on quasi-two-dimensional systems [35]. Further work also looked at the effect of patch size and number in three dimensions [36].

1.3 Predicting crystal structures of patchy colloids

So far, existing simulation work on the crystal structures of patchy colloids has focussed on preselected structures. This selection can be based on assumptions on correspondence between patch geometry and crystal lattice. A general method to generate likely equilibrium structures is not yet available.

In this thesis we present such a method show results of applying it to two different patchy particle systems. This method is based on the work of Filion et al. [37] for hard-particle systems. We have adapted this for systems with anisotropic interactions and looked at ways to analyze our simulation data.

The method generates crystal unit cells by running variable box shape simulations on systems with only a small number of particles. These are Monte Carlo simulations in the isobaric-isothermal ensemble where the system volume is varied by changing both the length and relative orientation of the box vectors.

From these unit cells we construct candidate crystal structures. We then again perform Monte Carlo simulations on these candidates and monitor the systems for changes in the internal energy and the orientational order to see if these crystal structure melt or crystallize in another configuration. This all is done in the regime of low temperature and pressure, because this is where the patchy interactions will play an important role.

We have applied this method to systems of triblock spheres and a similar, type of patchy particle, but non-spherical in shape. This particle consists of a hard-core sphere with two spherical patchy protrusions. The patches exhibit an attractive interaction which we model with a square-well potential. One can think of this particle as the simplest anisotropic extension of the triblock sphere. We believe that this simple model is a good way to study the interplay of shape and interaction anisotropy.

Chapter 2

Model

There are a number of different patchy particle models. In this thesis we used the so-called Kern-Frenkel model [24], that we briefly describe below. This is a general model for a directional square-well potential of spheres. Other well known models are the sticky spot model [23], and the patch-antipatch model [25].

First we describe the original Kern-Frenkel model. Then we present a possible extension to the Kern-Frenkel model to adapt it to spheres with spherical protruding patches.

2.1 Kern-Frenkel

The Kern-Frenkel model was made to represent strongly directional interactions. In this model particles are impenetrable spheres. Part of the surface of these spheres functions as attractive patches. When two patches of different spheres are close enough and aligned they interact. This results in a negative energy contribution. We call the patches "aligned" when the vector joining the center of mass of the spheres intersects both patches. The pair potential between two patches is defined as the product of the square-well potential with the angle as

$$u_{ij}(r_{ij};\Omega_i,\Omega_j) = u_{ij}^{hssw}(r_{ij}) \cdot f_{ij}(r_{ij};\Omega_i,\Omega_h). \tag{2.1}$$

The square-well potential depends on the radial distance as

$$u_{ij}^{hssw}(r_{ij}) = \begin{cases} \infty \text{ for } \|r_{ij}\| < \sigma \\ -\epsilon \text{ for } \sigma \leq \|r_{ij}\| < r_c \\ 0 \text{ for } \|r_{ij}\| \geq r_c. \end{cases} \tag{2.2}$$

Here σ is the sphere diameter and r_c is the square-well range.

The shape of the patches is modelled by the intersection of the particle surface by a cone constructed from an angle around a vector $\hat{e}_a$ from the center of the sphere, where $\hat{e}_a$ denotes the unit vector for patch a. The angle is the half-open angle δ. This patch definition is illustrated in figure 2.1.

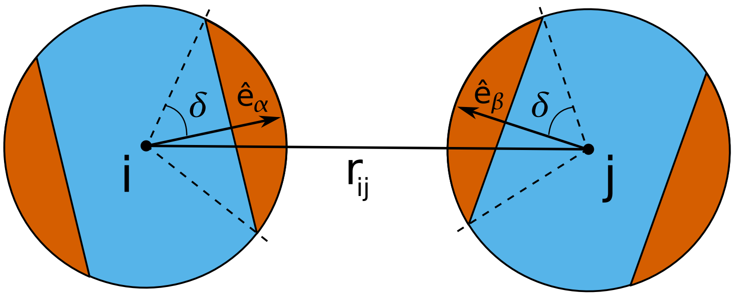


Figure 2.1: Schematic of the Kern-Frenkel model. The two spheres are at center-of-mass distance $\|\vec{r}_{ij}\| = |\vec{r}_i - \vec{r}_j|$ apart. The cone of attraction around the orientation vectors $\hat{e}_a$ and $\hat{e}_b$ is determined by the angle δ and depicted by the dotted lines.

$$f_{ij}(r_{ij};\Omega_i,\Omega_h) = \begin{cases} 1 & \text{if } \hat{e}_\alpha \cdot \hat{r}_{ij} \geq \cos\delta \text{ for some patch } \alpha \text{ on } i \\ & \text{and } \hat{e}_\beta \cdot \hat{r}_{ij} \geq \cos\delta \text{ for some patch } \beta \text{ on } j \\ & \text{or } \|r_{ij}\| < \sigma, \\ 0 & \text{otherwise} \end{cases} \tag{2.3}$$

This model allows for any number of patches in any configuration. Particles with a single patch are called 'Janus spheres'.

In our simulations we focus on spheres with two patches of the same size on opposite sides. These specific particles are known in the literature as 'triblock spheres' or the somewhat misleading 'triblock Janus spheres'.

2.2 Spherical protrusions

The Kern-Frenkel model was originally designed for spherical particles with patches on their surface. Because we are also interested in shape-anisotropic particles we must adapt it for this case. Here we want our model to capture the effects of shape while not sacrificing the simplicity.

The particles we studied are spherical and have two spherical protrusions. We can think of one such particle as a set of overlapping spheres. The center sphere is then a hard sphere, the attractive protrusions are Janus spheres with the non-overlapping part being the patch. The orientation vector of the patch points away from the center sphere. The half-open angle is defined by the intersection with the center sphere.

The pair potential is the same as in the regular Kern-Frenkel model. A schematic overview of these particles with patchy protrusions can be seen in figure 2.2.

The half open angle of the patches is no longer a parameter that can be chosen freely, but depends on the geometry of the spheres: the ratio of the core sphere and patchy sphere diameters (σ_c and σ_p) and the center-to-center distance (L) as

$$\delta = \pi - \arccos\left(\frac{\sigma_p^2 + 4L^2 - \sigma_c^2}{4\sigma_p L}\right). \tag{2.4}$$

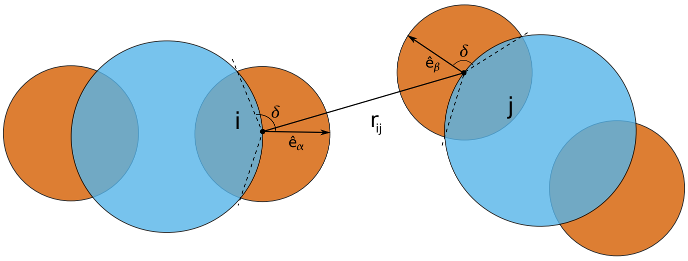


Figure 2.2: Schematic of the Kern-Frenkel model adapted for patchy colloids. The orientation vectors are now relative to the center of the spherical protrusion, at distance L of the hard-sphere core.

Using this method one can construct many different particles by varying the size, shape, number and position of the patchy spheres. We chose particles with two patchy spheres on opposite sides. We fixed the center-to-center distance to $L = \sigma_c/2$ (e.g. on the surface of the core sphere).

2.3 Volume of overlapping spheres

To calculate the density of our system we need to know the volume of a single particle. Our particles consist of overlapping spheres. We determine the volume of a single particle by taking the sum of the separate spheres and subtract the overlap:

$$V = V_c + 2V_p - V_{overlap}. \tag{2.5}$$

The volume of the spheres is easily calculated. The volume of the overlap region is slightly more complicated. Consider two spheres with radius r_a and r_b at $(0,0,0)$ and $(r_a,0,0)$. The surfaces of the two spheres intersect at

$$x = \frac{r_a^2 + r_b^2}{2r_b} \tag{2.6}$$

$$y^2 + z^2 = r_a^2. \tag{2.7}$$

The overlapping volume of the two consists of two spherical caps [38] with a base at x and height

$$h_a = r_a - \frac{r_b^2 + r_a^2}{2r_b} \tag{2.8}$$

$$h_b = r_b - r_a + \frac{r_b^2 + r_a^2}{2r_b}. \tag{2.9}$$

The volume of a spherical cap is given by

$$\frac{1}{3}\pi h^2 \left(3r - h\right). \tag{2.10}$$

Solving this for the two caps gives

$$V_a + V_b = \pi r_b^3 \left(\frac{8r_a - 3r_b}{12r_b} \right). \tag{2.11}$$

Combining this with equation (2.5), the volume of a single particle is

$$V = \frac{4}{3}\pi(r_c^3 + 2r_p^3) - 2\pi r_p^3 \left(\frac{8r_c - 3r_p}{12r_p} \right). \tag{2.12}$$

Chapter 3

Methods

3.1 Generating candidate structures

In order to determine the phase behavior for a system of patchy particles, we should first predict the candidate crystal structures. We are primarily interested in the regime where the anisotropic interaction is dominant, not the entropy or packing. We thus focus on a regime of low temperature and low pressure.

The problem we are facing is that there is a large number of possible colloidal crystal structures. Because of this, enumeration is not a viable option and using manually constructed crystals suffers from selection bias. Therefore we use a simulation method called "variable shape unit cell Monte Carlo" [37], sometimes also named "floppy box MC".

This method consists of a number of steps. A schematic overview of method is given in figure 3.1. We start by performing isobaric-isothermal (NPT) simulations for systems with a small number of particles. Typically only two to twelve. This system is basically the unit cell of our candidate structure. For every number of particles we run five simulations with different starting configurations. From these we select the final configuration with the lowest energy.

From each of these candidate structures we generate a colloidal crystal by multiplying the small system that resembles a possible unit cell a number of times in all directions until we obtain a system with a number of particles between 200 and 500.

For this large system we run a new NPT simulation. Here we look at the stability of the candidate by monitoring the density, energy and orientation distribution of the particles. When the crystal is not stable there will be a phase transition to a fluid or a different crystal.

This method was originally designed for systems with a high entropic contribution to the free energy such as hard-particle systems. Therefore normally a simulated annealing step is performed in the small floppy box: the pressure is gradually increased to compress the system. In our case this is not desirable because we do not expect our crystal structures to be stable in the high pressure limit.

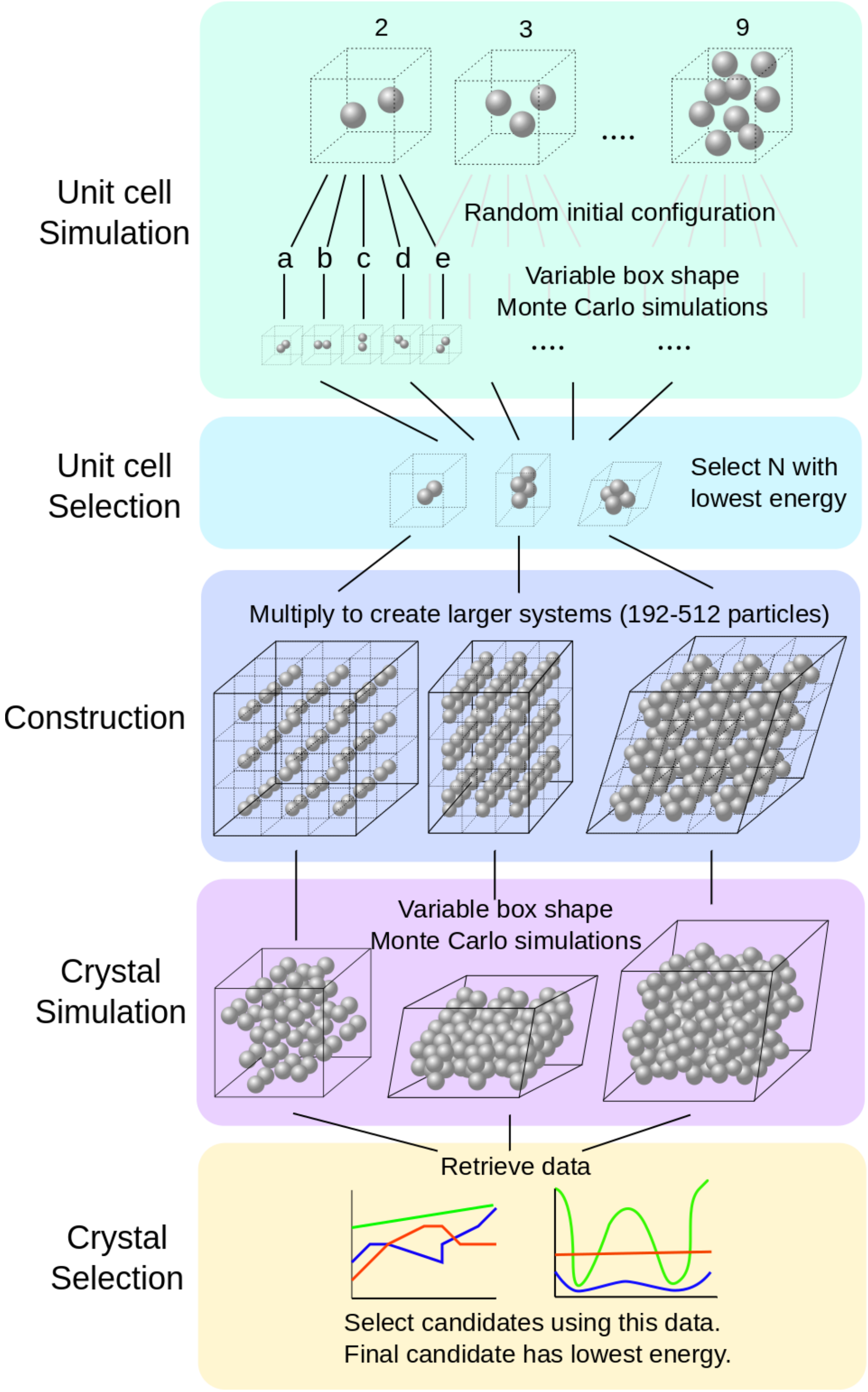


Figure 3.1: Schematic overview of the method showing the different steps we use to generate probably candidate crystal structures.

3.2 Statistical physics

Statistical physics allows us to study the macroscopic properties of systems with a large number of degrees of freedom such as many-particle systems. These systems have a number of different contributing states with different energies. Gibbs derived that for a system in equilibrium the probability of being in state i is given by

$$P_i = \frac{e^{-E_i/kT}}{\sum_j e^{-E_j/kT}}, \tag{3.1}$$

where E_i is the energy of state i and k is Boltzmann's constant. This probability distribution is known as the Boltzmann distribution. The summation in the fraction is called the partition function and denoted by Z.

Using this distribution we can define a expectation value for a macroscopic quantity A of our system

$$\langle A \rangle = \frac{1}{Z} \sum_i A_i e^{-E_i/kT}. \tag{3.2}$$

We can estimate this expectation value by generating a number of independent system states from this distribution and take the average of their values. This is what we are going to use in our Monte Carlo method.

3.3 Monte Carlo method

The Monte Carlo (MC) method is a computational tool to estimate the macroscopic quantities of a system. This is accomplished by random sampling of the phase space. Since its invention in 1946, the method has been used successfully in many different research fields, ranging from biology to finance. The following explanation focusses on the use of the MC method in statistical physics.

The methods we are using to determine the thermodynamic properties in equilibrium of our systems is the Monte Carlo method. This method entails that we construct a new configuration from an existing old configuration by making a small change. This can either be the displacement or rotation of a single particle or the deformation of the simulation box. We accept this new configuration with a probability that depends on the Boltzmann or statistical weights of the old and new configurations. If the new configuration is not accepted, the old one is restored. By repeating this process many times we can measure the thermodynamic equilibrium properties of the system, such as volume, number of bonds, and the internal (interaction) energy.

In this process there are two important properties our method should have: *ergodicity* and *detailed balance*. Ergodicity means that from every possible starting configuration we can reach every other possible configuration (after many steps). Because of this it is important to consider every potential configuration with a chance larger than zero. Detailed balance means that in equilibrium the probability of going from configuration A to B is equal to the probability of the reverse.

The system consists of a three-dimensional volume with a number of particles (triblock spheres or spheres with patchy protrusions). The initial simulation box is cubic, but we allow shape changes to a more general parallelepiped shape. We apply periodic boundary conditions on the box: on every face we put an image copy of the system. When a particle moves out of the box during the simulation its position is updated to the periodic image that it entered. The simulations were performed in the NPT ensemble, which means a constant number of particles N, constant pressure P, and constant temperature T.

3.3.1 Particle moves

We distinguish two different moves where we modify the properties of one randomly selected particle: displacement and rotation moves. A number of algorithms exist that modify the positions (and orientations) of multiple particles in a single step. We did not use these so called "cluster algorithms" in our simulations.

In a displacement move we shift the center of mass of a particle by a random value in the range $[-\Delta x, \Delta x]$. The value of Δx is adjusted in the first part of the simulation such that about 30% of the moves is accepted. In a rotation move a particle is rotated around its center of mass.

For rotation moves we use the strategy from [39]. Every particle has a orientation unit vector $\hat{u}$. To vary this vector we first construct a random unit vector $\hat{v}$ and multiply this by a scale factor γ. We get a new (non-unit) vector $\vec{t}$ by adding $\hat{u} + \gamma\hat{v}$. Our new unit vector is then normalized $\hat{t} = \vec{t}/\|\vec{t}\|$.

3.3.2 Box moves

In the NPT ensemble we use two box moves to modify the volume and shape of the system: resizing and rotating. The simulation box is parameterized by three box or lattice vectors $\vec{L}_x$, $\vec{L}_y$, $\vec{L}_z$. These vectors span a parallelepiped, whose volume is given by $\vec{L}_x \cdot (\vec{L}_y \times \vec{L}_z)$. The two box moves modify one of these three vectors.

The resizing move changes the length of one of the vectors by multiplying it by $(X + 1)$, with X a random number in the range $[-\Delta V, \Delta V]$. (The $+1$ term is needed to satisfy detailed balance). This move changes only the length of the vector, not the orientation.

The box rotation move slightly rotates one of the lattice vectors. This is done in a way similar to the rotation move for single particles. The important difference is that in the last step the lattice vector is not normalized to unity, but to the initial length before the move. Thus this move changes only the orientation of the vector, and not its length. We note that the volume of the simulation box changes in both box moves. Figure 3.2 shows a possible move of both types.

None of the lattice vectors are fixed to any of the Euclidean axes or planes. This means that the simulation box rotates freely. This can make identification of different systems difficult. To prevent collapse of the box we reject unphysical distorted lattice vectors. We do this by rejecting angles between the lattice vectors smaller than 30 degrees and larger than 150 degrees.

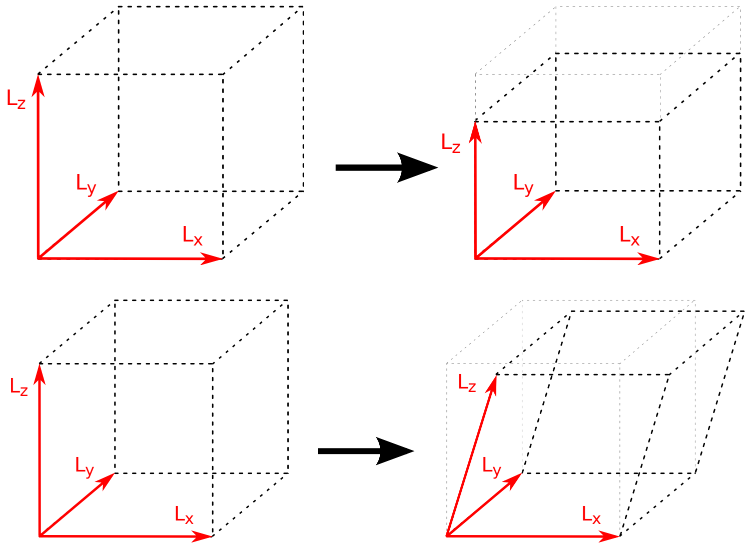


Figure 3.2: Example of the two different box moves. At the top we see a resize move that decreases the length of lattice vector $\vec{L_z}$, shrinking the box volume and making the box less cubic. Below that the box rotation move rotates the lattice vector $\vec{L_z}$, thus deforming the box.

3.3.3 Acceptance rules

The probability to go from one configuration to another is called the acceptance probability and is determined by the acceptance rules. We have used the Metropolis choice, which results in the following rules:

$$P_{particle} = \min(1, \exp\left(-\beta(U' - U)\right)) \tag{3.3}$$

$$P_{box} = \min(1, \exp\left(-\beta(U' - U) + P(V' - V) - N\beta^{-1} ln\left(V'/V\right)\right)). \tag{3.4}$$

Here U' and V' denote the energy and volume of the new configuration.

3.4 Distribution function

To identify and classify the resulting structures one can use distribution functions. We have used the orientational distribution function, which provides us information about the average relative orientation of the particles and their patches.

Because our focus here is on anisotropic particles, we study the patch orientation distribution. This distribution shows the density as a function of the relative angle of patches on the particles.

We calculate this distribution by taking the inner product of the orientation unit vectors of every patch with every patch on all other particles. The resulting value is the cosine of the angle between the unit vectors, which we bin in a histogram. The histogram is normalized to unit area.

The domain of the resulting histogram is $[-1, 1]$. Because our particles have two patches on opposite sides, the histogram is plane symmetric around the value 0. For a system in a fluid phase there is almost zero orientation preference, and thus the patch orientation distribution is a flat line. Crystal structures will show Gaussian peaks around -1 and 1 (particles with the same alignment) and possibly other angles.

3.5 Implementation

A few things are different from a normal hard-sphere MC simulation. We note here some of the details of the implementation that might not be obvious. A major part of the simulation time is spent on calculating the potential energy and particle overlaps. Because of this it is wise to focus on doing this as efficient as possible. We also mention some details about the complications of combining cell lists with a variable box shape and periodic images in small systems. We have provided some code examples in appendix A.

3.5.1 Energy calculation

As explained, our particles exhibit both hard-particle and directed square-well interactions. We calculate both in a single go, but stop as soon as we encounter a particle overlap somewhere in the system.

The only shape we have to check for overlap is the sphere, but there are three per particles. For a pair this means nine checks. We try to minimize the number of checks by first checking the circumscribed spheres of the particles. We include the square-well range in the radius of this sphere. So when the circumscribed spheres do not overlap we are sure there is no interaction whatsoever between this pair. When they do overlap, we first check all nine pairs of spheres for overlap. If this is not the case, we check if the patchy spheres are within the square-well range and oriented.

3.5.2 Cell lists

When we move a single particle we have to check the interaction with every other particle for overlap and contribution to the energy. This means that a single particle will take simulation time $\sim N$, and a single MC step where every particle is moved on average once take simulation time $\sim N^2$. This is not a problem in our small simulations of two to nine particles, but for big systems we want to avoid this. We can do that using the fact that system has only short range interactions.

There are two well known tools to speed up the energy calculation: Verlet list [40] and cell lists. We have used cell lists for our simulations. The simulation box is divided into cells. The size of the cells should be just large enough such that a particle in a cell only interacts with particles in surrounding cells. No matter how large we make our system, we only have to check a total of 27 cells.

Chapter 4

Results

In this chapter we present the results of applying the method we described to two types of particles. These are triblock spheres and spheres with attractive protruding spheres. We will express the simulation parameters in dimensionless units, also called reduced units. These are the dimensionless pressure P^*, square-well depth ϵ^* and square-well range r_c^*. These are defined as

$$P^* = \frac{P\sigma_c^3}{\epsilon} \tag{4.1}$$

$$\epsilon^* = \frac{\epsilon}{kT} \tag{4.2}$$

$$r_c^* = \frac{r_c - \sigma_p}{\sigma_c}. \tag{4.3}$$

where σ_c and σ_p are the diameters of the center sphere and patchy sphere respectively. For the triblock spheres $\sigma_p = \sigma_c = \sigma$.

When comparing the internal energy of systems we use two different terms. The first is $U/(N\epsilon^*)$, the internal energy per particle divided by the square-well depth. We also use the *average number of bonds per patch*. Because all the particles we consider have two patches, the relation between the two terms is simply

$$\langle \text{no. bonds / patch} \rangle = -1 \cdot U/(N\epsilon^*). \tag{4.4}$$

4.1 Triblock spheres

Here we describe the results of applying our method to triblock spheres. We focussed on two different patch sizes: a half open angle δ of 0.6 rad and 0.9 rad. These correspond to a patch surface coverage $\chi = 2 \cdot \sin^2(\frac{\delta}{2})$ of approximately 0.17 and 0.38.

For both of those we ran a number of unit cell NPT simulations: we varied the number of particles from 2 till 9, where we performed five simulations with different starting conditions each. The system parameters were $P^* = 1$, $\epsilon^* = 3$, $r_c^* = 0.2$. For every run we recorded the lowest energy state the system reached, because we hypothesize that at low temperatures the crystals with high bonding energy are most likely candidates for equilibrium structures.

4.1.1 Triblock spheres where $\delta = 0.6$

For every unit cell simulation we recorded the lowest energy per particle during the simulation. The results for this are shown in table 4.1. There is not a lot of difference between the results: the values for 2, 3 and 4 particles are higher than for the unit cells with 5 or more particles. The lowest observed energy we see is $U/(N\epsilon^*) = -3.0$: in all the systems with this energy all particles are aligned in the same direction.

We selected five different systems based from these unit cell simulations and constructed candidate crystal structures. On these we performed another NPT simulation. When we look at the internal energy of the system as a function of Monte Carlo time (fig. 4.1), we see the energy of the system changed during the simulation. This might mean the system was not in equilibrium and is a good indicator of a phase transition, meaning that the crystal structures melt. We checked this by looking at the orientational order in the system. We did this in two ways: we calculated the orientation distribution function and made a projection of the patch orientation on the unit sphere. Figure 4.2 shows the orientational distribution function for a number, and the unit sphere projection of one of them. From both we see a lack of orientational order.

Because all systems melted we repeated the simulations with a lower temperature by making the bond energy stronger (from $\epsilon^* = 3$ to $\epsilon^* = 7$). In the unit cell simulations the results were not that different, as can be seen in table 4.2. In some cases we see higher energies. This is most likely due to the stronger bond energy causing the system to get stuck in a local minimum. The lowest energy observed is still $U/(N\epsilon^*) = -3.0$.

For the candidate crystal simulations we see that the systems with all aligned particles did not melt. This is confirmed by the graph of the orientational distribution function in 4.3. Here we see peaks around $\cos(\theta_{ij}) = \pm 1$. Again, the unit sphere projection of one shows that all patches point in the same direction.

A snapshot of one of the candidate crystal structures is shown in figure 4.4. We see this is an HCP stacking with the patches aligned perpendicular to the stacking layers. This result is in agreement with [36], where they observed this for patch surface coverage between 0.0572 and 0.118, but with a longer interaction length than we have used in our simulations.

4.1.2 Triblock spheres where $\delta = 0.9$

We performed the same simulations as before, this time with triblock spheres with larger patches. The patch half open angle is $\delta = 0.9$. When we look at the results for lowest energies for the unit sphere simulations in table 4.3. We see that almost all simulations reached a lowest energy of $U/(N\epsilon^*) = -4.0$. In all configurations the patches are aligned in the same direction.

We constructed a candidate crystal structure from this configuration and ran an NPT simulation. The energy as a function of simulation time remained constant, as can be seen in figure 4.5. This suggests that the crystal does not melt. This was confirmed by the orientation distribution function and projection on the unit sphere (fig. 4.6). Both show that all particles are aligned in the same direction. The configuration is different from the lowest energy crystal we found for $\delta = 0.6$. In this case, the layers are square, with a particle in the next layer

Table 4.1: Obtained minimum values $U/(N\epsilon^*)$ for triblock spheres with $\delta = \mathbf{0}.\mathbf{6}$ rad. For every system with $N = 2, 3. \dots, 9$ we performed five different simulations with the same parameters but a different random seed. These are labelled a, b, c, d, e. We used the parameters $P^* = 1$, $\epsilon^* = 3$ and $r_c^* = 0.2$.

	2	3	4	5	6	7	8	9
a	-2.5	-2.3	-2.7	-3.0	-3.0	-2.9	-2.9	-3.0
b	-2.3	-2.5	-2.3	-3.0	-3.0	-2.9	-2.8	-3.0
c	-2.5	-2.3	-2.7	-3.0	-3.0	-2.9	-2.9	-2.9
d	-2.5	-2.3	-2.7	-3.0	-2.9	-2.9	-3.0	-2.8
e	-2.3	-2.3	-2.7	-3.0	-3.0	-2.9	-2.9	-3.0

Table 4.2: Obtained minimum values $U/(N\epsilon^*)$ for triblock spheres with $\delta = \mathbf{0}.\mathbf{6}$ rad, this time with a stronger interaction energy. For every system with $N = 2, 3. \dots, 9$ we performed five different simulations with the same parameters but a different random seed. These are labelled a, b, c, d, e. We used the parameters $P^* = 1$, $\epsilon^* = \mathbf{7}$ and $r_c^* = 0.2$.

	2	3	4	5	6	7	8	9
a	-2.0	-2.3	-3.0	-3.0	-3.0	-2.6	-3.0	-2.7
b	-2.0	-2.0	-3.0	-3.0	-3.0	-3.0	-3.0	-2.7
c	-2.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-2.7
d	-2.0	-3.0	-3.0	-3.0	-3.0	-2.3	-3.0	-2.7
e	-2.0	-2.3	-3.0	-3.0	-2.5	-3.0	-3.0	-3.0

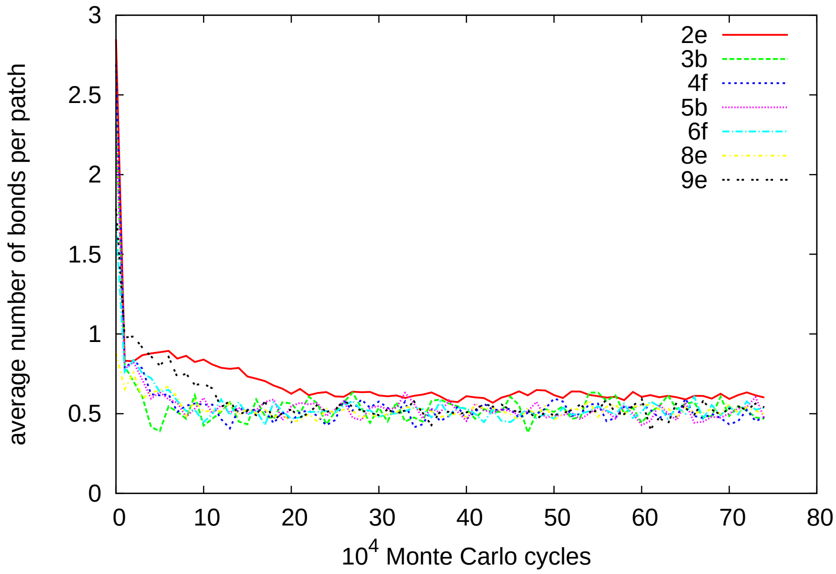


Figure 4.1: Evolution of the average number of bonds per patch for triblock spheres with half open angle $\delta = 0.6$. Every line corresponds to an NPT simulation of 192 to 512 particles. The starting configuration is constructed from the unit cell of one of the earlier runs. The lines are labelled by the number of particles in the unit cell and the corresponding run in the floppy box, denoted by a lower case letter. The parameters for all systems are $P^* = 1$, $\epsilon^* = 3$ and $r_c^* = 0.2$. We can see that a number of systems melted over the first 20×10^4 MC cycles.

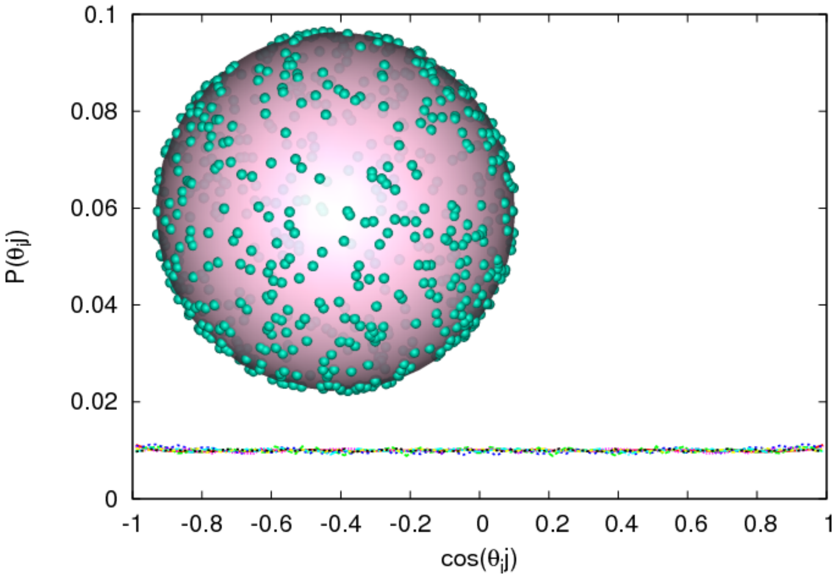


Figure 4.2: Orientation distribution function for triblock spheres with half open angle $\delta = 0.6$. For a number of unit cells we have constructed candidate crystals of at least 192 particles. The simulation parameters were $P^* = 1$, $\epsilon^* = 3$, $r_c^* = 0.2$. We calculated the orientational order after 1×10^6 MC cycles. This graph shows that all the systems have melted and show no orientational order. The inset shows the projection of the orientation of the patches on a unit sphere for one of the candidates. Both show no preferential orientation.

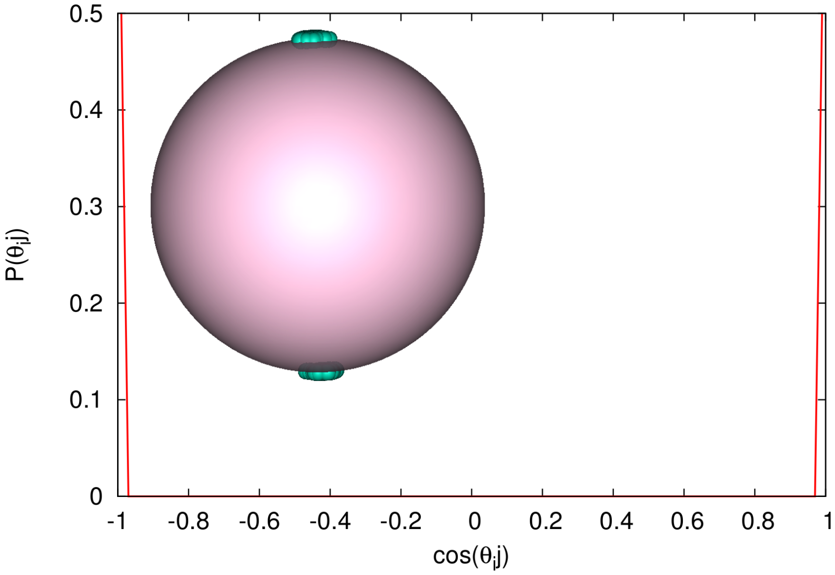


Figure 4.3: Orientation distribution function for triblock spheres with half open angle $\delta = 0.6$. We have constructed candidate crystals of at least 192 particles. The simulation parameters were $P^* = 1$, $\epsilon^* = 7$, $r_c^* = 0.2$. We calculated the orientational order after 1×10^6 MC cycles. The inset shows the projection of the orientation of the patches on a unit sphere. Both show that the patches align in the same direction.

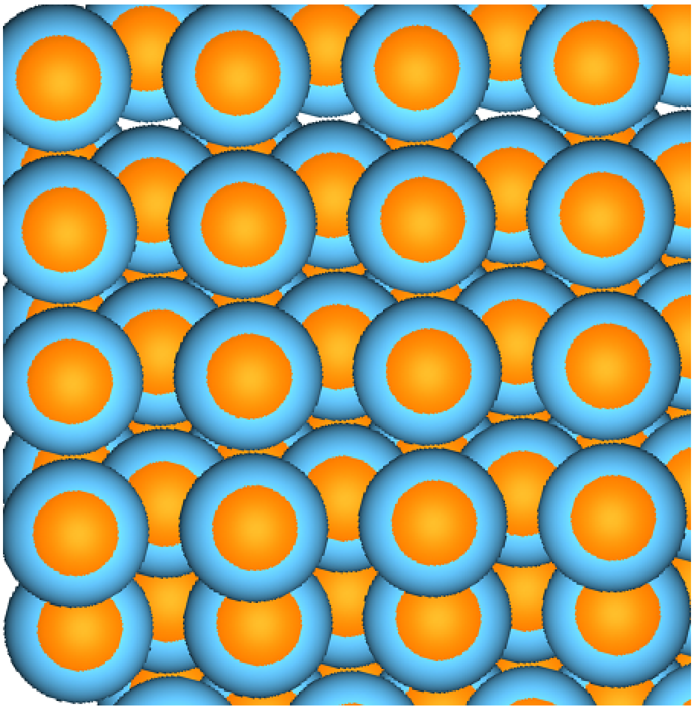


Figure 4.4: Snapshot of a crystal structure for triblock spheres with $\delta = 0.6$ with the lowest energy. This crystal is from the simulations with $\epsilon^* = 7$, constructed from the unit cell labelled $3d$. The structure is an HCP stacking with all particles aligned such that the patches bond between the stacking layers, forming three bonds per patch.

being in the center of four others. A snapshot of the configuration is shown in figure 4.7. This appears to be an FCC crystal.

Table 4.3: Obtained minimum values $U/(N\epsilon^*)$ for triblock spheres with $\delta = \mathbf{0.9}$ rad. For every system with $N = 2, 3. \ldots, 9$ we performed five different simulations with the same parameters but a different random seed. These are labelled a, b, c, d, e. We used the parameters $P^* = 1$, $\epsilon^* = 3$ and $r_c^* = 0.2$.

	2	3	4	5	6	7	8	9
a	-4.0	-4.0	-4.0	-4.0	-4.0	-4.0	-4.0	-4.0
b	-4.0	-4.0	-4.0	-4.0	-4.0	-4.0	-4.0	-3.7
c	-4.0	-4.0	-4.0	-4.0	-4.0	-4.0	-4.0	-3.6
d	-4.0	-4.0	-4.0	-4.0	-4.0	-4.0	-4.0	-3.6
e	-4.0	-4.0	-4.0	-4.0	-4.0	-4.0	-4.0	-4.0

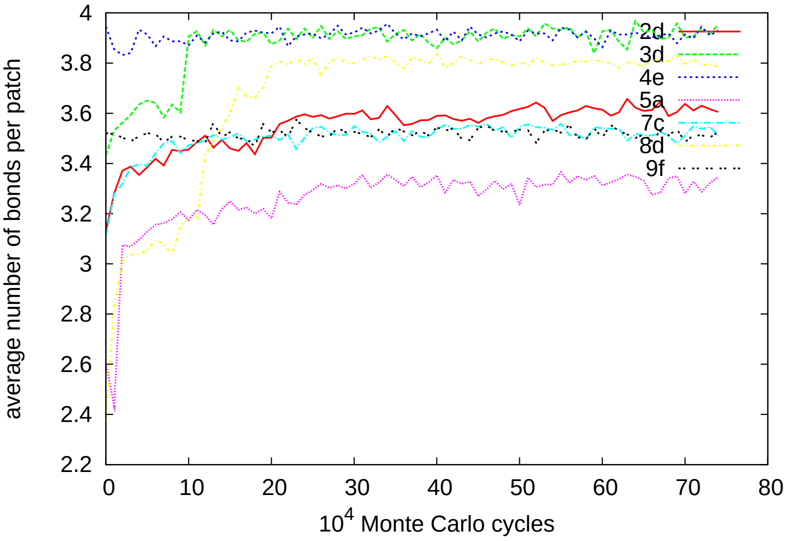


Figure 4.5: Evolution of the average number of bonds per patch for triblock spheres with half open angle $\delta = 0.9$. Every line corresponds to an NPT simulation of 192 to 512 particles. The starting configuration is constructed from the unit cell of one of the earlier runs. The lines are labelled by the number of particles in the unit cell and the corresponding run, denoted by a lower case letter. The parameters for all systems are $P^* = 1$, $\epsilon^* = 3$ and $r_c^* = 0.2$. We can see that a number of systems melt over the first 20×10^4 MC cycles.

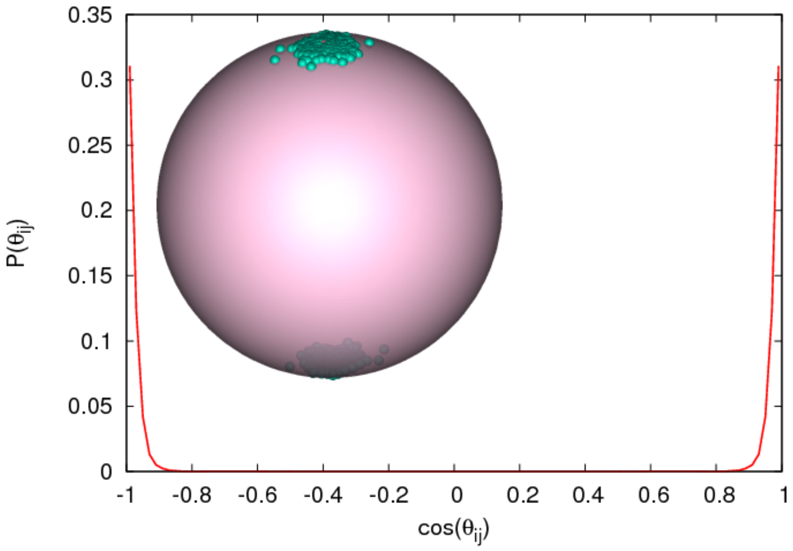


Figure 4.6: Orientation distribution function for triblock sphere with half open angle $\delta = 0.9$. For a number of unit cells we have constructed candidate crystals of at least 192 particles. On these crystal structures we performed a NTP simulation. The simulation parameters were $P^* = 1$, $\epsilon^* = 3$, $r_c^* = 0.2$. We calculated the orientational order after 1×10^6 MC cycles. This graph shows that all structures still have strong orientational order, they show peaks around (only) $\cos(\theta_{ij}) = \pm 1$. This means that all triblock spheres are aligned in the same direction. The inset shows the projection of the patch orientation on the unit sphere. Here we can also clearly see that all patches are aligned.

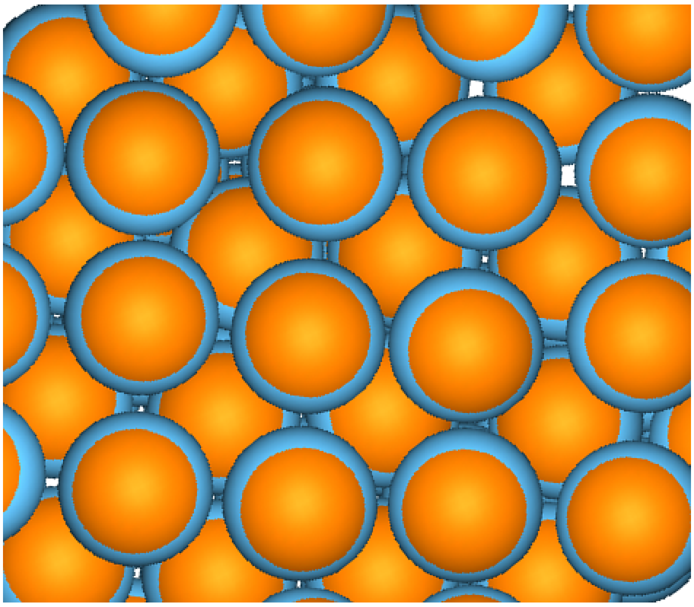


Figure 4.7: Snapshot of the crystal structure for triblock spheres with $\delta = 0.9$ with the lowest energy. The structure is an FCC crystal with the patches all aligned such that they can form four bonds each.

4.2 Spherical protrusions

Here we describe the results of applying our method to particles with spherical patchy protrusions. We looked at three different protrusion sizes. Expressed as the ratio of the diameter of the patchy sphere (σ_p) and diameter of the central hard sphere (σ_c), these are $q = \sigma_p/\sigma_c = 0.2$, 0.3 and 0.5. A side view is depicted in figure 4.8. The center of the patchy spheres is always placed on the surface of the central sphere.

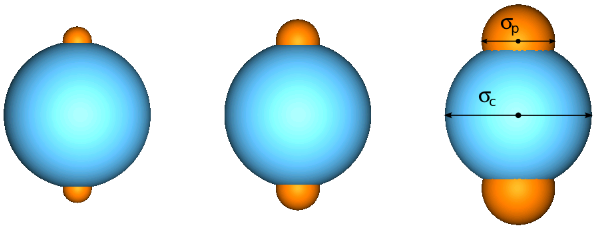


Figure 4.8: Spheres with spherical protrusions with size ratio $q = \sigma_p/\sigma_c = 0.2$, 0.3 and 0.5. The blue sphere has diameter σ_c and has a hard-core interaction. The orange spheres are places with their center on the surface of the blue sphere and have diameter σ_p. The orange spheres have an attractive square-well interaction.

For each particle we ran a number of small unit cell simulations: we varied the number of particles from 2 till 9, where we performed six simulations with different starting conditions each. The system parameters for all simulations were $P^* = 1$, $\epsilon^* = 3$, $r_c^* = 0.2$.

From these unit cell simulations we selected the candidates with the lowest energy per particle. As explained earlier, we are interested in structures at low temperature and pressure, where the patchy interactions are expected to be important. From these we constructed candidate crystals with a number of particles between 192 and 512, depending on the number of particles in the unit cell. For the NPT simulations we performed on these candidates we used the system parameters $P^* = 1$, $\epsilon^* = \mathbf{5}$, $r_c^* = 0.2$.

4.2.1 Spherical protrusions where $q = 0.2$

The particles with $q = 0.2$ have the smallest protrusions and can form the lowest number of bonds of the three parameter values that we investigated here. In table 4.4 we show the results of the unit cell simulations. For every simulation we show the minimum value of $U/(N\epsilon^*)$, e.g. the highest average number of bonds per patch. We see that most runs result in only two bonds with a lowest value of only -2.0. Three simulations for unit cells with eight particles have a lower energy of $U/(N\epsilon^*) = -3$. These three are the same structure.

For the candidate crystal simulations we selected six with two bonds and one of the unit cells with three bonds. The first candidate structures are all two-dimensional, sheet-like crystals that melted rapidly. The candidate crystal with three bonds did not melt. For this candidate we calculated the orientational distribution. As can be seen in figure 4.10, both the orientational distribution and the unit sphere projection clearly showed orientational order. In the distribution function we see the peaks around $\cos(\theta_{ij}) = \pm 1$ we have seen before and peaks around $\cos(\theta_{ij}) \approx \pm 0.35$.

The projection on the unit sphere shows eight distinct spots, which corresponds with four different orientations in the system. A snapshot (fig. 4.11) from the candidate crystal shows that the particles are in a close-packed FCC structure with the patches clustered in groups of four.

Table 4.4: Obtained minimum values $U/(N\epsilon^*)$ for particles with $q = 0.2$. For every system with $N = 2, 3. \dots, 9$ particles we performed six different simulations with the same parameters but a different random seed. These are labelled a, b, c, d, e. We used the parameters $P^* = 1$, $\epsilon^* = 3$ and $r_c^* = 0.2$.

	2	3	4	5	6	7	8	9
a	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0
b	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-3.0	-2.0
c	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-3.0	-2.0
d	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0
e	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0
f	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-3.0	-2.0

4.2.2 Spherical protrusions where $q = 0.3$

We performed the same simulations as before, this time with larger spherical protrusions. When we look at the results for lowest energies for the unit sphere simulations (table 4.5), we see values varied from $U/(N\epsilon^*) = -3.0$ to -4.5. In all cases the particles preferred to align in a perpendicular direction relative to its neighbours. The unit cells with the lowest energy we observe both with 3 and with 9 particles in the unit cell. Here the unit cell with 9 particles is a multiple of the 3 particle cell.

We performed the candidate crystal NPT simulations on a number systems for different unit cells. If we look at the energy as a function of MC cycles (fig. 4.12), we observed that a number did not melt, not only the candidate with the lowest energy. Therefore we lower the pressure after 10^6 MC cycles. This caused the melting of six more candidate structures. We see that the structures that melt slowly crystallized to structures similar to the candidate with the lowest energy.

We examined the orientational order of the candidate crystal structure, presented in figure 4.13. The orientational distribution function shows the perpendicular orientations by the extra peak around $\cos(\theta_{ij}) = 0$. In the unit sphere projection we see that the patches align in three different directions, indeed all perpendicular to each other. We show a snapshot of crystal structure in figure 4.14.

4.2.3 Spherical protrusions where $q = 0.5$

Finally we looked at the crystal structure formation of particles with spherical protrusions half the diameter of the center sphere. For the unit cell simulations we see a higher number of bonds than for smaller patches. The lowest energies in table 4.6 show a minimum of $U/(N\epsilon^*) = -6$.

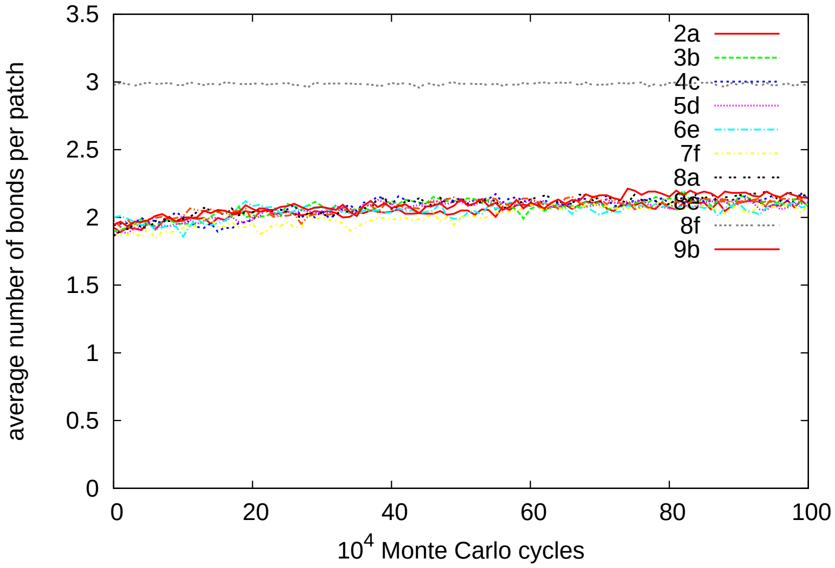


Figure 4.9: Evolution of the average number of bonds per patch for our particles with $q = 0.2$. Every line corresponds to an NPT simulation of 192 to 512 particles. The starting configuration is constructed from the unit cell of one of the earlier runs. The lines are labelled by the number of particles in the unit cell and the corresponding run, denoted by a lower case letter. The parameters for all systems are $P^* = 1$, $\epsilon^* = 5$ and $r_c^* = 0.2$. We can see that a number of systems melt over the first 20×10^4 MC cycles. All systems but one are in the fluid state and show no orientational order. The system of run f with a unit cell of eight particles $(8f)$ has a significantly lower energy with an average number of bonds of 3 and does not melt.

Table 4.5: Obtained minimum values $U/(N\epsilon^*)$ for particles with $q = \mathbf{0.3}$. For every system with $N = 2, 3. \ldots, 9$ particles we performed six different simulations with the same parameters but a different random seed. These are labelled a, b, c, d, e, f. We used the parameters $P^* = 1$, $\epsilon^* = 3$ and $r_c^* = 0.2$.

	2	3	4	5	6	7	8	9
a	-3.0	-2.7	-3.5	-4.0	-3.3	-3.4	-3.0	-4.2
b	-3.0	-2.7	-4.0	-4.0	-3.2	-3.7	-3.5	-4.3
c	-3.0	-4.5	-3.5	-4.0	-3.5	-4.1	-3.1	-3.2
d	-3.0	-3.3	-4.2	-4.0	-3.5	-3.1	-3.5	-4.5
e	-3.0	-2.7	-3.8	-4.0	-3.5	-3.4	-3.9	-3.4
f	-3.5	-2.7	-3.5	-4.0	-3.3	-4.1	-3.0	-4.3

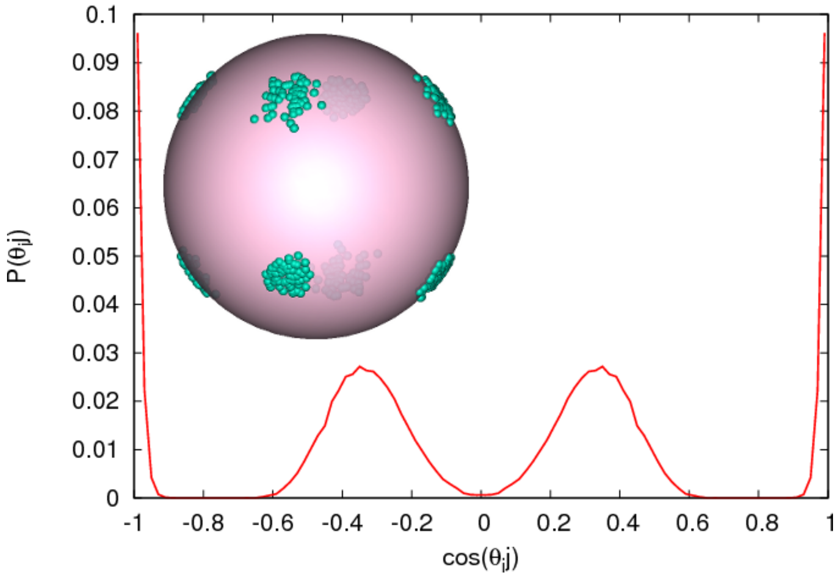


Figure 4.10: Orientation distribution function for patchy particles with $q = 0.2$. For a number of unit cells we have constructed candidate crystals of at least 192 particles. On these crystal structures we performed an NTP simulation. The simulation parameters were $P^* = 1$, $\epsilon^* = 5$, $r_c^* = 0.2$. We calculated the orientational order after 100×10^4 MC cycles. We can see from the graph that all but one system have no orientational order. Only one of the unit cells with eight particles ($8f$) has a profile with peaks at zero angle ($\cos(\theta_{ij}) = \pm 1$) and an angle of 70 degrees.

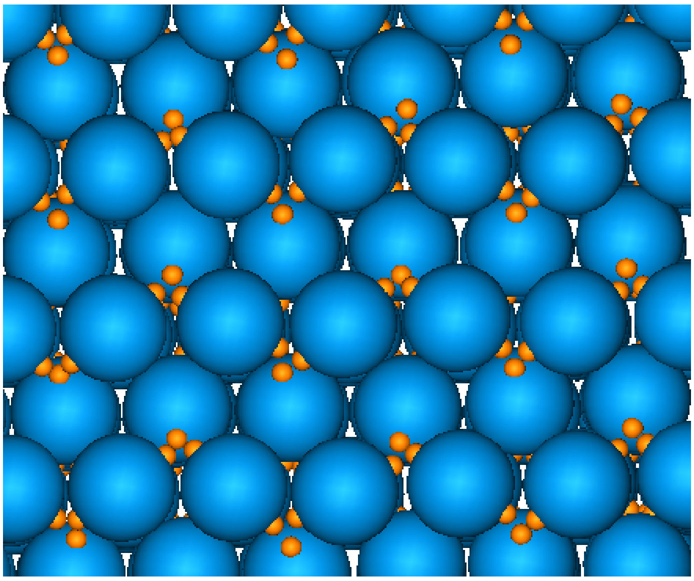


Figure 4.11: Snapshot from the candidate $8f$ with $q = 0.2$ that did not melt in the NPT simulations. The spacial structure is an FCC crystal with every patch forming three bonds.

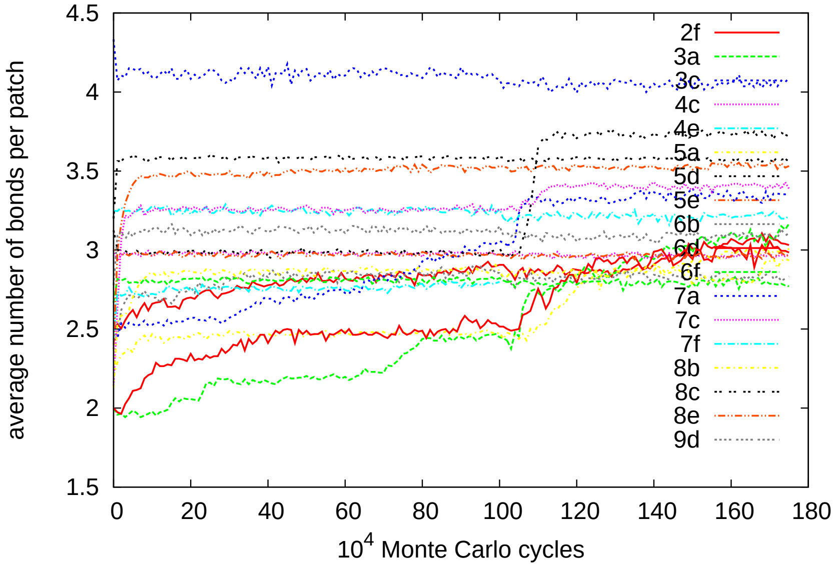


Figure 4.12: Evolution of the average number of bonds per patch for our particles with $q = 0.3$. The different candidate systems contain between 192 and 512 particles, depending on the unit cell. The start with very different values for the average number of bonds, and a number of systems with a relative low number does not melt initially. We therefore decreased the pressure to $P^* = 0$ after $100 \cdot 10^4$ MC cycles. This results in melting of a number of systems. The trial system with the lowest energy is run c of the unit cell with three particles ($3c$), which also has the most orientation distribution.

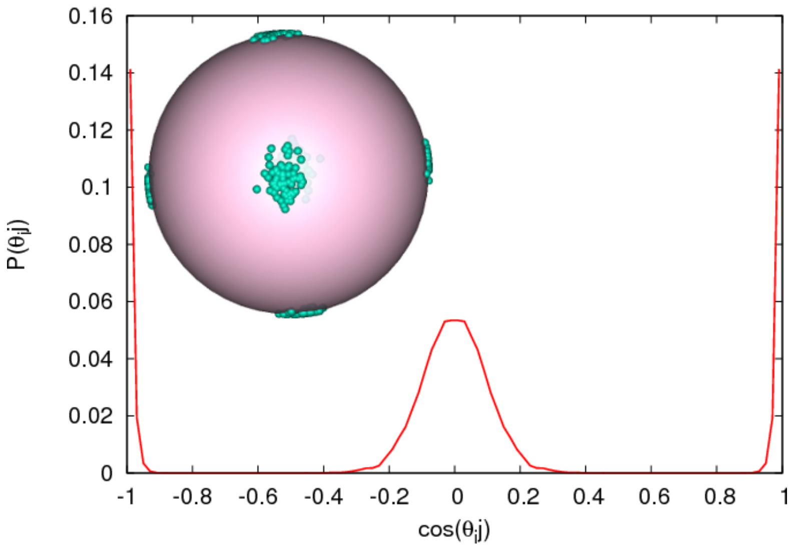


Figure 4.13: Orientation distribution function for patchy particles with $q = \mathbf{0.3}$. For a number of unit cells we have constructed candidate crystals of at least 192 particles. On these crystal structures we performed an NPT simulation. The simulation parameters were $P^* = 1$, $\epsilon^* = 5$, $r_c^* = 0.2$. We calculated the orientational order after 180×10^4 MC cycles. Many of the candidate crystal structures of this size ratio show orientation profile with peaks at ± 1 and 0. This means that the particles are aligned with each other in a perpendicular direction.

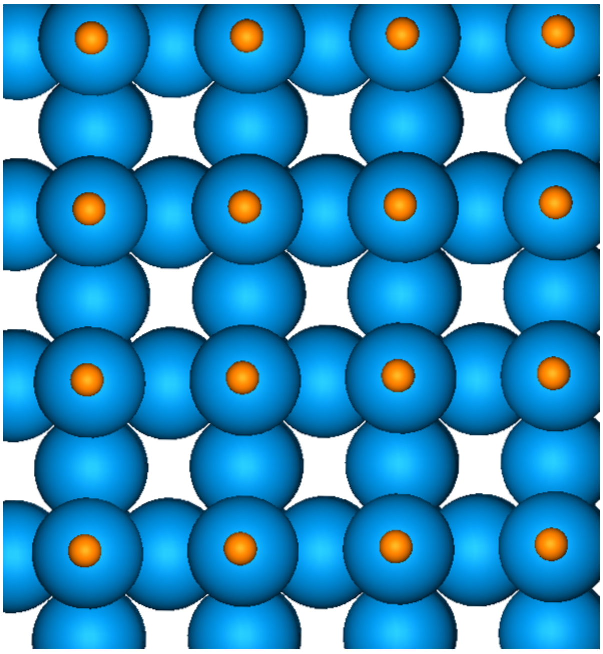


Figure 4.14: Snapshot of the crystal structure with the lowest energy for $q = 0.3$. The particles have formed an BCT crystal with patches in groups of six. This candidate crystal was constructed from the unit cell labelled $3c$ and has an average number of bonds per patch of 4.1.

Because of the small differences in energy we select a candidate for every number of particles in the unit cell. As can be seen in figure 4.15, most of these candidate structures melt. We select the remaining structure with the lowest energy for further analysis.

We examined the orientational order of the candidate structure as shown in figure 4.17. The orientational distribution function (fig. 4.16) shows extra peaks around $\cos(\theta_{ij}) \approx 0.5$. This looks similar to the distribution for $q = 0.2$, but the peaks are shifted. The projection on the unit sphere reveals that all the particles are oriented in a plane, which is very different from what we have seen for $q = 0.2$ and 0.3. This shows that examining the unit sphere project is useful, as we cannot extract this information from the orientational distribution function.

Table 4.6: Obtained minimum values $U/(N\epsilon^*)$ for particles with $q = \mathbf{0.5}$. For every system with $N = 2, 3. \ldots, 9$ particles we performed six different simulations with the same parameters but a different random seed. These are labelled a, b, c, d, e, f. We used the parameters $P^* = 1$, $\epsilon^* = 3$ and $r_c^* = 0.2$.

	2	3	4	5	6	7	8	9
a	-5.0	-5.7	-6.0	-5.6	-5.8	-5.7	-5.0	-5.1
b	-5.0	-5.3	-6.0	-5.6	-5.5	-5.7	-5.3	-5.3
c	-5.0	-5.7	-5.5	-5.6	-5.5	-6.0	-5.0	-5.2
d	-5.0	-5.7	-6.0	-5.6	-5.5	-5.6	-5.3	-5.2
e	-5.0	-5.3	-6.0	-5.6	-5.8	-5.0	-5.0	-5.3
f	-5.0	-5.3	-6.0	-5.6	-5.8	-6.0	-5.3	-5.2

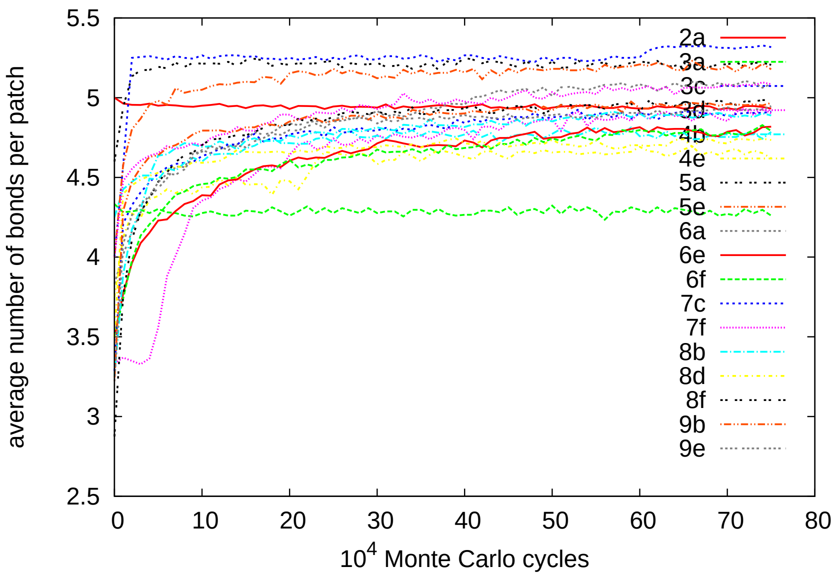


Figure 4.15: Evolution of the average number of bonds per patch for our particles with $q = 0.5$. Every line corresponds to an NPT simulation of 192 to 512 particles. The starting configuration is constructed from the unit cell of one of the earlier runs. The lines are labelled by the number of particles in the unit cell and the corresponding run, denoted by a lower case letter. The parameters for all systems are $P^* = 1$, $\epsilon^* = 5$ and $r_c^* = 0.2$. We can see that a number of systems melt over the first 20×10^4 MC cycles. The crystal structure with the lowest energy is run c with a unit cell of three particles ($3c$).

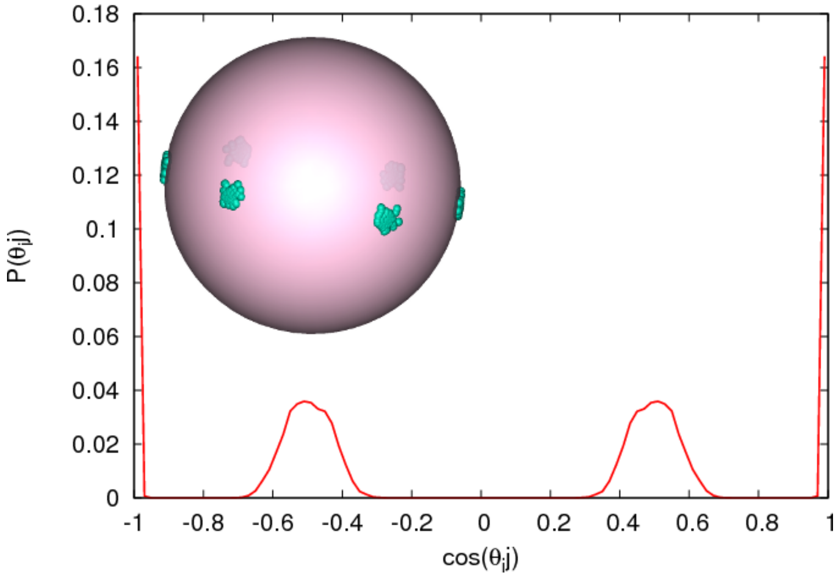


Figure 4.16: Orientation distribution function for patchy particles with $q = 0.5$. We have constructed candidate crystals of at least 192 particles. On this crystal structure we performed an NTP simulation. The simulation parameters where $P^* = 1$, $\epsilon^* = 5$, $r_c^* = 0.2$. We calculated the orientational order after 80×10^4 MC cycles. The inset shows the projection of the patch orientation on the unit spheres. We see that all are aligned in a plane with only three different orientations.

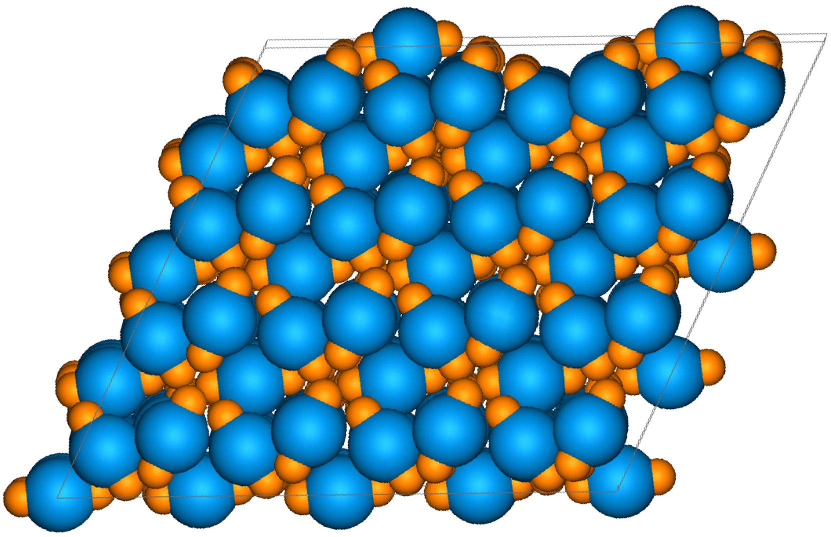


Figure 4.17: Snapshot of the crystal structure with lowest energy for $q = 0.5$. This was constructed from the unit cell labelled $3c$ and has an average of 5.3 bonds per patch. It is interesting to see that all the particles are oriented in the plane.

Chapter 5

Conclusions

Here we present our conclusions and discuss the validity of the results of our work on the prediction of crystal structures of patchy particles. We will also give some recommendations for future work on this subject.

We performed variable box shape Monte Carlo simulations to find candidate unit cells for crystal structures of patchy particles. To this end, we performed simulations in the NPT ensemble for a small number of particles. Then, we multiplied the found unit cells to obtain a crystal and investigated if it remained or melted in this large simulation box.

We highlighted the candidates with the highest number of bonds per patch, since we hypothesize that at the low pressures and temperatures that we investigated they are the most likely stable candidates.

We have shown that the floppy box method can be used to generate crystal structure candidates for particles with anisotropic interactions. The fact that we did not observe melting for the unit cells with the lowest energies shows the viability of the method for these systems.

By applying our method we have predicted crystal structures for two types of patchy particles. For triblock spheres we find structures that consist of all aligned particles. The size of the patches is an important factor. It determines the number of bonds a patch can form. For patches with half open angle $\delta = 0.6$ rad, they form three bonds per patch, which results in an HCP stacking. The triblock spheres with a larger half open angle of $\delta = 0.9$ rad can form four bonds per patch. This results in an FCC crystal structure. These results are similar to the results presented in [36].

The spheres with attractive spherical protrusions form candidate crystal structures with varying patch orientation when we change the patch/core size ratio $q = \sigma_p/\sigma_c$. The particles with $q = 0.2$ arrange in a close-packed FCC structure with three bonds per patch. Increasing the size of the protrusion to $q = 0.3$ gives candidate structures with particles aligned in a perpendicular fashion. The structure with the lowest energy that we find most often is a low density BCT crystal with every patch forming four bonds. When the patch size is increased to $q = 0.5$, the structure with the lowest internal energy is a crystal where all particles are aligned in a plane with well-defined angles between neighbouring particles.

For future work and more conclusive predictions, it is required to extend the results to more temperatures and pressures. To predict the thermodynamic stability of the crystal structures and draw up a phase diagram free energy calculations will have to be performed. The crystal structures we have presented here will be a good starting point for this.

Further refinements to our method are certainly possible. Improvements can be made to prevent that the unit cell simulations get stuck in a local minimum state. One improvement could be the addition of a lattice reduction step, such as the algorithm in [41], to minimize the deformation of the unit cell.

Particles with two protrusions on opposite sides are a relative simple model for simulations, but synthesis of these particles might be difficult. Therefore applying our method to patch arrangements that are easier to construct would be worthwhile. An example of this are spheres with protrusions in a tetragonal pattern, which could form diamond-like crystals [42].

Appendix A

Code examples

A.1 Vector functions

These are a number of helper functions we wrote to handle vectors in C, which does not have any build-in support for vectors.

```
#include <math.h>

/* vec_inprod: returns the inproduct of two vectors */
static inline double vec_inprod(double a[3], double b[3])
{
    double t = 0.0;

    for (int d = 0; d < 3; d++) {
        t += a[d] * b[d];
    }
    return t;
}

/* vec_norm: normalizes input vector to unit length */
static inline void vec_norm(double v[])
{
    double t = sqrt(vec_inprod(v, v));

    for (int d = 0; d < 3; d++) {
        v[d] /= t;
    }
}

/* vec_set: shorthand to set all three elements of a vector */
static inline void vec_set(double v[], double x, double y, double z)
{ v[0] = x; v[1] = y; v[2] = z; }

/* vec_cross: store cross product of vector a and b in c */
static inline void vec_cross(double c[], double a[], double b[])
{
    c[0] = a[1] * b[2] - a[2] * b[1];
    c[1] = a[2] * b[0] - a[0] * b[2];
    c[2] = a[0] * b[1] - a[1] * b[0];
}

/* vec_rotate: rotate vector v by t rad around vector w */
static inline void vec_rotate(double v[], double w[], double t)
{
    double cos_t, sin_t, wdv,
    double v2[3];

    cos_t = cos(t);
    sin_t = sin(t);
    wdv   = vec_inprod(w, v);

    v2[0] = v[0]*cos_t+(w[1]*v[2]-w[2]*v[1])*sin_t+w[0]*wdv*(1-cos_t);
    v2[1] = v[1]*cos_t+(w[2]*v[0]-w[0]*v[2])*sin_t+w[1]*wdv*(1-cos_t);
    v2[2] = v[2]*cos_t+(w[0]*v[1]-w[1]*v[0])*sin_t+w[2]*wdv*(1-cos_t);

    for (int d = 0; d < 3; d++) {
        v[d] = v2[d];
    }
}
```

A.2 Particle-particle interaction

This is the code that calculates the interaction between two different particles with spherical protrusions. It converts the particle coordinates from the box system to Cartesian coordinates. We check every pair of spheres first for possible overlap and then for alignment of their patch.

```
/* pair_energy: calculate interaction energy between particles
                p and q and adds this value to *en
                returns 1 if particles overlap, 0 otherwise */
int pair_energy(int p, int q, double *en)
{

    double pc[3][3], qc[3][3], dc[3];
    double pb[3][3], qb[3][3], db[3];
    double dist;

    if (p == q)
        return 0;

    /* for particle p */
    /* center pos in cartesian */
    cartesian(pc[0], part[p].x);
    /* patch pos in cartesian */
    for (int d = 0; d < 3; d++) {
        pc[1][d] = pc[0][d] + offset*part[p].p[1].u[d];
        pc[2][d] = pc[0][d] + offset*part[p].p[2].u[d];
    }
    /* all pos in box */
    vec_cp(pb[0], part[p].x);
    boxsys(pb[1], pc[1]);
    boxsys(pb[2], pc[2]);
    /* put in box! */
    periodic(pb[1]);
    periodic(pb[2]);

    /* for particle q */
    /* center pos in cartesian */
    cartesian(qc[0], part[q].x);
    /* patch pos in cartesian */
    for (int d = 0; d < 3; d++) {
        qc[1][d] = qc[0][d] + offset*part[q].p[1].u[d];
        qc[2][d] = qc[0][d] + offset*part[q].p[2].u[d];
    }
    /* all pos in box */
    vec_cp(qb[0], part[q].x);
    boxsys(qb[1], qc[1]);
    boxsys(qb[2], qc[2]);
    /* put in box */
    periodic(qb[1]);
    periodic(qb[2]);

    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            /* distance in box */
            for (int d = 0; d < 3; d++) {
                db[d] = qb[j][d] - pb[i][d];
            }
            image(db); /* apply nearest image convention */
            cartesian(dc, db);
            dist = sqrt(dc[0]*dc[0] + dc[1]*dc[1] + dc[2]*dc[2]);
            if (i == 0 && j == 0 && dist > (2.0*(offset+radius+rc)))
                continue;
            if (dist <= (part[p].p[i].r + part[q].p[j].r))
                return 1;
            if (dist > (part[p].p[i].r + part[q].p[j].r + 1.0*rc)) {
                continue;
            }
            vec_norm(dc);

            if (vec_inprod(part[p].p[i].u, dc) <= part[p].p[i].d)
                continue; /* patch on p not aligned */
            if (vec_inprod(part[q].p[j].u, dc)*-1.0 <= part[q].p[j].d)
                continue; /* patch on q not aligned */
            *en += en_u;
        }
    }
    return 0;
}
```

A.3 Cell lists

This code shows how we have implemented the cell lists.

```
#define NCELL   (32) /* maximum number of cells in every direction */

struct cells {
    int head[NCELL][NCELL][NCELL]; /* heads of cell lists */
    const int neighbor[27][3];     /* neighbour offsets by proximity */
    int num[3];                    /* number of cells per dimension */
    double size[3];                /* size of one cell */
};

/* cell lists - initialize with fixed offsets of neighbour cells */
struct cells cell = {
        { { {0} } },
        { { 0, 0, 0}, /* cell self */
          { 0, 0, 1}, { 0, 1, 0}, { 1, 0, 0}, { 0, 0,-1}, { 0,-1,0},
          {-1, 0, 0}, /* closest six */
          { 0, 1, 1}, { 0,-1, 1}, { 0, 1,-1}, { 0,-1,-1}, { 1, 0, 1},
          {-1, 0, 1}, { 1, 0,-1}, {-1, 0,-1}, { 1, 1, 0}, {-1, 1, 0},
          { 1,-1, 0}, {-1,-1,0}, /* next twelve */
          { 1, 1, 1}, { 1,-1, 1}, {-1, 1, 1}, {-1,-1, 1}, { 1, 1,-1},
          { 1,-1,-1}, {-1, 1,-1}, {-1,-1,-1} /* final eight */
         },
        {0}, {0.0}
};

/* cell_part: calculate cell where particle p belongs */
static inline void cell_part(int p, int pc[])
{
    for (int d = 0; d < 3; d++) {
        pc[d] = part[p].x[d] / cell.size[d];
    }
}

/* cell_first: returns first particle in cell list or -1 if empty */
static inline int cell_first(int r[]) { return cell.head[r[0]][r[1]][r[2]]; }

/* cell_next: returns next particle in cell list or -1 if last */
static inline int cell_next(int i) { return part[i].next; }

/* cell_add: adds a particle to the right cell list */
void cell_add(int p)
{
    int x, y, z;

    cell_part(p, part[p].cell);
    x = part[p].cell[0];
    y = part[p].cell[1];
    z = part[p].cell[2];
    part[p].next = cell.head[x][y][z];
    cell.head[x][y][z] = p;
}
```

```
/* cell_init: create cell lists */
void cell_init(void)
{
    /* calculate number of cells and cell size */
    for (int d = 0; d < 3; d++) {
        cell.num[d]  = box[d] / (1.0*(2*radius+1.+rc));
        cell.size[d] = box[d] / cell.num[d];
    }
    /* clear cells */
    for (int x = 0; x < cell.num[0]; x++) {
        for (int y = 0; y < cell.num[1]; y++) {
            for (int z = 0; z < cell.num[2]; z++) {
                cell.head[x][y][z] = -1;
            }
        }
    }
}

/* cell_neighbour: switch to ith neighbour of cell */
void cell_neighbour(int pc[3], int i, int neighbour[3])
{
    for (int d = 0; d < 3; d++) {
        neighbour[d] = pc[d] + cell.neighbor[i][d];
        if (neighbour[d] < 0) {
            neighbour[d] += cell.num[d];
        } else if (neighbour[d] >= cell.num[d]) {
            neighbour[d] -= cell.num[d];
        }
    }
}

/* cell_update_all: recreates cell lists when needed */
void cell_update_all(void)
{
    int update;

    for (int d = 0; d < 3; d++) {
        cell.size[d] = box[d] / cell.num[d];
    }
    update = 0;
    for (int d = 0; d < 3; d++) {
        if (cell.num[d] != (int) (box[d] / (1.0*(2*radius+1.+rc)))) {
            update = 1; /* update is needed */
        }
    }
    if (update) {
        cell_init();
        /* add particles to cells */
        for (int i = 0; i < NPART; i++) {
            cell_add(i);
        }
    }
}
```

```
/* cell_update: updates the cell of a single particle */
void cell_update(int p)
{
    int x, y, z, q, s[3];

    /* update cell */
    x = part[p].cell[0];
    y = part[p].cell[1];
    z = part[p].cell[2];
    cell_part(p, s);
    if (s[0] == x && s[1] == y && s[2] == z)
        return; /* p still in same cell */
    /* remove p from old cell */
    if (cell.head[x][y][z] == p) {
        cell.head[x][y][z] = part[p].next;
    } else {
        for (q = cell_first(part[p].cell);
             q != -1 && cell_next(q) != p;
             q = cell_next(q))
            ; /* empty loop */
        part[q].next = part[p].next;
    }
    /* add p to new cell */
    cell_add(p);
}
```

A.4 Orientation distribution function

The main part of the calculation of the orientation distribution function. We make use of the fact that our input contains pairs of anti-aligned unit vectors.

```
#include <math.h>

#define NBIN 200
#define BIN(x) ((int)((x+1.0)*0.5*NBIN))

/* orientdistrib: calculate orientation distribution function
                  from unit vectors in v and store in bin
                  returns the total number of values binned */
int orientdistrib(int bin[], double vec[][3], int m)
{
    int k = 0;

    /* 'first patches' - note start value of j */
    for (int i = 0; i < m; i += 2) {
        for (int j = i + 2; j < m; j++) {
            x = vec_inprod(v[i], v[j]);
            ++bin[BIN(x)];
            ++k;
        }
    }
    /* 'second patches' - note start value of j */
    for (int i = 1; i < m; i += 2) {
        for (int j = i + 1; j < m; j++) {
            x = vec_inprod(v[i], v[j]);
            ++bin[BIN(x)];
            ++k;
        }
    }
    return k;
}
```

Bibliography

[1] R. Brown, “XXVII. A brief account of microscopical observations made in the months of June, July and August 1827, on the particles contained in the pollen of plants; and on the general existence of active molecules in organic and inorganic bodies,” *Philosophical Magazine Series 2*, vol. 4, no. 21, pp. 161–173, 1828.

[2] T. Graham, “Liquid diffusion applied to analysis,” *Philosophical Transactions of the Royal Society of London*, vol. 151, pp. 183–224, 1861.

[3] A. Einstein, “Über die von der molekularkinetischen Theorie der Wärme geforderte Bewegung von in ruhenden Flüssigkeiten suspendierten Teilchen,” *Annalen der Physik*, vol. 322, no. 8, pp. 549–560, 1905.

[4] W. Sutherland, “LXXV. A dynamical theory of diffusion for non-electrolytes and the molecular mass of albumin,” *Philosophical Magazine Series 6*, vol. 9, no. 54, pp. 781–785, 1905.

[5] J. Perrin, “Mouvement brownien et réalité moléculaire,” in *Annales de Chimie et de Physique*, vol. 18, pp. 5–104, 1909.

[6] W. Wood and J. Jacobson, “Preliminary results from a recalculation of the Monte Carlo equation of state of hard spheres,” *The Journal of Chemical Physics*, vol. 27, p. 1207, 1957.

[7] B. Alder and T. Wainwright, “Phase transition for a hard sphere system,” *The Journal of Chemical Physics*, vol. 27, p. 1208, 1957.

[8] K. Fontell, P. Vol, G. Gray, P. Winsor, and P. Wtnsor, “Phase behaviour of concentrated suspensions of nearly hard colloidal spheres,” *Nature*, vol. 320, p. 27, 1986.

[9] M. Leunissen, C. Christova, A. Hynninen, C. Royall, A. Campbell, A. Imhof, M. Dijkstra, R. Van Roij, and A. Van Blaaderen, “Ionic colloidal crystals of oppositely charged particles,” *Nature*, vol. 437, no. 7056, pp. 235–240, 2005.

[10] A. Yethiraj, “Tunable colloids: control of colloidal phase transitions with tunable interactions,” *Soft Matter*, vol. 3, pp. 1099–1115, 2007.

[11] S. Glotzer and M. Solomon, “Anisotropy of building blocks and their assembly into complex structures,” *Nature materials*, vol. 6, no. 8, pp. 557–562, 2007.

[12] S. V. Savenko and M. Dijkstra, "Sedimentation and multiphase equilibria in suspensions of colloidal hard rods," *Phys. Rev. E*, vol. 70, p. 051401, Nov 2004.

[13] Y. Sun and Y. Xia, "Shape-controlled synthesis of gold and silver nanoparticles," *Science*, vol. 298, no. 5601, pp. 2176–2179, 2002.

[14] L. Rossi, S. Sacanna, W. T. M. Irvine, P. M. Chaikin, D. J. Pine, and A. P. Philipse, "Cubic crystals from cubic colloids," *Soft Matter*, vol. 7, pp. 4139–4142, 2011.

[15] T. S. Ahmadi, Z. L. Wang, T. C. Green, A. Henglein, and M. A. El-Sayed, "Shape-controlled synthesis of colloidal platinum nanoparticles," *Science*, vol. 272, no. 5270, pp. 1924–1925, 1996.

[16] K. Miszta, J. de Graaf, G. Bertoni, D. Dorfs, R. Brescia, S. Marras, L. Ceseracciu, R. Cingolani, R. van Roij, M. Dijkstra, *et al.*, "Hierarchical self-assembly of suspended branched colloidal nanocrystals into superlattice structures," *Nature Materials*, 2011.

[17] C. Quilliet, C. Zoldesi, C. Riera, A. van Blaaderen, and A. Imhof, "Anisotropic colloids through non-trivial buckling," *The European Physical Journal E: Soft Matter and Biological Physics*, vol. 27, pp. 13–20, 2008. 10.1140/epje/i2007-10365-2.

[18] F. Smallenburg, L. Filion, M. Marechal, and M. Dijkstra, "Vacancy-stabilized crystalline order in hard cubes," *ArXiv e-prints*, Nov. 2011.

[19] R. Ni, A. Prasad Gantapara, J. de Graaf, R. van Roij, and M. Dijkstra, "Phase diagram of colloidal hard superballs: from cubes via spheres to octahedra," *ArXiv e-prints*, Nov. 2011.

[20] A. van Blaaderen, "Materials science: Colloids get complex," *Nature*, vol. 439, no. 7076, pp. 545–546, 2006.

[21] W. R. Smith and I. Nezbeda, "A simple model for associated fluids," *The Journal of Chemical Physics*, vol. 81, no. 8, pp. 3694–3699, 1984.

[22] M. Wertheim, "Fluids with highly directional attractive forces. i. statistical thermodynamics," *Journal of Statistical Physics*, vol. 35, no. 1, pp. 19–34, 1984.

[23] E. Bianchi, J. Largo, P. Tartaglia, E. Zaccarelli, and F. Sciortino, "Phase diagram of patchy colloids: Towards empty liquids," *Phys. Rev. Lett.*, vol. 97, p. 168301, Oct 2006.

[24] N. Kern and D. Frenkel, "Fluid–fluid coexistence in colloidal systems with short-ranged strongly directional attraction," *The Journal of Chemical Physics*, vol. 118, no. 21, pp. 9882–9889, 2003.

[25] E. Bianchi, R. Blaak, and C. N. Likos, "Patchy colloids: state of the art and perspectives," *Phys. Chem. Chem. Phys.*, vol. 13, pp. 6397–6410, 2011.

[26] Y.-S. Cho, G.-R. Yi, S.-H. Kim, M. T. Elsesser, D. R. Breed, and S.-M. Yang, "Homogeneous and heterogeneous binary colloidal clusters formed by evaporation-induced self-assembly inside droplets," *Journal of Colloid and Interface Science*, vol. 318, no. 1, pp. 124 – 133, 2008.

[27] D. J. Kraft, W. S. Vlug, C. M. van Kats, A. van Blaaderen, A. Imhof, and W. K. Kegel, "Self-assembly of colloids with liquid protrusions," *Journal of the American Chemical Society*, vol. 131, no. 3, pp. 1182–1186, 2009.

[28] A. Alivisatos, K. Johnsson, X. Peng, T. Wilson, C. Loweth, M. Bruchez, and P. Schultz, "Organization of'nanocrystal molecules' using DNA," 1996.

[29] S. Angioletti-Uberti, B. Mognetti, and D. Frenkel, "Re-entrant melting as a design principle for DNA-coated colloids," *Nature Materials*, 2012.

[30] A. B. Pawar and I. Kretzschmar, "Patchy particles by glancing angle deposition," *Langmuir*, vol. 24, no. 2, pp. 355–358, 2008. PMID: 18076199.

[31] P. G. de Gennes, "Soft matter," *Rev. Mod. Phys.*, vol. 64, pp. 645–648, Jul 1992.

[32] F. Sciortino, A. Giacometti, and G. Pastore, "A numerical study of one-patch colloidal particles: from square-well to janus," *Phys. Chem. Chem. Phys.*, vol. 12, pp. 11869–11877, 2010.

[33] F. Sciortino, A. Giacometti, and G. Pastore, "Phase diagram of janus particles," *Phys. Rev. Lett.*, vol. 103, p. 237801, Nov 2009.

[34] Q. Chen, S. C. Bae, and S. Granick, "Directed self-assembly of a colloidal kagome lattice," vol. 469, pp. 381–384, 2011.

[35] F. Romano and F. Sciortino, "Two dimensional assembly of triblock janus particles into crystal phases in the two bond per patch limit," *Soft Matter*, vol. 7, pp. 5799–5804, 2011.

[36] A. Giacometti, F. Lado, J. Largo, G. Pastore, and F. Sciortino, "Effects of patch size and number within a simple model of patchy colloids," vol. 132, no. 17, p. 174110, 2010.

[37] L. Filion, M. Marechal, B. van Oorschot, D. Pelt, F. Smallenburg, and M. Dijkstra, "Efficient method for predicting crystal structures at finite temperature: Variable box shape simulations," *Phys. Rev. Lett.*, vol. 103, p. 188302, Oct 2009.

[38] E. W. Weisstein, "Spherical cap," *MathWorld*.

[39] D. Frenkel and B. Smit, *Understanding molecular simulation: from algorithms to applications*, vol. 1. Academic Pr, 2002.

[40] L. Verlet, "Computer "experiments" on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules," *Phys. Rev.*, vol. 159, pp. 98–103, Jul 1967.

[41] D. Gottwald, G. Kahl, and C. N. Likos, "Predicting equilibrium structures in freezing processes," *The Journal of Chemical Physics*, vol. 122, no. 20, p. 204503, 2005.

[42] F. Romano, E. Sanz, and F. Sciortino, "Phase diagram of a tetrahedral patchy particle model for different interaction ranges," *The Journal of Chemical Physics*, vol. 132, no. 18, p. 184501, 2010.