Towards Simulation Studies of Crystal Structures for Superellipsoids by Floppy Box Monte Carlo Method

Evert Hildebrand
Supervisors:
Prof. Dr. Ir. M. Dijkstra
J. de Graaf, MSc.

August 18, 2011

Abstract

We will present a simulation technique, by which the crystal structure of superballs can be analysed. Superballs are particles which can be defined by the implicit equation $|x|^\epsilon + |y|^\epsilon + |z|^\epsilon <= 1$, $\epsilon \in [0, \infty[$. These particles have been recently synthesized and analyzed by using experiments and simulations [6]. The study of such particles is therefore a hot topic. Using our [10] technique we have generated crystal structures which can be obtained for spheres and superballs. Moreover we have studied the relation between pressure and density in the liquid phase by determining the Equation of State. We also considered how the structure of the liquid deviates from that of spheres upon changing the pressure p, by constructing the Radial distribution function. We verified that the simulation we wrote, functions well within the numerical tolerances required for physical studies. Finally, our method may be easily extended upon to determine a phase diagram for superballs, for which the particles may interact with soft potentials in addition to their hard particle overlap.

Contents

Chapter 1

Introduction

In the 18th century Buffon did experiments by throwing a needle onto the ground. He threw a needle a number of times on a part of the ground which was made out of stripes. He asked himself what the probability was to find the needle across the line between two stripes. By keeping track of the ratio between the number of times the needle is crossing a line and the number of times the needle falls exactly onto one stripe, he got an estimate of the number π. This experiment includes an amount of uncertainty, because we never know how many crossings there will be if we throw n times. Physicists used statistical sampling to estimate uncertainties in previously understood deterministic problems. Before computers were developed, this was done by statistical modeling, which basically means that statistical functions were used to analyse the uncertainties. Von Neumann and Ulam developed a method which deals with this statistical sampling, such that uncertainties can be used to understand physical problems. This method is called the Monte Carlo Method. When computers in the first half of the 20th century were developed, this method became of interest. The method was used to investigate many-particle systems in so called computer simulations. These computer simulations, also known as computer experiments, can be used to complement experimental studies. In 1942 during World war II the United States with participation from the United Kingdom and Canada started to develop a hydrogen bomb. Although the functionality of computers were limited, they used Monte Carlo Simulations as part of their development strategy. In 1954 Monte Carlo Simulations were used in the Los Alamos Scientific Laboratory to simulate 2 dimensional hard discs. The simulations were done with the purpose to get insight into liquid and solid phases, which led to a phase diagram. Simulations are also used to measure quantities that can be experimentally ascertained, in order to get insight into the mechanisms which underly the physics of the often more complicated experimental systems. It is also possible to directly measure quantities and dependencies which can only indirectly be extracted from experimental data.

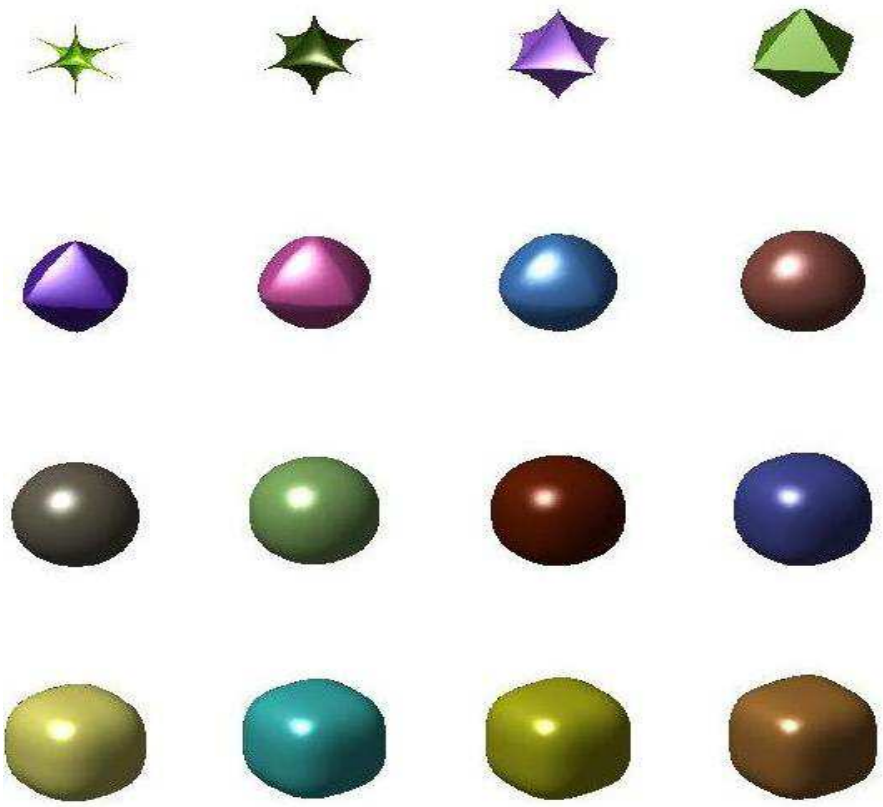

Figure 1.1: Examples of superballs on the right side of the spectrum we find cubic like particles and on the left side of the spectrum we get more star shaped particles.

In this report we present results for systems consisting of hard superballs, which we generated by using the Floppy Box Monte Carlo (FBMC) method, by which the crystal structure can be analysed. Superballs are particles which can be defined by the implicit equation $|x|^\epsilon + |y|^\epsilon + |z|^\epsilon \leq 1$, $\epsilon \in [0, \infty[$ (Fig 1.1). These particles have been recently synthesized and analyzed by using experiments and simulations [6][11]. By generating configurations of superballs with a hard overlap potential, we try to get insight in how hard superballs are related to hard spheres, by studying the equation of state, i.e. the relation between pressure and packing fraction η. We also considered how the structure of superballs in the liquid phase deviates from that of spheres upon changing the pressure p, by constructing the Radial distribution function. We were able to implement a program, which generates structures with superballs for $1.4 \lessapprox \epsilon <\lessapprox 3.6$. From data generated by the program, we constructed the equation of state in the liquid phase. We also managed to determine the Radial Distribution Function and see how this deviates from the one of spheres.
Finally, the FBMC method can be easily extended to determine the phase diagram for superballs, for which the particles may interact with soft potentials in additions to their hard particle overlap. The overlap routine for superballs or

general ellipsoids is only stable for values of ϵ mentioned above. Physically it is of interest to simulate systems with more cubic shaped superballs (large value of ϵ), because cubes are often used as a starting point for synthesis of other particles.

Chapter 2

Introduction to Simulation Techniques

The simulation studies we have performed are restricted to systems which may be described by the laws of classical mechanics. A system of N particles can be described by the position, orientation, momentum and shape of each particle. We use the language of statistical mechanics and thermodynamics to describe our systems. A system with volume V and a fixed number of N particles is in a state fixed by the momenta and positions of all the particles at a certain time t. This means that all the particles are at a time t at possible positions $\{\mathbf{r}_i\}$ with momenta $\{\mathbf{p}_i\}$, with $i = 1, \ldots, N$. A possible state in which a system can be is called a microstate. A system with fixed energy E can typically be found in one of the many possible microstates, i.e., there are many microstates which achieve a total energy E. We denote by $\Omega(E, V, N)$ the number of microstates with energy E of a system with N particles and volume V. We assume that every system with fixed E, V and N can be equally likely be found in one of its microstates. This is the fundamental assumption of statical mechanics.
Suppose we have a system with a total energy E which is subdivided into two subsystems, say 1 and 2, which can exchange energy between one another but no particles or volume. The energy of system 1 is E_1 and of system 2 is E_2. The subsystems have fixed volumes V_1 and V_2 with a fixed number of particles N_1 and N_2, respectively. The number of microstates of these two subsystems are denoted by $\Omega_1(E_1, N_1, V_1)$ and $\Omega_2(E_2, N_2, V_2)$ respectively. The total system is closed and the energies E_1 and E_2 therefore satisfy $E_1 + E_2 = E$. The number of microstates of the combined system $\Omega(E = E_1 + E_2)$ is given by

$$\Omega(E, N_1, V_1, N_2, V_2) = \sum_{E_1} \Omega_1(E_1, V_1, N_1)\Omega_2(E - E_1, V_2, N_2), \tag{2.1}$$

where summation is used to account for all possible combinations. Because the microstates of the combined system all have the same energy, the states are also called degenerate states. The microstates compatible with a given

energy form an ensemble. By the fundamental assumption every microstate of $\Omega(E, N_1, V_1, N_2, V_2)$ is equally likely. Therefore the probability $W(E_1)$ to find subsystem 1 with energy E_1 is equal to the fraction of microstates in which system 1 is at energy E_1. Therefore

$$W(E_1) = \frac{\Omega_1(E_1)\Omega_2(E - E_1)}{\Omega(E)}, \tag{2.2}$$

where we have dropped the dependence on the number of particles and the volume for notational convenience. The most probable value of E_1 is the one for which $W(E_1)$ is at its maximum. The extremum can be obtained by

$$\frac{dW(E_1)}{dE_1} = 0. \tag{2.3}$$

Combining Eq. (2.2) with Eq. (2.3) we derive that

$$\frac{d}{dE_1}\Omega_1(E_1)\Omega_2(E - E_1) = \frac{d}{dE_1}\Omega_1(E_1)\Omega_2(E_2) = 0. \tag{2.4}$$

By applying the product rule to Eq. (2.4), we obtain[9]

$$\Omega_2(E_2)\frac{d\Omega_1(E_1)}{dE_1} + \Omega_1(E_1)\frac{d\Omega_2(E_2)}{dE_2}\frac{dE_2}{dE_1} = 0. \tag{2.5}$$

The system has total energy $E = E_1 + E_2$, which is fixed and hence $dE_1 = -dE_2$. Equation (2.4) therefore becomes

$$\Omega_2(E_2)\frac{d\Omega_1(E_1)}{dE_1} - \Omega_1(E_1)\frac{d\Omega_2(E_2)}{dE_2} = 0, \tag{2.6}$$

and hence

$$\frac{d\ln\Omega_1(E_1)}{dE_1} = \frac{d\ln\Omega_2(E_2)}{dE_2}. \tag{2.7}$$

This implies that the most probable distribution of the energy of two systems in thermal contact is such that the energy derivative of the logarithm of the number of microstates in the two systems is equal. Eq. (2.7) gives the condition for the most likely division of energy between the two systems, which are allowed to exchange energy. When two systems are allowed to exchange energy and this satisfies Eq. (2.7), we say that "The systems are at the same temperature". Therefore, we identify temperature T with $d\ln\Omega(E)/dE$. We define temperature as

$$\beta = \frac{1}{k_B T(E)} = \frac{\partial\ln\Omega(E)}{\partial E}. \tag{2.8}$$

were k_B is the Boltzmann constant.

Suppose system 1 is in state i with energy E_i, clearly system 2 has an energy $E_2 = E - E_i$ and the degeneracy of system 2 is given by $\Omega_2(E - E_i)$. The conditional probability P_i to find system 1 in state i is:

$$P_i = \frac{\Omega_2(E - E_i)}{\sum_j \Omega_2(E - E_j)}. \tag{2.9}$$

Applying Taylor expansions to $\ln \Omega_2(E - E_i)$ and $\Omega_2(E - E_j)$ around $E_i = 0$, we get that $\ln \Omega_2(E - E_i) = \ln \Omega_2(E) - \frac{\partial \ln \Omega_2(E)}{\partial E}\epsilon + \mathcal{O}(\epsilon^2)$ for some ϵ in $]E, E_i[$. A similar expression holds for $\ln \Omega_2(E - E_j)$. Combining this with Eq. (2.8), we can write

$$P_i = \frac{\exp(-E_i/k_B T)}{\sum_j \exp(-E_j/k_B T)}, \tag{2.10}$$

which is the well-known Boltzmann distribution. Now that the energy distribution of the canonical system has been established, we can calculate the average energy $\langle E \rangle$ simply by $\langle E \rangle = \sum_i E_i P_i$. If we want to know the average value of an observable A, we can also apply Eq 2.10 to get $\langle A \rangle = \sum_i A_i P_i$

$$\langle A \rangle = \frac{\sum_i \exp(-E_i/k_B T) A_i}{\sum_j \exp(-E_j/k_B T)}, \tag{2.11}$$

where A_i denotes the expectation value of the observable A in state i. In classical mechanics a system of N particles, the total energy of a system is expressed as the sum of the total kinetic energy $\mathcal{K}$ and the potential energy $\mathcal{U}$. Hence, a system is described by the Hamiltonian $\mathcal{H}(\mathbf{p}^N, \mathbf{r}^N) = \mathcal{K}(\mathbf{p}^N) + \mathcal{U}(\mathbf{r}^N)$, where $\mathbf{r}^N$ is shorthand for the set of position coordinates, i.e. $\mathbf{r}^N = \{\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_n\}$ and $\mathbf{p}^N$ the shorthand for the set of momenta coordinates, i.e. $\mathbf{p}^N = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n\}$. The combination $(\mathbf{r}^N, \mathbf{p}^N)$ is called a phase point in a 6N-dimensional phase space. In time a system moves from one phase point to another phase point in a continuous way, which is called a phase trajectory. Therefore we may integrate over the continuous phase space instead of summing the Boltzmann factor over discretely spaced energy state. Using $\mathcal{H}$ and the phase points $(\mathbf{r}^N, \mathbf{p}^N)$ we rewrite equation Eq. (2.11)[4] as:

$$\langle A \rangle = \frac{\int \mathrm{d}\mathbf{p}^N \mathrm{d}\mathbf{r}^N A(\mathbf{p}^N, \mathbf{r}^N) e^{-\beta \mathcal{H}(\mathbf{p}^N, \mathbf{r}^N)}}{\int \mathrm{d}\mathbf{p}^N \mathrm{d}\mathbf{r}^N e^{-\beta \mathcal{H}(\mathbf{p}^N, \mathbf{r}^N)}} \tag{2.12}$$

The ensemble averaged A has been expressed as function of coordinates $\mathbf{p}^N$ and $\mathbf{r}^N$. The factor $\exp\left(-\beta \mathcal{H}(\mathbf{p}^N, \mathbf{r}^N)\right)$ is known as the Boltzmann-factor. Solving these integrals analytically can only be done in some exceptional cases. Therefore numerical methods are needed to get an approximate solution for $\langle A \rangle$. Using quadrature rules for approximating this integral is unpractical, as such schemes suffer from large statistical errors and require too many sample points.

Another approach to solve the integrals is by using importance sampling. Importance sampling is an integration technique, which we will explain here for a

one dimensional (1D) integration. The one dimensional integral $\int_a^b f(x)\mathrm{dx}$ can be written as $\int_a^b f(x)\mathrm{dx} = (b-a)\overline{f(x)}$, where $\overline{f(x)}$ is the average of $f(x)$ on the domain $[a,b]$. By sampling random points $x_i \in [a,b]$, we get an approximation for $\overline{f(x)}$.

$$\overline{f(x)} = \frac{1}{L}\sum_{i=1}^{L} f(x_i), \tag{2.13}$$

with L the number of sampling points. These points can be uniformly distributed on $[a,b]$, but other choices are possible.
The integrals of Eq. (2.12) are such that, when subjected to importance sampling techniques, many contributions to the average are zero, which leads to slow convergence to the desired ensemble average. By selecting a proper distribution function for the sample points, such that most contributions to the average are nonzero in the importance sampling, faster convergence can be obtained. Here we demonstrate this for a weight function $w(x)$ in the 1D case. Applying weighted importance sampling to a one dimensional integral, we may write (Ref [4]),

$$I = \int_a^b f(x)\mathrm{dx} = \int_0^1 \frac{f[x(u)]}{w[x(u)]} \approx \frac{1}{L}\sum_{i=1}^{L} \frac{f[x(u_i)]}{w[x(u_i)]}, \tag{2.14}$$

where $w(x)$ is normalized and $u'(x) = w(x)$. Unfortunately for the case of the integrals in Eq. (2.12), a weight function such that contributions of the Boltzmann-factor is nonzero cannot easily be found. All roads to an acceptable solution therefore appear blocked. However, this problem can be overcome by considering the entire average instead. We are only interested in observables, which do not depend on the momenta of the particles.

$$\langle A\rangle = \frac{\int \mathrm{d}\mathbf{r}^N \exp\left(-\beta\mathcal{U}(\mathbf{r}^N)\right) A(\mathbf{r}^N)}{\int \mathrm{d}\mathbf{r}^N \exp\left(-\beta\mathcal{U}(\mathbf{r}^N)\right)}, \tag{2.15}$$

where we have dropped the $\mathbf{p}$ dependence. By writing $\mathcal{N}(\mathbf{r}^N)\mathrm{d}\mathbf{r}^N$

$$\mathrm{d}\mathbf{r}^N\mathcal{N}(\mathbf{r}^n) \equiv \frac{\mathrm{d}\mathbf{r}^N \exp\left(-\beta\mathcal{U}(\mathbf{r}^N)\right)}{\int \mathrm{d}\mathbf{r}^N \exp\left(-\beta\mathcal{U}(\mathbf{r}^N)\right)}, \tag{2.16}$$

we denote the probability of finding the system in a region $\mathrm{d}\mathbf{r}^N$ around $\mathbf{r}^N$ a region of size $\mathrm{d}\mathbf{r}^N$. Suppose now that we are somehow able to randomly generate points in configuration space according this probability $\mathcal{N}(\mathbf{r}^N)$. On average the number of points n_i generated per unit volume around point $\mathbf{r}^N$ is equal to $L\mathcal{N}(\mathbf{r}^N)$, where L is the total number of points generated. In other words:

$$\langle A\rangle = \int \mathrm{d}\mathbf{r}^N A(\mathbf{r}^N)\mathcal{N}(\mathbf{r}^N) \approx \frac{1}{L}\sum_{i=1}^{L} A(\mathbf{r}_i^N), \tag{2.17}$$

where in the last step we have used importance sampling. In contrast with regular importance-sampling, we are now only sampling through configurations which have nonzero contribution to the integral. This type of sampling is called the Metropolis scheme.
We will give now a brief and less abstracted explanation about the idea of the Metropolis scheme and apply this to our problem. Suppose we want to know the average depth of a non-rectangular shaped pond (Fig. 2.1a). We can simply put a homogenous rectangular grid over the pond and measure the depth on each position in the water corresponding to each point on the grid (Fig. 2.1b). After measuring all depths, we calculate the average depth by simply calculating the mean value. If we know the shape of the pond beforehand we can construct the grid in such a way that all grid points lie in the pond. When the space between the grid points is too big, it is possible that we miss local depths.
Now suppose we do not know anything about the pond and finding a suitable grid is impossible. In this case it is still possible to find an acceptable average value for the depth of the pond. We simply jump in the pond somewhere and measure the depth and update the average value, which is of course the same in the beginning. Next we make a step of fixed size in a random direction. If we get on land, we go back to the same point in the water and measure the depth again and contribute this value to the average (Fig. 2.1c).

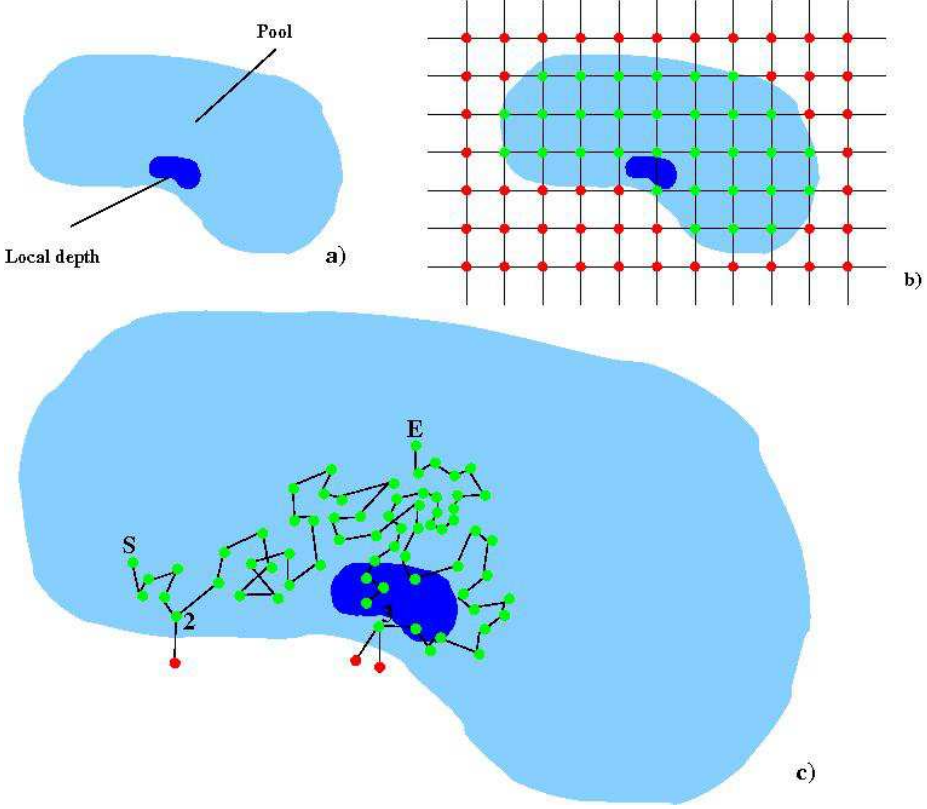

Figure 2.1: a) A pond with a local depth. b) We can put a rectangular grid over the pond and measure the value at each point on the grid. The red dots represent points above land with contribution zero to the average. Green points are above the water with nonzero contribution to the average. Clearly there are no points above the local depth. c) We make a random walk through the pond and we start at a random point S and end at some point E. If we step on land we go back to our last point and measure the depth again. This can happen many times in a row.

On the other hand if we are still in the water we measure the new depth and update the average with this value. We repeat this procedure until we think the average does not fluctuate too much anymore. We say that the system is in equilibrium if the measured value fluctuates around some value, no matter how long the measurements take place from that moment on. When measuring long enough we may find the local depth and take the measured value into account. Suppose we are in equilibrium and did not find this local depth, then we know that we measured long enough such that a possible contribution of this local depth has no significant influence on the average. The idea of the Metropolis scheme is that we generate configurations, followed by a measurement and the update of the average value of the observable of interest. We start the measurements from the beginning, but we start averaging the values after equilibrium is reached. If the generated configuration is accepted (valid) we use this configuration to update the average and otherwise we reject it and use the

old configuration again to update the average.
We can apply this metropolis scheme to our problem. We start with a configuration o (old) with N particles, which has a nonvanishing Boltzmann factor $\exp(-\beta\mathcal{U}(o))$, where $\mathcal{U}(o)$ is the potential energy of configuration o. Next we generate a new trial configuration n (new) also with N particles, by adding a small configurational displacement Δ to o ($n = o+\Delta$). For the new configuration the Boltzmann factor is $\exp(-\beta\mathcal{U}(n))$. Now we must decide whether we will accept the new configuration or reject it. Many rules can be constructed for this decision, such that the probability of accepting the configuration n is proportional to the Boltzmann weight of the energy difference $\exp(-\beta(\mathcal{U}(n) - \mathcal{U}(o)))$. The transition probability to go from configuration o to n is denoted by $\pi(o \to n)$. The transition matrix must satisfy one condition: Once equilibrium has been reached, that state should not be left. This means that, in equilibrium, the average number of accepted moves that results in leaving system o must be exactly equal to the number of accepted trial moves from all other states n to o. For convenience we apply a stronger condition. Namely, in equilibrium the average number of accepted moves from o to any other state n is exactly canceled by number of reverse moves. This condition is known as detailed balance and implies the following:

$$\mathcal{N}(o)\pi(o \to n) = \mathcal{N}(n)\pi(n \to o). \tag{2.18}$$

In Monte Carlo simulations we first perform a trial move. Let $\alpha(o \to n)$ denote the probability of performing a trial move from state o to state n and let $\mathrm{acc}(o \to n)$ denote the probability of accepting a trial move from o to n. Then we get

$$\pi(o \to n) = \alpha(o \to n) \times \mathrm{acc}(o \to n). \tag{2.19}$$

Here we use the original Metropolis scheme, where α is chosen to be symmetric ($\alpha(o \to n) = \alpha(n \to o)$). Combining Eqs (2.18) and (2.19) we obtain

$$\frac{\mathrm{acc}(o \to n)}{\mathrm{acc}(n \to o)} = \frac{\mathcal{N}(n)}{\mathcal{N}(o)} = \exp\left(-\beta([\mathcal{U}(n) - \mathcal{U}(o)]\right). \tag{2.20}$$

It should be noted that there are more choices for $\alpha(o \to n)$, which satisfy these conditions, but we will not go into these here. The probability of acceptance cannot exceed 1, therefore

$$\mathrm{acc}(o \to n) = \begin{cases} \mathcal{N}(n)/\mathcal{N}(o) & \mathcal{N}(n) \geq \mathcal{N}(o) \\ 1 & \mathcal{N}(o) \geq \mathcal{N}(n) \end{cases} \tag{2.21}$$

With this acceptation rule the basic Monte Carlo Algorithm is given by:

1. Create a valid start configuration C.
2. Apply a small random change to configuration C, $C \to C'$.
3. Check if this configuration C' is accepted by the acceptance rule $\mathrm{acc}(o \to n)$ given above.
4. Continue with C' if the new configuration is accepted otherwise continue with $C' = C$.
5. Add the contribution of C' to the ensemble average.
6. Go back to step 2.

This procedure is continued until sufficient sampling is achieved. Note that (steps 4 & 5) both accepted and rejected moves contribute to the ensemble average. In the next chapter we explain in more detail what a (valid) configuration C is, for systems of our interest how to apply changes to get an configuration C', and what acceptation rules we use.

Chapter 3

Floppy Box Monte Carlo simulations for crystals of Hard Spheres

3.1 Crystal structures

In this chapter we study crystal structures using the Metropolis scheme in the form of the floppy box method [3]. The structure of crystals is periodic and can be broken down into small substructures. The smallest of these substructures which still maintains the morphology of the crystal is the so-called unit cell. For example one grain of salt (NaCL) contains about 10^{18} atoms. However, the smallest substructure consists only of two particles namely one sodium (Na^+) ion and one chlorine (Cl^-) ion in a given configuration. We study this crystal using the floppy box method.
The idea behind the Floppy Box Monte Carlo (FBMC) method is to investigate the entire crystal by studying its unit cell. In FBMC simulations the particles are allowed to move/translate through the simulation box and rotate, if they are anisotropic. The box itself acts as a unit cell and is allowed to deform to determine the optimal configuration. The contribution of these three kinds of moves provides a way for an efficient exploration of configuration space. This method works for all kind of interaction potentials. Here, we start with the most simple potential, namely that of hard spheres.

3.2 The simulation box and hard spheres

We start with a simulation box, which we see as a part of three dimensional (3D) real-space. In this subspace we are going to model a number of particles. Particles/atoms can be seen as (small) objects that occupy space. Particles can be represented as spheres, cubes, tetrahedra, etc or just a point. We start the

simulation by modeling the atoms/particles as a spheres. In classical physics, particles cannot overlap, therefore we do not allow any configurations with an overlap. To achieve this, we introduce an interaction potential. The hard sphere overlap potential [11] $\varphi(r_{ij})$ is defined as

$$\varphi(r_{ij}) = \begin{cases} 0 & d(B_i, B_j) \geq r_i + r_j \\ \infty & d(B_i, B_j) < r_i + r_j \end{cases}, \tag{3.1}$$

where $d(\mathbf{x}, \mathbf{p}_i)$ is simply the Euclidean distance between the vectors $\mathbf{x}$ and $\mathbf{p_i}$, thus $\|\mathbf{x} - \mathbf{p_i}\|_2$ with $\|.\|_2$ the standard Euclidean vector norm. This potential is zero in case there is no overlap between any two distinct particles and infinity otherwise. At this point a configuration can be seen as a number of marbles in a box which are not moving and without gravity. As explained before, we restrict our simulation to a small number of particles which are in a periodic structure, therefore a finite-sized simulation box with periodic boundary conditions suffices to describe an otherwise infinite system. The latter is convenient, because we use a computer for the simulations, which has finite memory and processor speed. The simulation box, which we refer as $V \in \mathbb{R}^3$, is the parallelepiped spanned by three basis vectors $\mathbf{v}_1, \mathbf{v}_2$ and $\mathbf{v}_3$.
We consider a system of N hard spheres $B(\mathbf{p}_i; r_i)$ with centers $\mathbf{p}_i$ and radii r_i with i an index. Here we define a sphere as follows: $B_i = B(\mathbf{p}_i; r_i) := \{\mathbf{x} \in \mathbb{R}^3 | d(\mathbf{x}, \mathbf{p}_i) \leq r_i\}$. For a configuration of hard spheres to be valid, it follows from the overlap potential, that we require $\bigcap_{0 \leq i < \infty} B(\mathbf{p}_i; r_i) = \emptyset$. A configuration $C = C(V; B)$ consists of a basis V (the box) and a number of hard-spheres B_i which all lie in the box V. With this configuration, we completely describe the crystal system. The center of a sphere $\mathbf{p}_i \in \mathbb{R}^3$ can be written in terms of the basis vectors V. Let $\mathbf{p}'_i$ be a center in V-space then we get:

$$\mathbf{p}_i = V\mathbf{p}'_i = \sum_{j=0}^{2} \alpha_j v_j, \text{ with } \alpha \in [0, 1[. \tag{3.2}$$

Let $\mathbf{p}_i$ be the midpoint of a sphere B_i in the simulation box V. Then the images $B_j^{w_1,w_2,w_3}$ induced by the periodic boundary conditions are copies of the sphere B_i with midpoint $p_i + w_1\mathbf{v}_1 + w_2\mathbf{v}_2 + w_3\mathbf{v}_3$, with $w_i \in \mathbb{Z}\backslash\mathbf{0}$, where $\mathbf{0}$ is the zero vector. Note that when $w_1 = w_2 = w_3 = 0$ we have the original sphere. All images of a sphere B_i are denoted as

$$B_i^{w_1,w_2,w_3} = B(p_i + \sum_{w=0}^{2} w_i\mathbf{v}_i; r_i), \; w_i \in \mathbb{Z}\backslash\mathbf{0} \tag{3.3}$$

With Eq. 3.3 we can redefine the simulation box as $C(V, B^{0,0,0})$, where $B^{w_1,w_2,w_3} = \bigcup_j B_j^{w_1,w_2,w_3}$. It is possible that a sphere B_i has an overlap with one of its own images $B_i^{w_1,w_2,w_3}s$ or with an image of other sphere $B_j^{w_1,w_1,w_3}$ $(j \neq i)$. We define the entire crystal as $\bigcup_{j,w_i \in \mathbb{Z}} B_j^{w_1,w_2,w_3}$. The crystal structures we are studying are infinite of size. To be more precisely the crystal consists of the particles in

the box and their periodic images. If we want to check if there is an overlap within the crystal, we have to keep this in mind. First we check if there is any overlap between the particles within the box. If there is an overlap then potential U goes to infinity and therefore the configuration is rejected, because of the acceptance rule Eq (2.20). If there is no overlap in the box, we check if a particle in the box may have an overlap with one of its own periodic images. Like before, the configuration is rejected in case of an overlap. When there are still no overlaps, we check if a particle in the box has an overlap with a periodic image of an other particle. If we have performed the three overlap checks and there is still no overlap, then we conclude that there is no overlap in the whole crystal structure, since the three individual checks are the same as a check of the whole system. In the next sections we go into more details about the different overlap checks and how to implement these efficiently.

3.2.1 Overlap of particles within the box

In figure 3.1 we plotted a simulation box (grey rectangle) with four particles and the eight direct neighboring images, in which the particles are indicated in green. First we check if there is no overlap between the spheres within the box of a given configuration (red particles in the grey rectangle). To do this, we simply loop over all combinations of spheres B_i and B_j with $i \neq j$ and check if distance between the two spheres is smaller than the sum of the radius of both spheres Eq (3.1). If we find an overlap, we can stop with all the overlap checks and reject the configuration.

3.2.2 Overlap of a particle with own periodic images

Suppose we have a simulation box with basis V and one sphere $B(\mathbf{p}, r)$ (figure 3.2a). When the simulation box is small compared to the radius r of sphere B, it is possible that the sphere has an overlap with one of its images B^{w_1,w_2,w_3} (figure 3.2b). To find a possible overlap, we have to find the minimum distance between B and one of its images. First we note that two spheres with radius r have an overlap whenever the midpoints are a distance smaller than $2r$ apart. Without loss of generality we can place sphere B in left lower corner of the simulation box V, in essence translating the entire crystal. We can thus write $\mathbf{p} = (0, 0, 0)$. Now we put a larger sphere B' with the midpoint in the origin and radius $2r$ (figure 3.3a), all spheres with radius r and midpoint outside B' do not overlap with N, because the distance to the origin exceeds $2r$. We therefore only need to check the images against B, which lie within B space. It proves easier and faster to determine which images lie inside a cube however. Therefore we center a cube on the origin with edge length $4r$ (figure 3.3b).

The time to check the extra images we take into account by using a cube rather than a sphere is weighted against the time we save by computing which images lie within the cube rather than within the sphere. The coordinates c_i of these cubes are $c_1 = (2r, 2r, 2r)$, $c_2 = (2r, -2r, 2r)$, $c_3 = (-2r, 2r, 2r)$, $c_4 = (-2r, -2r, 2r)$, $c_5 = (2r, 2r, -2r)$, $c_6 = (2r, -2r, -2r)$, $c_7 = (-2r, 2r, -2r)$ and

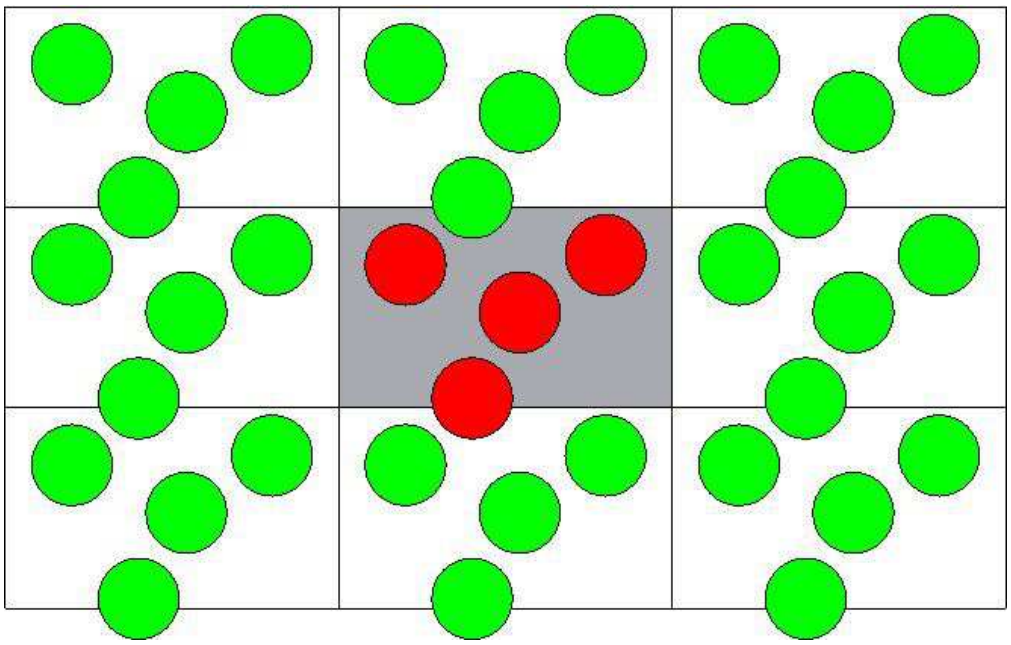

Figure 3.1: An illustration of a simulation box (grey rectangle) with four particles in it. (indicated in red) For simplicity we represent the system here as a 2D set of disks in a square box, but the siutation for 3D systems of spheres in a parallelepipedal box is analogous. In addition 8 periodic images of the original box are shown. These boxes are white and the image particles are indicated in green. The system does not have overlap.

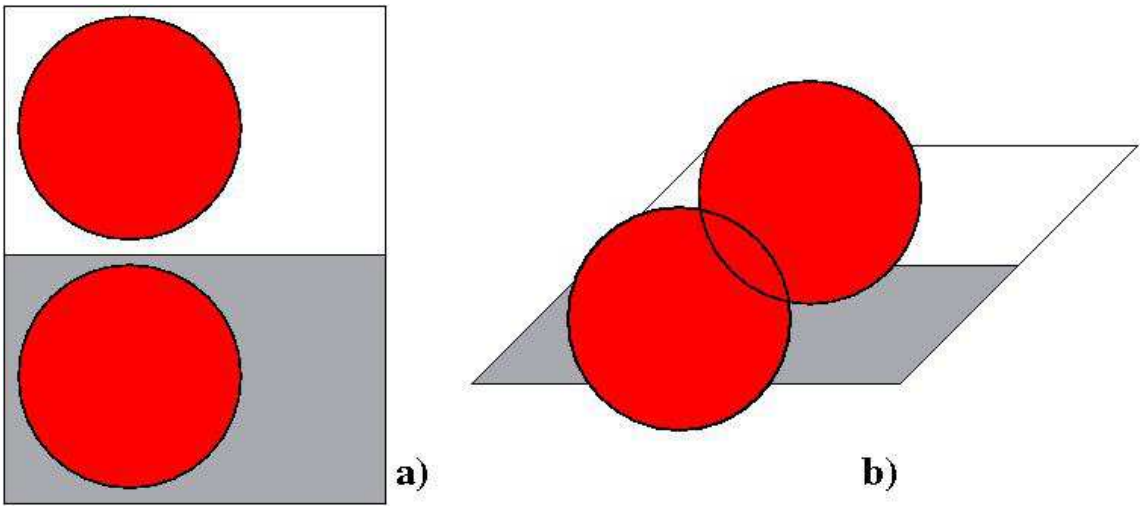

Figure 3.2: a) An illustration of a 2D simulation box (grey rectangle) with one particle in it and one neighboring image (white rectangle). There is no overlap in this system. b) A depiction of the same simulation box after a deformation. Clearly there is an overlap in the system and according to the acceptatance rule, this system is not valid.

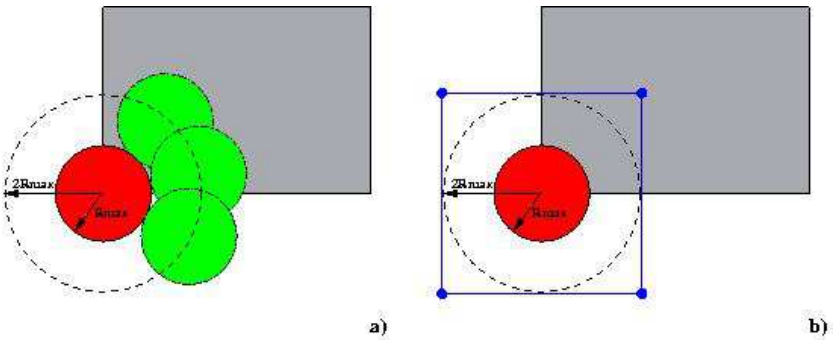

Figure 3.3: a) A 2D disk representation of a sphere with radius R_{max} in the origin does not have an overlap with spheres that have a distance to origin that exceeds $2R_{max}$. b) We must check all the copies of the simulation box that may have images of spheres that lie within the radius $2R_{max}$. To determine which copies we need, we work with the square around the sphere with radius $2R_{max}$, rather then spheres itself.

$c_8 = (-2r, -2r, -2r)$. The coordinates R_i are the coordinates in the real space, the coordinates in V-space can be easily calculated by applying the reverse transformation $R'_i = V^{-1}R_i$. With the V-space coordinates R'_i we can easily determine the upper and lower bounds of w_1,w_2 and w_3, which bound all the images we need to check. The upper and lower bound for w_j can be determined by

$$\begin{aligned} w_j^{low} &= \lfloor \min_{1 \le i \le 8} (R'_i \cdot e_j) \rfloor \\ w_j^{up} &= \lceil \max_{1 \le i \le 8} (R'_i \cdot e_j) \rceil, \end{aligned} \tag{3.4}$$

where e_j represents the unit basis vector with element j equal to one, $\lfloor . \rfloor$ denotes the floor function and $\lceil . \rceil$ denotes the ceiling function. For example, when $j = 1$ we get that $e_j = (1, 0, 0)$ and w_j^{low} will be the minimum x-coordinate of all R'_i rounded to the largest integer not smaller than x. Consequently w_j^{up} will be the maximum x-coordinate of all R'_i rounded to the smallest integer greater than x. For a system containing multiple spheres we could calculate the upper and lower boundary for every sphere B_i in the box for the self overlap check. However to reduce computational time of generating such an image list, we only do this for the worst case: The sphere B_i with the largest radius r_i will have the largest surrounding cube. We will create a self image list (SIL) based on this radius $r_{max} = \max_{0 \le i < n} r_i$ and use this for every sphere in the box. Again the extra checks weigh against the computational time required to make smaller list.

$$\text{SIL} \subset J,\ J = \{V\mathbf{x} | \mathbf{x} \in \mathbb{Z}^3 \backslash \mathbf{0}\}, \tag{3.5}$$

To be more precisely SIL $\subset J$ with $w_j^{low} \le \mathbf{x}e_j \le w_j^{up}$.

3.2.3 Overlap with images of other particles

Suppose we have a simulation box V with more than one particle B_i, $i > 1$. Then there is a possibility of an overlap between between particle B_i and one of the images of an other particle $B_j^{w_1,w_2,w_3}$. Obviously there is no need to check overlap between two images. because there will be an equivalent overlap between two particles or between a particle and an image of an other particle (Figure 3.5.).

Suppose with a sphere with the largest radius B_i fixed in the left lower corner of the simulation box and another sphere B_j with the same radius somewhere else in the box. When does these two spheres do have an overlap? It depends of course on the position of sphere B_j. Like in the case of the SIL, the spheres has an overlap whenever the midpoints are a distance smaller than the sum of the radii of both spheres. When sphere B_j is somewhere in the right upper corner of the box, it may have an overlap with a periodic image of sphere B_i.

To determine which images we have to take into account for the overlap check, we still can apply the same strategy as for the SIL image list (Fig 3.6a). The only difference is that we put the sphere B_i with the largest radius r_{max} in every corner of the simulation box V. Thus not only the left lower corner $\mathbf{p}'_i = (0, 0, 0)$

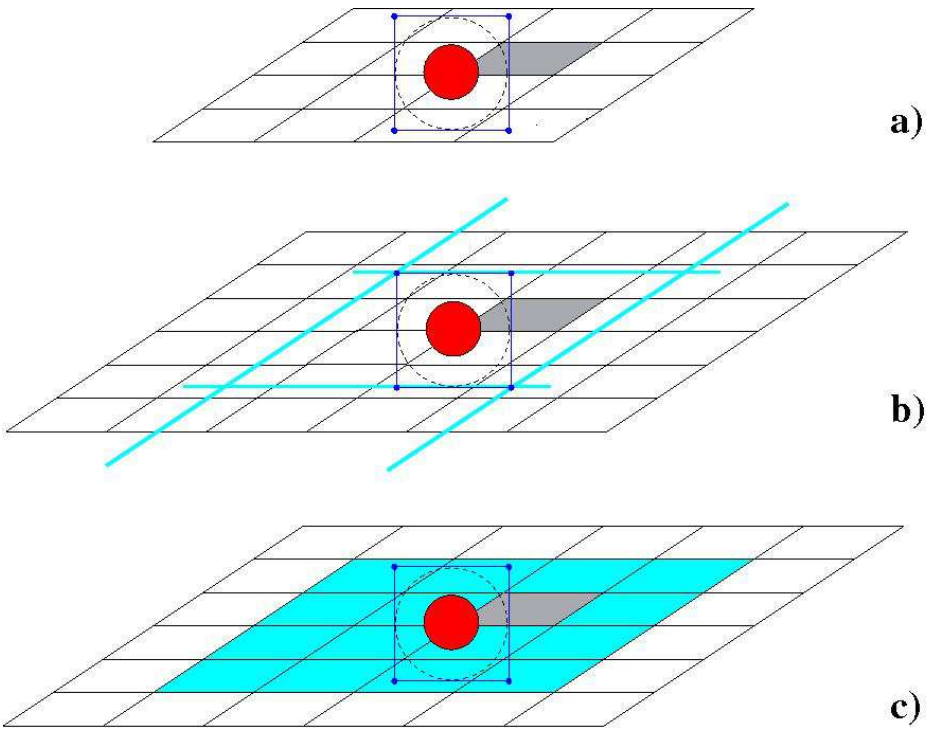

Figure 3.4: A recipe for determining the (self image list) SIL in 2D. a) First translate the sphere with the maximum radius to the origin and put a cube in the origin with edges of size $2r$. b) The edges of the cube determines a boundary. Images of the spheres outside this boundary do not have an overlap. c) With the boundary from B we can easily determine which copies of the simulation box we have to check. In a similar way the SIL list is created in 3 dimensions.

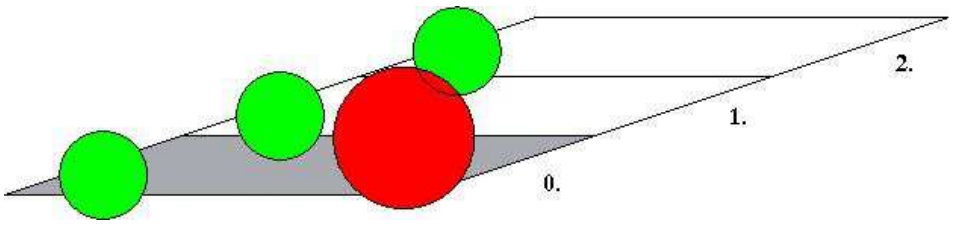

Figure 3.5: A 2D simulation box with two spheres. The largest sphere (red) has an overlap with an image of the smallest sphere (green). The overlap is not with the image in nearest copy of the simulation box (box 1), but with an image in a copy one further (box 2).

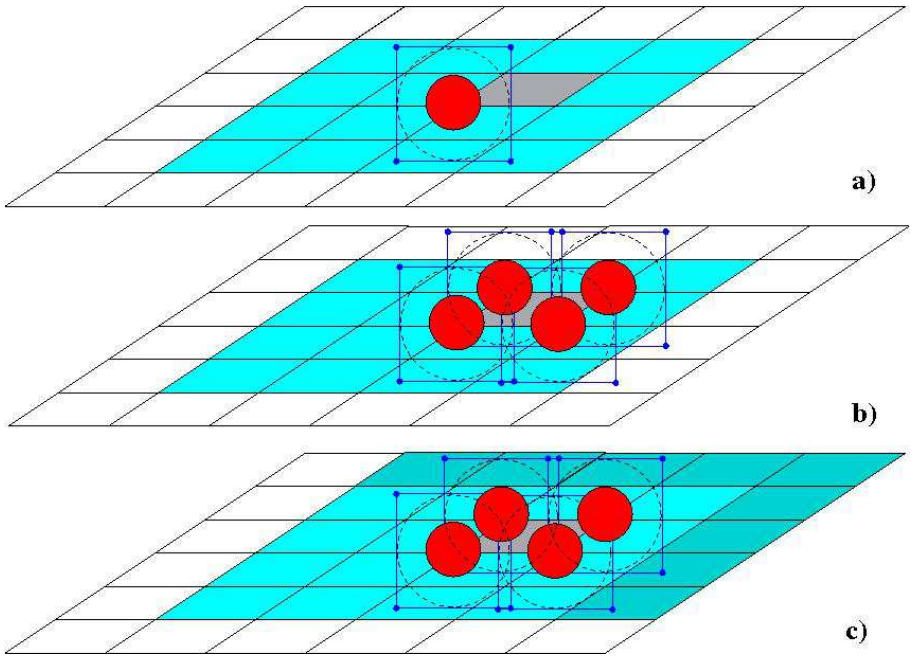

Figure 3.6: To determine the number of images we have to check for an overlap between two particles, we apply the same strategy as in the case of an overlap of a sphere with one of its own periodic image. a) We place the sphere with the largest radius in the left lower corner of the box and find the enclosing box of a sphere with twice the maximum radius. b) We repeat the same procedure in each corner. c) The number of images that have to be checked are the same images as in case of an self overlap, only one row/column of images is added in each direction.

, but also $(1,0,0)$, $(0,1,0)$, $(1,1,0)$, $(0,0,1)$, $(1,0,1)$, $(0,1,1)$ and $(1,1,1)$. We can again put a larger sphere B_i' with radius $2r$ in each corner of the simulation box (Fig 3.6b)..

These spheres are the corner of a rounded parallelepiped. The extrema of this parallelepiped can be determined by constructing an orthogonal basis out of V. A much easier way is to put cubes R_i around all this larger spheres B_i' like we did in the self overlap case. All of the cubes R_i must have the same orientation. We only have to determine the bounds for a self overlap which we already did and increase the upper bounds with one (Fig 3.6c). By extending the SIL list, we get a OIL (Other image list) list.
On occasion we perform an extra check to be sure that the SIL and OIL list indeed covers the space in which there can be a self or other overlap. For this we use the $\mathrm{E}-\mathrm{OIL}$ lists, which are extended versions of OIL list expanding the boundaries w_i by two in each direction. In principle this should not be necessary, but for expensive simulations catching bugs early on can save a lot of time.

3.3 Basic moves

In a MC simulation of hard spheres we want to create a new trial configuration in order to sample trough phase space as explained with the Metropolis scheme. To sample phase space, require two basic moves

- Translational moves of the particles.
- Volume moves.

The volume move involves a change in the shape and the volume of the box simultaneously

3.3.1 Translational moves

To move a sphere we randomly select one sphere $B_i(\mathbf{p}_i; r_i)$ from the simulation box $C(V, B^{0,0,0})$ and translate it with a random vector $\mathbf{x} = (x_1, x_2, x_3)$, where x_i is a random number homogeneously generated between $-x_{max}$ and x_{max}. We thus obtain the updated position vector $\mathbf{p}_i^{new} = \mathbf{p}_i^{old} + \mathbf{x}$. After this translation we apply the periodic boundary conditions to get the new configuration C'. In the case that a sphere exceeds the boundary of the simulation box V, at least one of the α_j's of Eq (3.2) will be larger than one or smaller than zero.
The procedure to get the sphere in the simulation box again is to reduce α_j by an integer such that $0 \leq \alpha_j < 1$. Suppose we have a sphere B_i with one or more $\alpha_j \geq 1$, then we subtract 1 from the α_j until $\alpha_j < 1$. Similar 1 is added when $\alpha_j < 0$. After this procedure the coordinates of the sphere in real space are updated.

The validity of the move depends on the acceptance rule $acc(C \to C')$ for a particle movement.

$$acc(C \to C') = e^{-\beta(\mathcal{U}(C)-\mathcal{U}(C'))}. \tag{3.6}$$

The potential function $\mathcal{U}(C) = \sum_{\forall i \neq j} \mathcal{U}(r_{ij})$ is simply the hardsphere potential.

$$\begin{aligned} \mathcal{U}(r_{ij}) &= \begin{cases} 0 & d(B_i, B_j) \geq r_i + r_j \\ \infty & d(B_i, B_j) < r_i + r_j \end{cases} \Rightarrow \\ acc(C \to C') &= \begin{cases} 0 & \exists i, j \; d(B_i, B_j) \geq r_i + r_j \\ 1 & \forall i, j \; d(B_i, B_j) < r_i + r_j. \end{cases} \end{aligned} \tag{3.7}$$

3.3.2 Volume move

The volume of the simulation box is determined by the basis V, as $|\mathrm{Det}(V)|$. Altering the volume of the box can be done by selecting randomly one of the basis vectors v_i and alter one of its components randomly by adding a small amount Δv_i with $v_{min} \leq \Delta v_i \leq v_{max}$. Adding a small amount Δv_i leads to a volume change Δvol.
Like a particle move we must obey detailed balance, and thus $v_{min} = -v_{max}$. The new volume becomes $\mathrm{vol}_{new} = \mathrm{vol}_{old} + \Delta\mathrm{vol}$. The acceptance of this volume change in the NPT ensemble depends on the following acceptance rule: [4]

$$\begin{aligned} acc(o \to n) = \min(1, \exp\{-\beta[&\mathcal{U}(\mathbf{s}^N, V') - \mathcal{U}(\mathbf{s}^N, V) + \\ &P(V' - V) - (N+1)\beta^{-1} \ln(V'/V)]\}). \end{aligned} \tag{3.8}$$

The potential $\mathcal{U}$ of a valid system with hard spheres is always zero. Because checking for overlaps is expensive we first preliminarily accept or immediately reject the configuration on the basis of the following criterion: Eq. (2.11), where we assume that there are no overlaps. After this we check to see if there are in fact no overlaps, if there are we reject the move. If there are no overlaps, we change the preliminary acceptance into a definite acceptance. Note that this is the same as calculating (3.8) in one step. Only far less expensive on average. Therefore we can simplify the first part of the acceptance check to the following rule

$$acc(o \to n) = \min(1, exp\{-\beta[P(V' - V) - (N+1)\beta^{-1} \ln(V'/V)]\}). \tag{3.9}$$

When the volume $V \to V'$ change is accepted, the coordinates p_i' in V are the same, as the p_i in V', but the coordinates of spheres in real space have changed. Therefore we have to recheck if there is any overlap after we transform the V' coordinates to V. Before we do a check for any overlap, we must first update the image lists again, because the shape of V has changed.
When the volume change is accepted and there is no overlap of any kind, the configuration C with the changed box V, will be the new configuration. Otherwise we continue with the old configuration and its image lists. It becomes clear that the volume change is the most time consuming move of all basic moves in

the simulation. In a MC simulation with a system of N particles we perform N trial moves in each MC cycle. We perform $(N-1)/N$ translational/rotational moves and $1/N$ volume moves on average. The ratio between the number of translations and rotations is 0.5 on average in the case of spheres. After a multiple volume changes, the volume of the box V can become small, while the surface gets larger. This means that the box is getting flatter and longer. This can lead to a very large image list. With lattice reduction, we can overcome this problem.

3.4 Lattice reduction

Suppose we have a valid configuration $C(V, B)$. We may always change the simulation box V to a differently shaped box W by taking linear combinations of the basis vectors of V to obtain a basis of W. The system these boxes generate $C(V, B)$ and $C(W, B)$ respectively is the same; we merely rewrite the system in terms of an equivalent basis. Suppose that we can find a box W which is more cubic then V, then the size of SIL and OIL becomes smaller. The number of images $B_i^{r,s,t}$ we have to check is smaller in case of a rectangular box than in the case of a box that is very parallelepipedal. Lattice reduction is a procedure by which more rectangular basis W may be obtained [5]. This procedure is as follows.

Let P be the matrix defined as follows:

$$P = \begin{cases} P_{ij} = 1, & i = j \\ P_{ij} = 1 \vee -1, & \exists i \neq j,\ i, j \in 0, 1, 2 \end{cases} . \tag{3.10}$$

This matrix generates a linear combination of basis vectors. Let $V = \{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ be the matrix spanned by the box vectors. If, for example $P_{1,2} = 1$, then

$$P = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \text{ and } W = VP = \{\mathbf{v}_1 + \mathbf{v}_2, \mathbf{v}_2, \mathbf{v}_3\} \tag{3.11}$$

There are twelve of such matrices P in total, each one with 1 nonzero off-diagonal element. Note that this is therefore a norm preserving operation: $|det(P)|$=1. Taking $W = VP$, we get that $|det(W)| = |det(VP)| = |det(V)det(P)| = |det(V)|$. This assures that the volume does not change after a reduction. We simply have to check if any box $W = VP$ has a smaller surface and we use this as our new basis, because a smaller surface area for a fixed volume implies a more rectangular shape. We take the new combination of basis vectors which has the smallest surface area. With this new basis, we can repeat the procedure again. If we can not find a basis with a smaller surface any more, we stop the procedure. In principle we check all bases $W = VP_1P_2 \ldots P_N$. Note that there is a possibility that we find that the smallest basis has an equal surface area to the basis we used to generate it. This means that the procedure described

can get stuck in an infinite loop. Therefore we chose a maximum number of iterations at which we stop our lattice reduction procedure.
In a volume move only a small change is applied to the box V, therefore it is not necessary to perform a reduction after each volume move. A reduction-norm $\|.\|_r$ is used to check when it necessary to perform a reduction.

$$\begin{aligned}\|V\|_r &= \frac{\text{Length}(\mathbf{v}_1) + \text{Length}(\mathbf{v}_2) + \text{Length}(\mathbf{v}_3)}{3} \cdot \frac{1}{6} \cdot \frac{\text{Surface}(V)}{\text{Volume}(V)} \\ &= \frac{\|\mathbf{v}_1\| + \|\mathbf{v}_2\| + \|\mathbf{v}_3\|}{3} \cdot \frac{1}{6} \cdot \frac{\|\mathbf{v}_1 \times \mathbf{v}_2\| + \|\mathbf{v}_2 \times \mathbf{v}_3\| + \|\mathbf{v}_3 \times \mathbf{v}_1\|}{|\det(V)|} \end{aligned} \tag{3.12}$$

The norm has been scaled such that a cube has $\|V\| = 1$. From empirical analysis it follows that as long as $\|V\|_r < 1.5$, a lattice reduction is not necessary. Note that Equation (3.12) is not a norm in mathematical sense, because the positive definite property fails ($\|V\| > 0 \Rightarrow V \neq 0$ and $\|V\| = 0 \Leftrightarrow V = 0$).

3.5 Results

We start our first computer simulations with a simple system of one sphere with diameter $\sigma = 1$ in a rectangular box. We apply the FBMC method to this system with a pressure that is high enough to form a crystal structure, i.e. $\beta P\sigma^3 = 14$. On the left side of Fig. 3.7 we see the simulation box after 10^5 MC cycles. The shape of box clearly changed from rectangular to something that look more parallelepiped. By copying this simulation box three times in the direction of each combination of the basis vectors of the simulation box V, we get a system that looks like a FCC crystal structure, which is depicted on the right side of Fig. 3.7.

We run multiple simulations with the same starting configuration and parameters. Although we start every simulation with the exact same configuration and parameters, the resulting crystal structure can be different each time. For example in Fig. 3.8 we see a FCC crystal structure at the top and a BCC structure at the bottom of the figure. Both are the results of simulations with exactly the same starting configuration and parameters.

Systems can also consist of multiple spheres. We define the *ratio* of a system of spheres as $ratio = \frac{\sigma_s}{\sigma_r}$, where σ_s is the sphere with the smallest diameter in the box and σ_r the sphere with largest diameter in the box. Next we look at systems with two spheres in the simulation box. Because we have only two spheres in the box, we can label the diameters of the spheres σ_s and σ_l respectively.
First we look at a system with two spheres with equal diameters $\sigma_s = \sigma_l = 1$ in the box. Like in the case of one single sphere in the box we apply the FBMC method with $\beta P\sigma^3 = 14$. After 10^5 MC cycles we get a BCC crystal structure (Fig 3.9), which is exactly the same structure as in some cases with only one sphere in the box.
For example when we take two spheres in a box with $\sigma_s = 0.4$ and $\sigma_s = 1.0$ ($ratio = 0.4$), we get a crystal structure that looks like NaCl (sodium chloride)

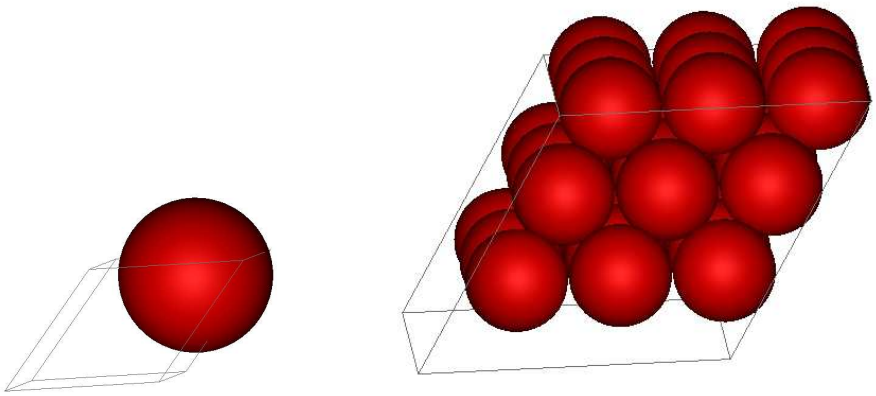

Figure 3.7: On the left side a simulation box is plotted with one sphere $\sigma = 1$) at $\beta P = 14$ after 100k FBMC (80k PR). On the right side is the same simulation box, with 26 images. An FCC crystal structure becomes visible.

(Fig 3.10). Suppose we take two spheres with $ratio = 0.8$ and do different simulations, we can get for example crystal structures that look like CrB, CsCl, etc or structures that can not be identified as a known crystal structure [7]. Crystal structure prediction and crystal structure analysis can be extensive job, therefore we restrict ourself to the above examples. The FCC crystal structure we found with the FBMC method, will be the basis for the Equation of states (EOS) for spheres. The EOS for spheres will be discussed in the next section.

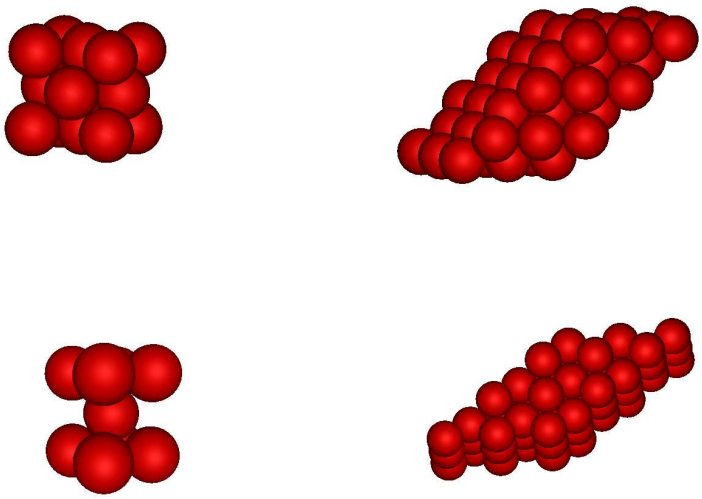

Figure 3.8: Results of FBMC simulations after 10^5 MC cycles with one sphere in a simulation box and $\beta p\sigma^3 = 20$. In the right top corner an FCC crystal structure is observed at the end of the simulation. In the top left corner a part of the same FCC crystal structure. On the bottom right corner a BCC crystal structure became visible after the end of an other simulation with the same parameters and starting configuration. On the bottom left corner a part of the same BCC crystal structure.

3.5.1 Equation of states for hard spheres

In addition, we determine the equation of state (EOS) of hard spheres and compare it with the well-known Carahan-Starling equation of state given by [1]:

$$\frac{P_{cs}}{\rho kT} = \frac{1 + \eta + \eta^2 - \eta^3}{(1-\eta)^3}, \tag{3.13}$$

where $\rho = N/V$ and η is called the packing fraction and is defined as $\eta = \rho V_{sphere} = NV_{sphere}/V$, with $V_{sphere} = \pi\sigma^3/6$ the volume of a sphere with diameter σ. To compare data from simulations with the Carnahan and Starling equation we can not use the FBMC method, because a volume change in the FBMC corresponds to $\Delta V \notin [-\Delta V_{max}, \Delta V_{max}]$ for a chosen value ΔV_{max}. To get a valid EOS we need that $\Delta V \in [-\Delta V_{max}, \Delta V_{max}]$. Therefore we do simple MC simulations of hard spheres in the NPT ensemble. Instead of deformations we only expand or contract the simulation box, which means that the box stays rectangular. To get approximations of the packing fraction, which are comparable with the EOS for infinite sized systems, we require enough spheres

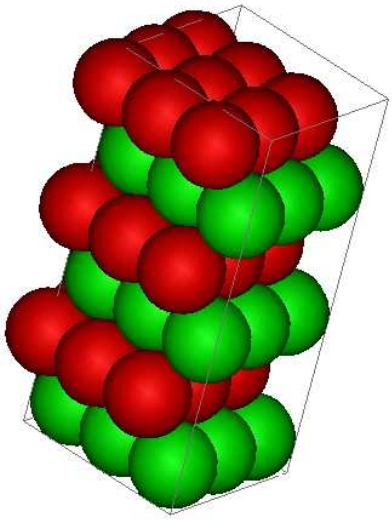

Figure 3.9: BCC Crystal structure with two spheres (ratio = 1) at $\beta P\sigma = 14$ after 100k MC cycles (80k Production).

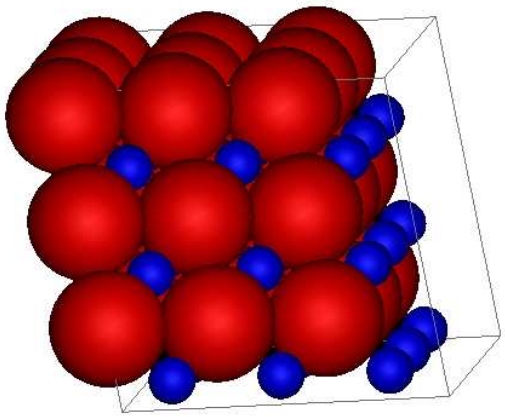

Figure 3.10: A Crystal structure with two spheres(ratio = 0.4) at $\beta P\sigma^3 = 14$ after 100k MC cycles (80k Production). The structure looks like sodium chloride (NaCl)

of equal size. We did our NPT simulations with 128 spheres with diameter $\sigma = 1$ in a rectangular box and measure the average volume or density of the system. We scale the system as follows:

$$\begin{aligned} \beta P &= \rho \, \mathrm{cs}(\eta) \\ \beta P \sigma^3 \frac{\pi}{6} &= \frac{\pi}{6} \sigma^3 \rho \, \mathrm{cs}(\eta) \\ \beta P \sigma^3 \frac{\pi}{6} &= \eta \, \mathrm{cs}(\eta) \\ \beta P \sigma^3 &= \frac{6}{\pi} \eta \, \mathrm{cs}(\eta). \end{aligned} \tag{3.14}$$

We started our simulation with hard spheres such that the system is sparse. We set the pressure at $\beta P \sigma^3 = 0.1$ and the spheres are randomly placed in the box without any overlap. After $4 \cdot 10^4$ equilibrium cycles we start measuring and averaging the volume of the box. The measurements stopped after $6 \cdot 10^5$ production cycles. The average value of the volume at this point, is used to calculate the first point of the EOS.
We continue the simulation by increasing the pressure with a small amount $\Delta(\beta P \sigma^3) = 0.1$. Because we increase the pressure by a small amount, the system is already close to its new equilibrium. Therefore we need less cycles to reach this new equilibrium. The measurement starts again when equilibrium is reached and stops as before after $6 \cdot 10^5$ production cycles. We repeat this procedure until we reach a pressure of $\beta P \sigma^3 = 12$.
The advantage of this procedure is that we need less equilibrium cycles for each point except for the first. Suppose we increase the pressure with a larger value $\Delta(\beta P \sigma^3)$, than the system is less close to the new equilibrium and more cycles are needed to reach it. It is also possible to run different simulations for each value $\beta P \sigma^3$, where each simulation is done by a different processor. The number of equilibrium cycles needed in this case will be larger. At very low pressures the spheres can move freely through the box and the accepted ratio (number accepted moves/number of trial moves) will be close to 1. When the pressure is increased, the volume of the box gets smaller and spheres will be packed close together. As a consequence more moves will be rejected, because the change of an overlap increases. If we start with a sparse system, it is almost impossible to reach equilibrium corresponding to high pressures. The spheres are arranged in the box in such a way that all movements are blocked, although equilibrium is not reached yet. Therefore we start with a crystal structure at high pressure and melt it by slowly decreasing the pressure during the simulation. This prevents us from getting into configurations where the spheres are stuck.
The FCC structure we found earlier with the FBMC method is suitable as a starting configuration. From the previous section we know that the simulation box was deformed when the crystal structure was formed. First we have to put this crystal structure into a rectangular box before we can do a NPT simulation. Next we apply the same strategy as with the sparse configuration, only we now we start at high pressure ($\beta P \sigma^3 = 15$) and decrease the pressure by small $\Delta(\beta P)\sigma^3$. After finishing melting the crystal structure until the pressure

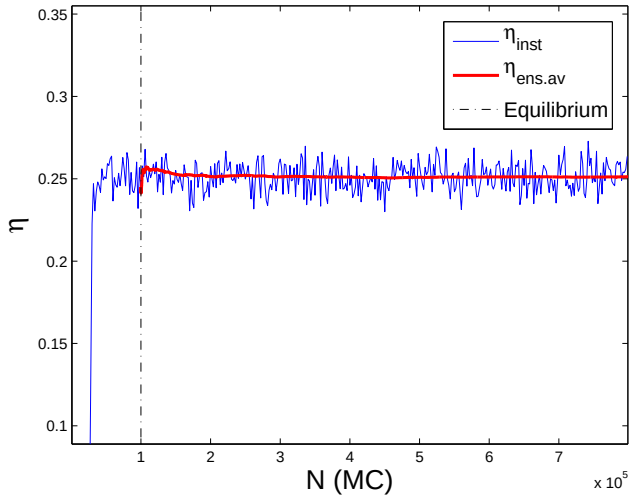

Figure 3.11: The packing fraction of a system with 128 spheres during a simulation of $8 \cdot 10^5$ MC cycles (10^5 equilibrium cycles) with the pressure set to $\beta P \sigma^3 = 2.5$. After about 10^5 cycles the packing fraction oscillates around $\eta = 0.25$ with a relative small amplitude.

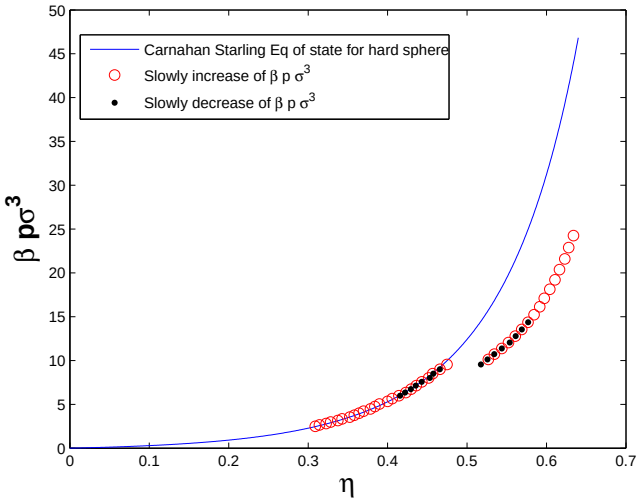

Figure 3.12: The equation of state of a system with 128 hard sphere particles. The first run consists of 400MC (300 production) followed by with runs of 400MC (320 production). The open red circles represents the process in which the pressure is increased slowly. The black dots represent the process of expansion. Clearly around $\beta P\sigma^3 = 10$ there is a jump in the packing fraction η, which indicates a phase transition.

$\beta P\sigma^3 = 7$ we calculate all the corresponding packing-fractions. We plot all points in a figure combined with the theoretical $P_{\text{cs}}(\eta)$ (Fig. 3.12). All points until $\beta P\sigma^3 \approx 10$ lies on the graph $P_{\text{cs}}(\eta)$. From this point there is a jump in the packing fraction, which means that there is a transition from the liquid phase to the crystal phase. The EOS can be used to determine the phase diagram for hard spheres. We will not go further into detail about the phase diagram. Until now we looked at systems with particles represented as individual spheres. In the next chapter we take a brief look at systems with particles that consists of two spheres, the so called dumbbells. Systems with these kind of particles requires an extra move, namely the rotational move. Like systems with spheres we look at some crystal structures and we determine the EOS.

Chapter 4

Hard dumbbells

In the previous chapter we looked at crystal structures based on hard spheres. FBMC Experiments showed BCC and FCC crystal structures based on a simulation box with one or two spheres. We also measured the equation of state for hard spheres. Before we investigate hard superballs, we first take a brief look at dumbbells. A dumbbell is a particle consisting of two spheres. In Fig 4.1 four examples of dumbbells consisting of two spheres of equal and different diameters are shown (fig 4.2a and fig 4.2b). The spheres can be connected to each other in one point or they can have a partial overlap (Fig 4.2c and Fig 4.2d). We define a dumbbell as a set of two spheres by $D = \{B_1, B_2\}$. Similar to a system of hard spheres we can define a simulation box with hard dumbbells as $C(V, D^{0,0,0})$. A sphere is completely symmetrical around its center and therefore all orientations are the same. Dumbbells however can have different orientations, i.e we can put two sphere on top each other in the z-direction or we can put the same spheres next to each other in the x-direction. If we want a proper sampling through phase space, we need one extra move, namely the rotational move.

4.1 Rotational moves

Simulating hard particles which are not spherically symmetric like dumbbells, requires full rotational moves in order to have proper sampling. To accomplish this we need to implement to extra vectors $\mathbf{c_m}$ and $\mathbf{o}$, which are the center of mass and orientation vector respectively (Figure 4.2). Let $\mathbf{p}_1$ and $\mathbf{p}_2$ be the midpoints of the two spheres of a dumbbell then we define $\mathbf{c_m} = (p_1 + p_2)/2$ and $\mathbf{o} = (p_1 - p_2)/2$ by which we can write $\mathbf{p}_1 = \mathbf{c_m} + \frac{1}{2}\mathbf{o}$ and $\mathbf{p}_2 = \mathbf{c_m} - \frac{1}{2}\mathbf{o}$. To rotate a dumbbell, we simply rotate the orientation vector $\mathbf{o}$ of the dumbbell and we simply calculate the new location of the two spheres by the previous equations. To actually rotate a vector we will make use of a quaternion Q, which is a four dimensional vector. We first select a random vector $\mathbf{r_{ax}} = \{x, y, z\}$ of length one, which will act as the axes of rotation. We will rotate the orientation vector

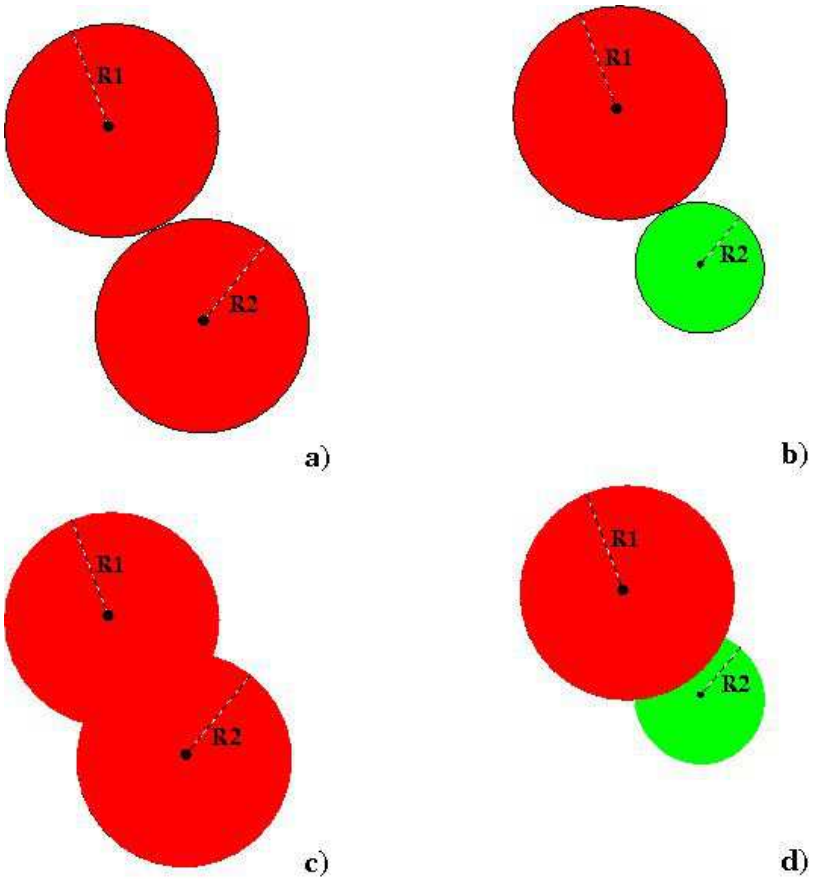

Figure 4.1: a) A dumbbell with spheres of equal diameter connected with each other. b) A dumbbell with spheres of different diameter connected with each other. c) A dumbbell with spheres of equal diameter with a certain overlap. d) A dumbbell with spheres of different diameter and with a certain overlap.

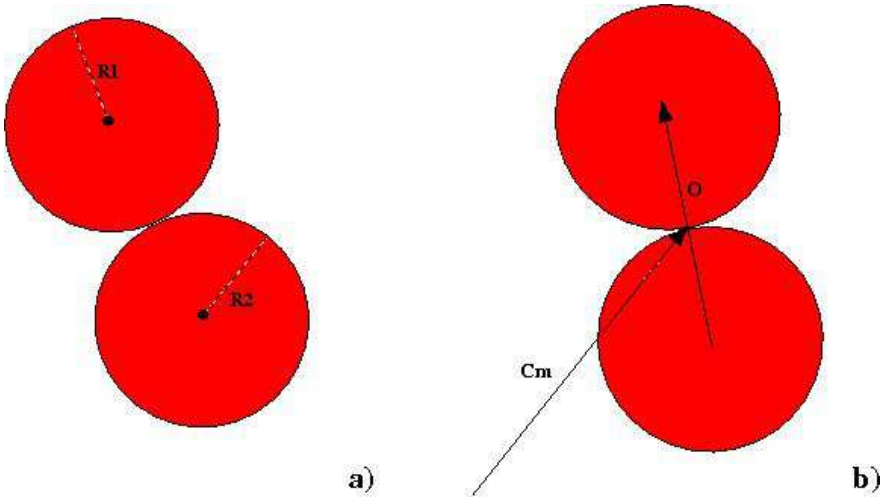

Figure 4.2: a) A dumbbell consisting of spheres of equal radii, which do not overlap. b) The position of the dumbbell is given by the center of mass $\mathbf{c}_m$ and the orientation is given by the vector $\mathbf{o}$.

$\mathbf{o}$ around this axes with a small random angle $\Delta\theta \in [-\theta_{max}, \theta_{max}]$. With $\mathbf{r_{ax}}$ and $\Delta\theta$ we construct the four dimensional vector $Q = \{q_0, q_1, q_2, q_3\}$, where $q_0 = \cos(\Delta\theta/2)$, $q_1 = x\sin(\Delta\theta/2)$, $q_2 = y\sin(\Delta\theta/2)$ and $q_3 = z\sin(\Delta\theta/2)$. The vector Q has the property that $\|Q\|_2 = 1$. The rotational matrix R_Q is now given by

$$R_Q = \begin{pmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_1q_2 + q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{pmatrix} \quad (4.1)$$

A rotation of the orientation vector $\mathbf{o}$ is simply $\mathbf{o}' = R_Q\mathbf{o}$. The new locations $\mathbf{p}_1'$ and $\mathbf{p}_2'$ of the spheres after the rotation can be calculated by $\mathbf{p}_1' = \mathbf{c_m} + \frac{1}{2}\mathbf{o}'$ and $\mathbf{p}_2' = \mathbf{c_m} - \frac{1}{2}\mathbf{o}'$.

4.2 Simulating dumbbells

The FBMC method applied to a systems of hard dumbbells works similar as with hard spheres. We create the SIL-List and OIL-List as explained in the previous chapter, except that the radius r_{max} is now based on two spheres. In case of hard spheres we have that $r_{max} = \max_{0\leq i<n} r_i$, where r_i is the radii of sphere B_i in the box. With dumbbells with no overlapping spheres we have that $r_{max} = \max_{0\leq i<n}(r_i^s + r_i^l)$, where r_i^s and r_i^l are the radii of both spheres of the dumbbell. A rotational move can be treated the same way as a translation move. Like before we check for a possible overlap of the dumbbell with its own periodic image and if necessary we check for an overlap with one of the other particles. The overlap check itself is simply a combination of four sphere overlap checks.

4.3 Results

We first take a brief look at some crystal structures and continue with the construction of the EOS for Dumbbells. We will concentrate us on dumbbells which do not have partial overlapping spheres (Fig. 4.1a). The FBMC method is applied to a system with one dumbbell ($\sigma = 1$) in the simulation box. After $4 \cdot 10^5$ FBMC cycles with $\beta P = 40$, we get a crystal structure that look like a BCC structure (Fig 4.3). Furthermore we did run some simulations with two equal dumbbells (Fig 4.4) and with different dumbbells (Fig 4.5). In the first case we got a BCC structure and in the latter case we end with a crystal structure, which is similar as a γCuTi crystal. For more details about dumbbell crystal structures we refer to Ref. [7].

4.3.1 Equation of states for dumbbells

In the previous section we constructed the equation of state for hard spheres. The FCC crystal structure we found with the FMBC method formed the basis of the melting process, needed to find points in the crystal phase. We also

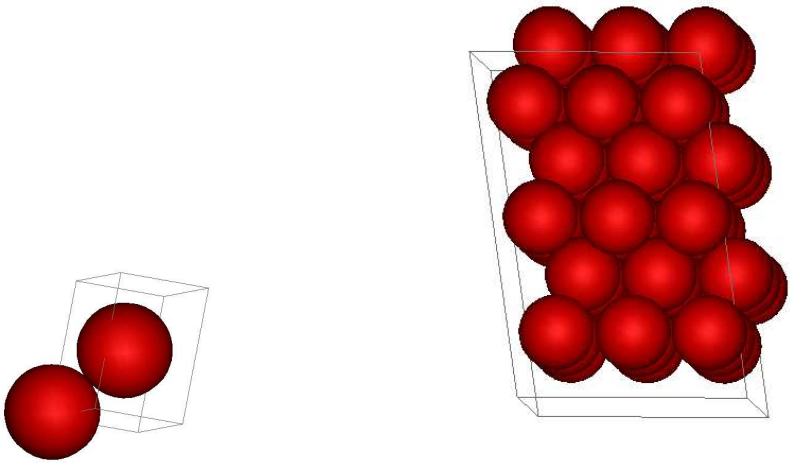

Figure 4.3: On the left a plot of the simulation box with one dumbbell with spheres of equal diameter $\sigma = 1$ at $\beta P = 20$ after $4 \cdot 10^5$ FBMC cycles (10^5 EC). On the right side is the same simulation box, with the 26 images in each direction. An BCC crystal structure is visible.

started in a more sparse system to find points in liquid phase. We used the same recipe for a system with hard dumbbells. Like a system of hard spheres, we studied a system with one type of dumbbell in a cubic box. We construct the EOS for hard dumbbells with the spheres connected to each other (Fig. 4.1a) and of equal diameter ($\sigma_1 = \sigma_2 = \sigma$). The volume of this kind of dumbbell is $V_{dumbbell} = (4/3)\pi(\sigma_1/2)^3 + (4/3)\pi(\sigma_2/2)^3 = (1/6)\pi(\sigma_1^3 + \sigma_2^3)$, which we can write as $(1/6)\pi(2\sigma^3) = (1/3)\pi(\sigma^3)$. The packing fraction is defined as $\eta = NV_{dumbbell}/V$.

For hard dumbbells the equation of state [5] is given by

$$\frac{P_{ST}}{\rho kT} = \frac{1 + F(l^*)\eta + G(l^*)\eta^2 - H(l^*)\eta^3}{(1-\eta)^3}, \tag{4.2}$$

where F, G and H are polynomial functions of $l^* = \sigma/d$ (Eq. 4.3). The value d is the distance between the midpoints of the two spheres of the dumbbell. In case of a dumbbell with non-overlapping spheres with $\sigma_1 = \sigma_2 = \sigma$), we find that $d = \sigma$ and thus $l^* = 1$. The polynomial values U, V, W, X, Y and Z are different for each value of l^*. The polynomial values for $l^* = 1$ are stated in Table. 4.1.

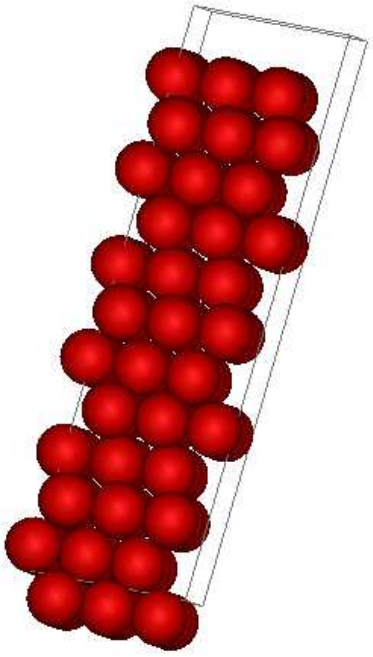

Figure 4.4: A crystal structure based on a system with two equal dumbbells with non overlapping spheres and equal radii $r_1 = r_2 = 1$. The structure that became visible looks like a BCC structure

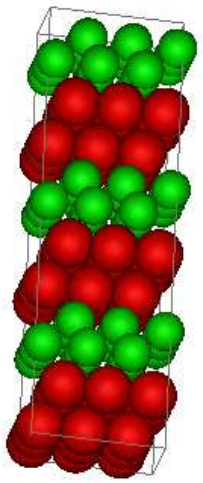

Figure 4.5: A crystal structure based on a system with two different dumbbells ($l^* = 0.8$) with non overlapping spheres. The crystal structure that became visible at the end of the simulation looks like γCuTi

U	0.37836	V	1.07860
W	1.30376	X	1.80010
Y	2.39803	Z	0.35700

Table 4.1: The polynomial values U, V, W, X, Y and Z for $l^* = 1$.

$$\begin{aligned} F(l^*) &= 1 + Ul^* + Vl^* \\ G(l^*) &= 1 + Wl^* + Xl^* \\ H(l^*) &= 1 + Yl^* + Zl^* \end{aligned} \tag{4.3}$$

Let us define $P_{\text{cs}}(\eta)$ as

$$P_{\text{cs}}(\eta) = \frac{1 + F(l^*)\eta + G(l^*)\eta^2 - H(l^*)\eta^3}{(1-\eta)^3}, \tag{4.4}$$

To construct points for the equation of states we used a system with 64 dumbbells. We simulated dumbbell structures using $4 \cdot 10^5$ MC cycles, where approximate 10^5 cycles where needed to reach equilibrium. In Fig 4.6 we plotted the $P_{\text{cs}}(\eta)$ as a continues line together with the values ontained from the simulations. Around $\beta P \sigma^3 = 12$ there is clearly a change from the liquid phase to a crystal phase.

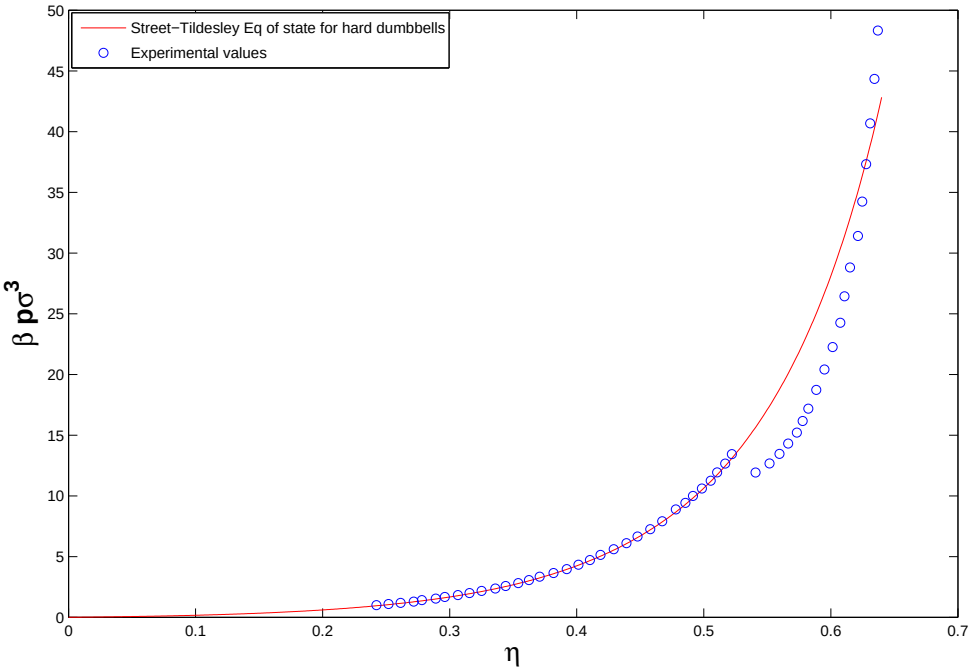

Figure 4.6: Street-Tildesley equation of state of a system with 64 dumbbells. First run consists of 400MC (300 production) continued with runs of 400MC (300 production).

Chapter 5

Superballs

So far we applied the FBMC method to systems with spheres and dumbbells. We used the resulting crystal structures to construct an EOS. In some experimental studies particles are synthesized, which have a more cubic form than spheres (Fig. 5.1). To simulate this kind of particles, we take a look at a special subset of superellipsoids called superballs. With spheres or dumbbells the overlap check is a relative simple and fast operation. In the next section we give a definition of the superballs and we explain how we check for a possible overlap between two of these particles.

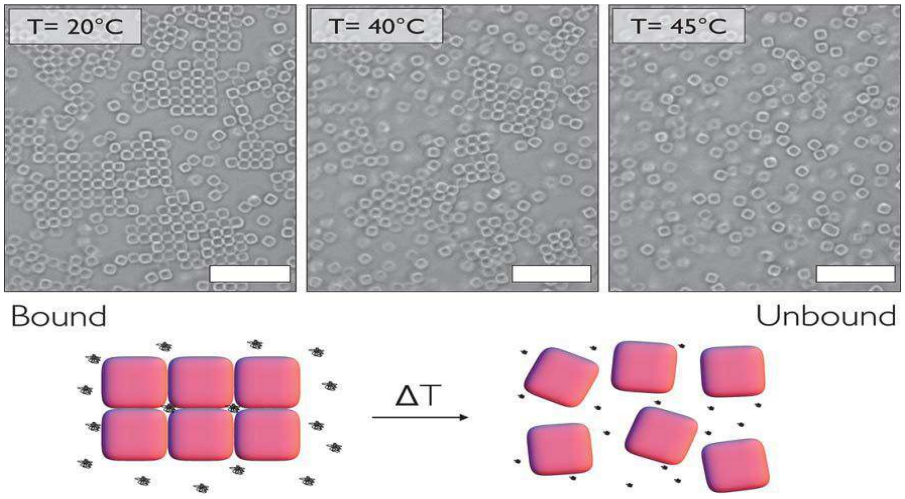

Figure 5.1: An example of an experiment with a crystal built out of cubic shaped particles, which is melting (Ref. [10]).

5.1 Formulation

The superellipsoids are a generalization of ellipsoids, which are defined by the implicit equation

$$\left(\left|\frac{x}{a}\right|^{\epsilon_1} + \left|\frac{x}{b}\right|^{\epsilon_1}\right)^{\epsilon_2/\epsilon_1} + \left|\frac{x}{c}\right|^{\epsilon_2} \leq 1, \tag{5.1}$$

where $a, b, c \in \mathbb{R}\backslash 0$ and $\epsilon_1, \epsilon_2 \in \mathbb{R}$ with $\epsilon_1, \epsilon_2 > 0$. In our simulation we focus ourselves on the so called superballs. These superballs are superellipsoids with the extra conditions $a = b = c$ and $\epsilon_1 = \epsilon_2 = \epsilon$. For further simplicity we assume that $a = b = c = 1$. With the additional conditions we get the following equation for superballs:

$$|x|^\epsilon + |y|^\epsilon + |z|^\epsilon \leq 1, \tag{5.2}$$

In Fig 5.2 some example of superballs are plotted for different parameters ϵ. When ϵ goes to infinity then the shape of the superballs get more cubic. On the other hand if ϵ goes to zero, the shape of the superball get more star like.

5.1.1 Overlap potential

To determine if two superballs overlap, we need an overlap potential, which take the value of ϵ and the orientation of the particles into account. Note that if $\epsilon = 2$, then Eq 5.2 becomes the equation for hard spheres and the overlap check can be done as explained in chapter 2. For $\epsilon \neq 2$, Donev [2] provided an algorithm to calculate the overlap potential between two superballs. To get to a working algorithm,which we can use for our simulations, we have to rewrite Eq. 5.2 first. Let us define the shape function $\zeta(\mathbf{r}) = g(\tilde{\zeta}(\tilde{\mathbf{r}})) - 1$ with

$$g(x) = x^{(1/\epsilon)}, \tag{5.3}$$

and

$$\tilde{\zeta}(\tilde{\mathbf{r}}) = \sum_i f(\tilde{\mathbf{r}}_i), \tag{5.4}$$

where $f(x) = |x|^{2\epsilon}$ and $\tilde{\mathbf{r}}$ is the relative position rotated and scaled according to the shape and orientation of the superball. The shape of a superball is spherically-symmetric and therefore the relative position depends only on the orientation of the superball. The orientation of a superball is given by a quaternion $\mathbf{q}$, from which we can derive the rotation matrix Q exactly the same way as what we did with the dumbbells. Let $\mathbf{r_0}$ the midpoint of a superball, than $\tilde{\mathbf{r}} = Q^{-1}(\mathbf{r} - \mathbf{r_0})$ with $\mathbf{r} = (x, y, z)$.

Let ζ_a and ζ_b be two superballs as in Eq 5.5, then we can construct an overlap potential $\zeta(\zeta_a, \zeta_b)$ with the following properties:

$$\begin{aligned} \zeta(\zeta_a, \zeta_b) > 0 &\text{ if } \zeta_a \text{ and } \zeta_b \text{ do not overlap} \\ \zeta(\zeta_a, \zeta_b) = 0 &\text{ if } \zeta_a \text{ and } \zeta_b \text{ are externally tangent} \\ \zeta(\zeta_a, \zeta_b) < 0 &\text{ if } \zeta_a \text{ and } \zeta_b \text{ do overlap} \end{aligned} \tag{5.5}$$

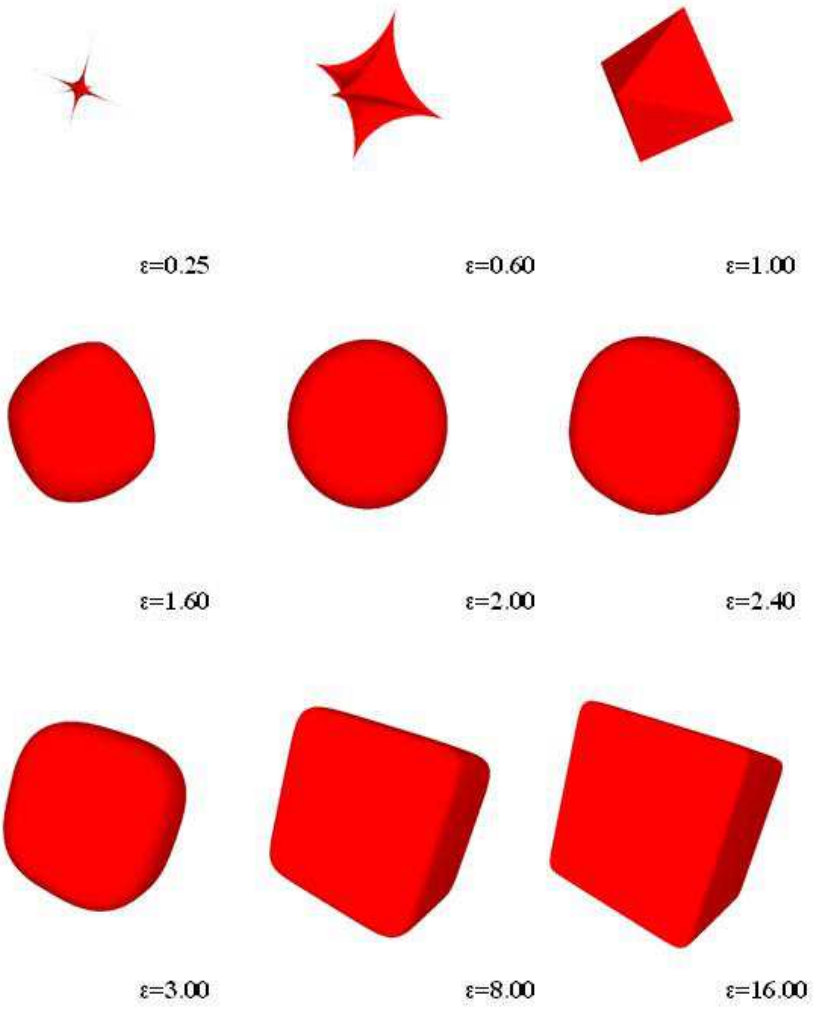

Figure 5.2: Different superballs of the type $|x|^\epsilon + |y|^\epsilon + |z|^\epsilon \leq 1$ for different values of ϵ. When ϵ goes to infinity the superballs get more cubic and the superballs get more star shaped when ϵ goes to zero.

The overlap potential is based on a procedure originally developed by the Perram and Wertheim(PW) (Ref. [2]) and is defined through

$$\zeta = \mu^2 - 1 = \max_{0\leq\lambda\leq 1} \min_{\mathbf{r}_c} [\lambda\zeta_A(\mathbf{r}_C) + (1-\lambda)\zeta_B(\mathbf{r}_c)] \tag{5.6}$$

The idea of the PW potential is to find a common scaling factor μ such that the scaled superballs $\zeta_a^\mu = \mu g[\tilde{\zeta}_a(\tilde{\mathbf{r}})] - 1$ and $\zeta_b^\mu = \mu g[\tilde{\zeta}_b(\tilde{\mathbf{r}})] - 1$ are externally tangent in the point $\mathbf{r}_C$. The uniqueness of the solution of the inner optimalisation over $\mathbf{r}_C$ for every value of λ depends on the value ϵ. For $\epsilon > 1$ the superballs are convex and the value $\mathbf{r}_C$ is unique and satisfies the condition

$$\Delta\mathbf{n} = \lambda\mathbf{n}_A(\mathbf{r}_C) + (1-\lambda)\mathbf{n}_B(\mathbf{r}_C) = 0, \tag{5.7}$$

where $n(\mathbf{r}) = \nabla\zeta(\mathbf{r})$ is the normal vector in point $\mathbf{r}$. The condition $\zeta = \zeta_a^m u(\mathbf{r}_C) = \zeta_b^m u(\mathbf{r}_C)$ optimizes the outer optimalisation over λ, which means that when the particles are scaled by the common factor $\mu = \sqrt{1+\zeta}$, the normal vectors in the point $\mathbf{r}_C$ are perpendicular (Fig 5.3)

Calculating the overlap potential for two ellipsoids can be one by solving λ in Eq. 5.8 and looking for the value $\mathbf{r}_C(\lambda)$ that solves $\zeta_a^m u(\mathbf{r}_C) = \zeta_b^m u(\mathbf{r}_C)$. One way to solve the system is to track continuously the solution from Eq. 5.8 as λ goes from 0 to 1, by solving the the following ODE.

$$M\frac{d\mathbf{r}_c}{d\lambda} = [\lambda\nabla^2\xi_A + (1-\lambda)\nabla^2\xi_B]\frac{d\mathbf{r}_c}{d\lambda} = \Delta g = \nabla\xi_B - \nabla\xi_A \tag{5.8}$$

The ODE with initial conditions $\mathbf{r}_C = \mathbf{r}_B$ and $\lambda = 0$ can be obtained by differentiating Eq. 5.8 with respect to λ. Instead of trying to solve the ODE, we take an initial guess for λ and use the Newton's method on Eq. 5.8 to solve the problem numerically. The Newton step is determined by

$$\begin{aligned} \Delta\lambda &= \frac{1}{\zeta_{\lambda\lambda}}[(\zeta_a - \zeta_b) - \Delta\mathbf{g}M^{-1}\Delta\mathbf{n}] \\ \Delta\mathbf{r}_C &= M^{-1}(\Delta\mathbf{g}\Delta\lambda - \Delta\mathbf{n}), \end{aligned} \tag{5.9}$$

where $\zeta_{\lambda\lambda} = \Delta\mathbf{g}^T M^{-1}\Delta\mathbf{g}$. We update the values of λ_n and $\mathbf{r}_C^n$ by $\Delta\lambda$ and $\Delta\mathbf{r}_C$ respectively as

$$\begin{aligned} \lambda_n &= \lambda_0 - \Delta\lambda \\ \mathbf{r}_C^n &= \mathbf{r}_C^0 + \Delta\mathbf{r}_C. \end{aligned} \tag{5.10}$$

Unfortunately there is no guarantee that the Newton method will converge. For superballs the value of ζ_a, ζ_b, $\nabla\zeta_a$ and $\nabla\zeta_b$ must be calculated as follows:

$$\begin{aligned} \nabla\zeta &= g'(\tilde{\zeta})\nabla\tilde{\zeta} \\ \nabla^2\zeta &= g'(\tilde{\zeta})(\nabla^2\tilde{\zeta}) + g''(\tilde{\zeta})(\nabla\tilde{\zeta})(\nabla\tilde{\zeta})^T \\ \nabla\tilde{\zeta} &= Q^T\mathbf{u} \\ \nabla^2\tilde{\zeta} &= Q^T ZQ, \end{aligned} \tag{5.11}$$

where Z is a diagonal matrix with $Z_{ii} = f''(\tilde{\mathbf{r}}_i)$ and the vector $\mathbf{u}$ with $\mathbf{u}_i = f'(\tilde{\mathbf{r}}_i)$.

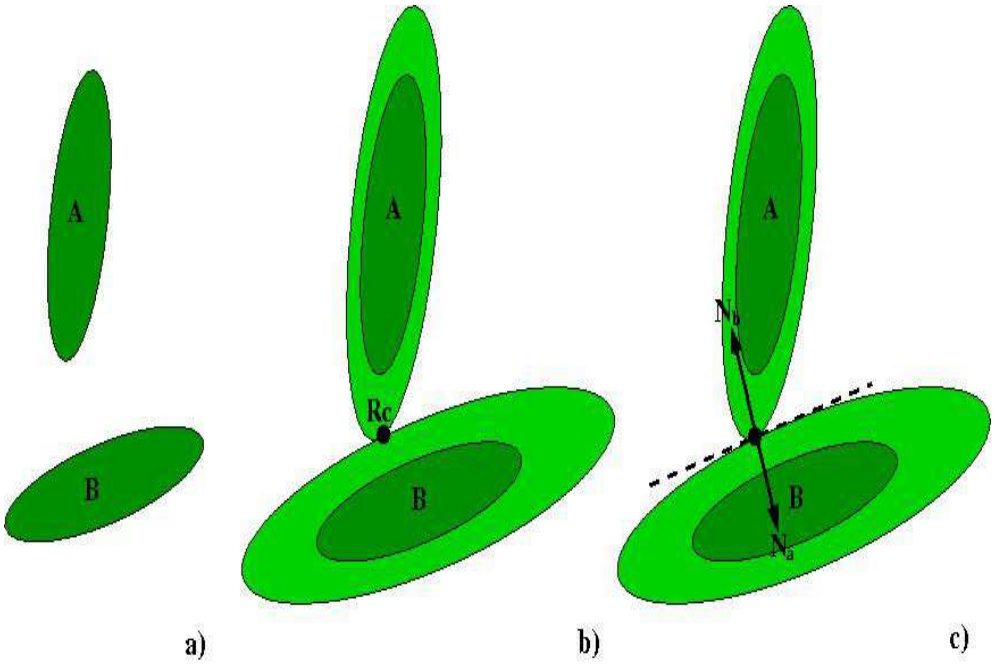

Figure 5.3: To check if two superballs do have an overlap, we have to find a common scaling factor μ. In this figure we sketch the idea by looking a two ellipsoids. The figure is inspired by Ref [2]. a) Two 2D ellipsoids A and B with clearly no overlap. b) The ellipsoids A and B are scaled by a common factor μ (light green) and touch each other in point R_c. c) In the point R_c the normal vectors of both ellipsoids are parallel and in opposite direction.

5.2 Results

We define $\epsilon = 2p$, $p > 0$ for further convenience, not to confuse with the P for pressure. With this definition we have simple hard spheres in case of $p = 1$, more cubic and convex shaped particles when $p > 1$ and non-convex particles when $p < 1$. First we apply the FBMC method to a system with one superball in a rectangular box and set the pressure at $\beta P = 80$. We let the system run over a period of 10^6 FBMC cycles for different values of p. With $p > 0.5$ we get different crystal structures with no overlaps (Fig. 5.4). This structures can be analysed as is Ref. [3] We also did a simulation with superballs with $p <= 0.5$ which resulted in crystal structures with a partial overlap, Unfortunately we can not analyse this structures, because they are not valid according to the overlap potential. The overlaps occur due to the fact the overlap routine is not stable for all values of p. It seems that there is no theoretical upper limit for the value of p in case of one superball in the box. On the other hand FBMC, NVT or NPT simulations with more than one particle int the box we get no overlaps when $0.8 \lesssim p \lesssim 1.8$. We can conclude that for certain values of p it is possible to predict some crystal structures with the PW-based overlap routine from the previous section. Unfortunately the algorithm is not stable for all values of p and the time needed to check for a possible overlap between two superballs is more time consuming then a simple overlap check for spheres.

5.2.1 EOS In the liquid phase

To construct an equation of state for hard spheres or hard dumbbells, we started deep in the gas/liquid phase and we increased the pressure to get dense systems. Eventually the system is so dense that compression is (almost) not possible anymore, although equilibrium is not reached yet. To overcome this problem, we started with a crystal structure found with the FBMC method and put this in a rectangular box. Unfortunately it is very difficult or some times impossible to create a crystal structure that fits in a box for most values of p. To create points for the EOS in the crystal phase, additions must be made to the FBMC method. Because this can not be done with a couple of ours, we diced to concentrate us only one the gas/liquid phase for now.

To create an EOS for superballs in the liquid phase, such that we get an acceptable approximation, we need at least 125 superballs. To be assured that the simulations are finished in time, we run NPT simulations with a cell-list construction. The idea is that when we have a large system of particles, a particle may have an overlap with some particles in the neighborhood. In other words a particle can only have an overlap with other particles within a radius of $2r$ or less, but the number of particles within this radius will be minor compared to the total number of particles in the simulation box. Therefore it is worthwhile to keep track of the particles which are in the neighborhood for each particle in a so called cell-list. For more details about the Cell-list and its implementation can be found in Ref. [4].

We constructed the EOS in the liquid/gas phase for superballs, by doing simulations for $p = 1.25$, $p = 2.0$, $p = 3.0$, $p = 4.0$ and $p = 5.0$. The simulations where done for pressures between $\beta P = 0.1$ and $\beta P = 1.1$ with step size $\Delta\beta P = 0.04$. We let the simulations run until our program had performed $2.5 \cdot 10^6$ MC Cycles ($5 \cdot 10^6$ EC). We Compare our results with the Carnahan and Starling EOS for spheres to see if there is a relation between spheres and superballs (Fig 5.5). Clearly the packing fraction η is larger for all $p > 1$ at every pressure P. Moreover when p increases the packing fraction increases as well when the pressure stays constant. We compared our data, which we got from our simulations for $p = 1.5$, with the data from Ran's simulations (Fig. 5.7). We compare the date by calculating the relative error

$$\mathrm{Rel}(\eta_i) = \frac{|\eta_i^R - \eta_i|}{\eta_i^R}, \tag{5.12}$$

where η_i and η_i^R are values from our and Ran's simulations respectively. The relative error $\mathrm{Rel}(\eta_i) \approx 10^{-3}$ for all the points we compared, which gives us confidence in the data from our simulations and the correctness of the EOS for all the values of p.

5.2.2 Fit

With the simulation data we got for superballs we tried to make a fit for the EOS for superballs. The EOS for superballs look similar as the one of hard spheres, therefore we try to make a fit for superballs with the model function

$$\mathrm{cs}_{\mathrm{model}}(\eta) = A\eta\frac{1 + B\eta + C\eta^2 - D\eta^3}{(1-\eta)^3}, \tag{5.13}$$

where A, B, C and D are constants determined by the fitting algorithm from MATLAB. To see what kind of error we can expect for the constants A, B, C and D, we started with a set of exact points $\{\eta_i, \beta P\sigma_i^3\}$ on the Starling and Carnahan EOS for spheres. We generated up to 250 points and added a small random error ϵ to every η_i. From simulations we know that the error in η is about 10^{-3}, therefore we choose $\epsilon = 10^{-3}$. If we make a fit through this perturbated points and compared the determined constants A, B, C and D with the theoretical constants, we get errors from order 10^{-1} up to order 10^0. To get a fit trough points with an error in η of order 10^{-3} and a same order 10^{-3} in the constants A, B, C and D, we need a least 2500 data points or we must simulate data points with smaller errors. Unfortunately we where no be able to generate such a amount of points or to generate points with such small errors.

5.2.3 The Radial Distribution Function

Suppose we have a system with a number of particles at a pressure βp, which is in equilibrium. Without loss of generality we can translate the system such that

one particle is in the origin. In stead of determining the density of the system as a whole, we can also determine the variation in the local density. This variation in local density is a function of r, which is the radial distance from the origin. This function is called the Radial Distribution Function (RDF) $G(r)$. In real experiments scientists can construct the $G(r)$ out of a experimental system by using scattering methods. This experimental $G(r)$ can be compared with the values from the simulations, which can help them to understand the physics behind the system.

To determine the $G(r)$ during a simulation, we have to keep track of all distances between all particles in the box. The $G(r)$ is defined as (Ref. [5][8])

$$G(r) = \frac{V}{N}\langle \sum_{i=1}^{N} \sum_{j=i+1}^{N} \delta(\mathbf{r} - \mathbf{r}_{ij}) \rangle \tag{5.14}$$

,where δ is the Dirac delta function and $\mathbf{r}_{ij} = \mathbf{r}_i + \mathbf{r}_j$ the displacement vector between particle i and j. While running a simulation we measure the histogram of $G(r)$ numerically. To do this properly, we run a NPT simulation with n particles in a rectangular box with size L. We create a system of nbins empty bins $(bin(m)|m \in \{1, \ldots, nbins\})$. The size of each bin is defined as $dr =$ L/2nbins.
In each MC Cycle we loop over all different particles pairs $\{(i,j)|i \neq j\}$ in the box. By calculating the distance $d(p_i, p_j)$ the corresponding bin can be determined by $m = \lfloor d(p_i, p_j)/dr \rfloor$. The corresponding bin is updated by two $(bin(m) = bin(m) + 2)$, because the particle pair (i,j) contributes to the same bin as the particle pair (j,i). At the and of the simulation we normalize all the bins $g(r + 0.5\text{dr})$ by dividing the values with the number of particles n and the average number of particles in the same interval in an ideal gas. The radial distribution function is then given by

$$g(r + 0.5\text{dr}) = \frac{\text{bin}(m)}{n \cdot \text{bin}_{id}(m)}, \tag{5.15}$$

where $\text{bin}_{id} = (4\pi\rho)[(r + \text{rm})^3 - r^3]/3$
During our simulations we kept track of the displacement vector between particles and constructed the radial distribution function $g(r)$. We did NPT simulations in the liquid/gas phase with a fixed $\eta = 0.4$ and $\eta = 0.3$ for superballs with $1 \leq p \leq 1.5$. We compared the results with the $g(r)$ function for hard spheres to see how the superballs are related.
In Fig. 5.8. we plotted the results for $\eta = 0.4$ and we made a subplot of the $g(r)$ in Fig. 5.9. The first peek of the $g(r)$ function is clearly getting smaller when p increases and shifts to the right. It looks like that the other peeks getting smaller as well, but only minor compared to the first peek. Furthermore the small peeks are shifted the same way as the large peeks. With $\eta = 0.3$ we get similar results (Fig. 5.10, 5.11), only thee peeks are smaller. To check the correctness of our simulation data, we compared the $g(r)$ for superballs with

$p = 1$ (hard spheres), with the theoretical $g(r)$ based on Ornstein-Zernike theory (Ref.[8]). Figure 5.12 shows us that the graphs are similar, which convince us in the correctness of out simulation data.

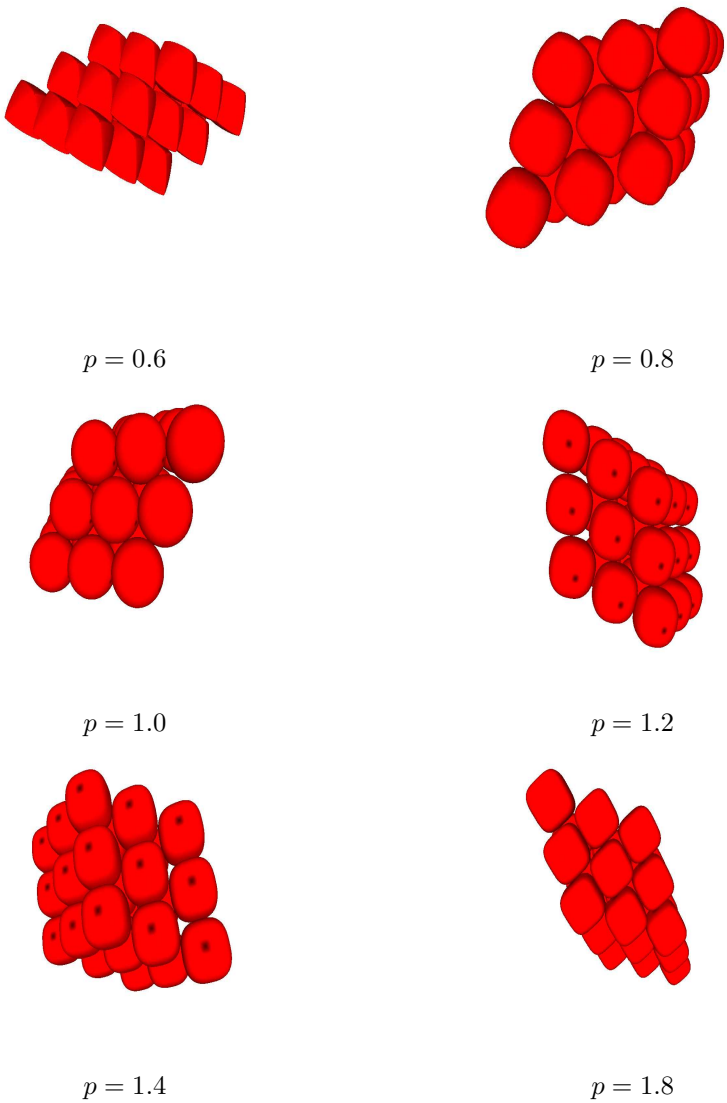

Figure 5.4: FMBC method applied to different systems with one particle in the box. Each system contains a different type of superball with a pressure of $\beta P = 40$. After 10^6 cycles we get different crystal structures for each type of superball. In case of $p = 1$(Hard Spheres) we get simply a FCC structure.

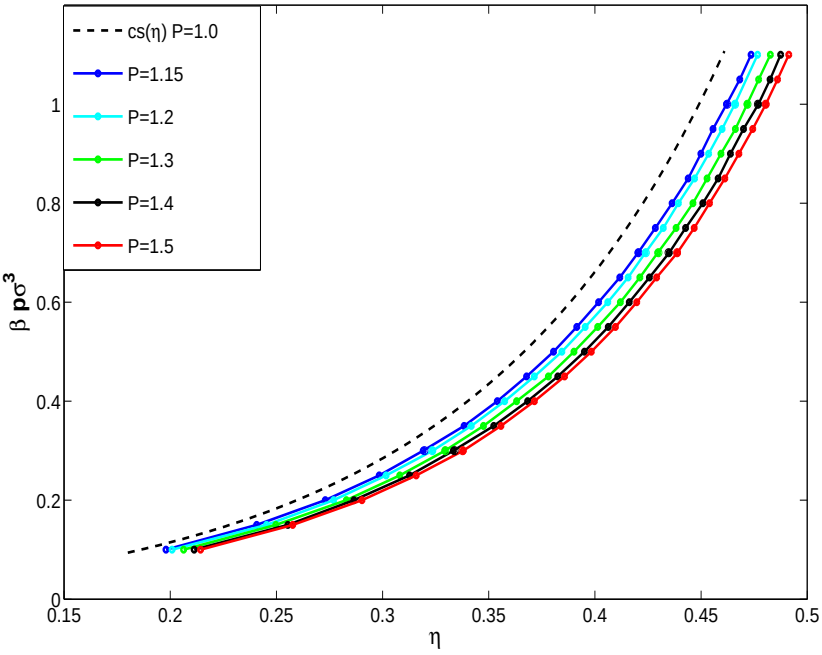

Figure 5.5: The equation of state for superballs $|x|^{2p} + |y|^{2p} + |z|^{2p} \leq 1$ determined for different values of p. The points for the EOS are result from simulations with pressures in the liquid phase ($0.1 \leq \beta P \leq 1.1$). Compared to the Carnahan and Starling EOS for spheres (dotted line), the fraction packing η is larger at every pressure for all $p > 1$. Moreover when the superball parameter p increases the packing fraction η increases. It seems that the packing fraction increases in a linear way when p increases.

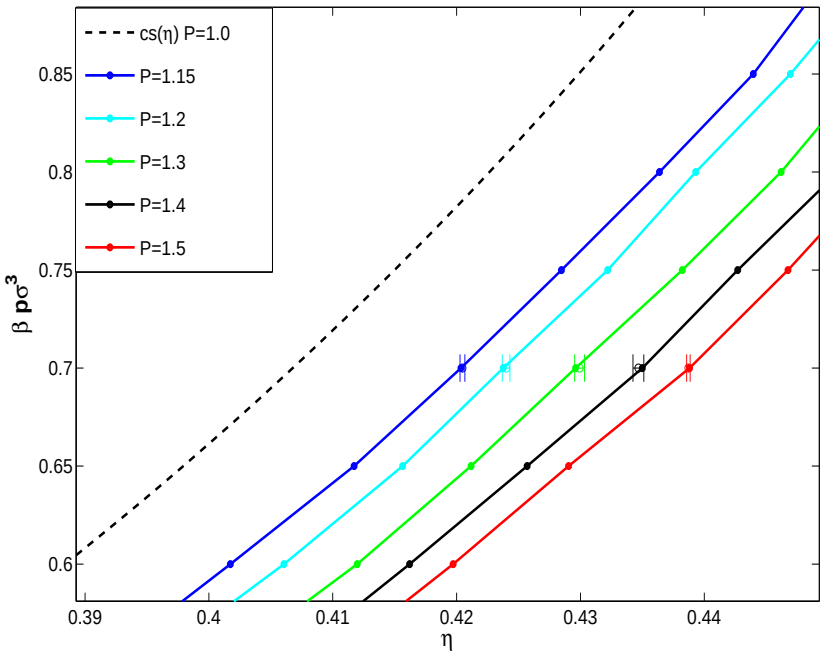

Figure 5.6: A subpart of figure 5.5 The equation of state for superballs $|x|^{2p} + |y|^{2p} + |z|^{2p} \leq 1$ determined for different values of p. The points for the EOS are result from simulations with pressures in the liquid phase ($0.1 \leq \beta P \leq 1.1$). Compared to the Carnahan and Starling EOS for spheres (dotted line), the packing fraction η is larger at every pressure for all $p > 1$. Moreover when the superball parameter p increases the packing fraction η increases. It seems that the packing fraction increases in a linear way when p increases.

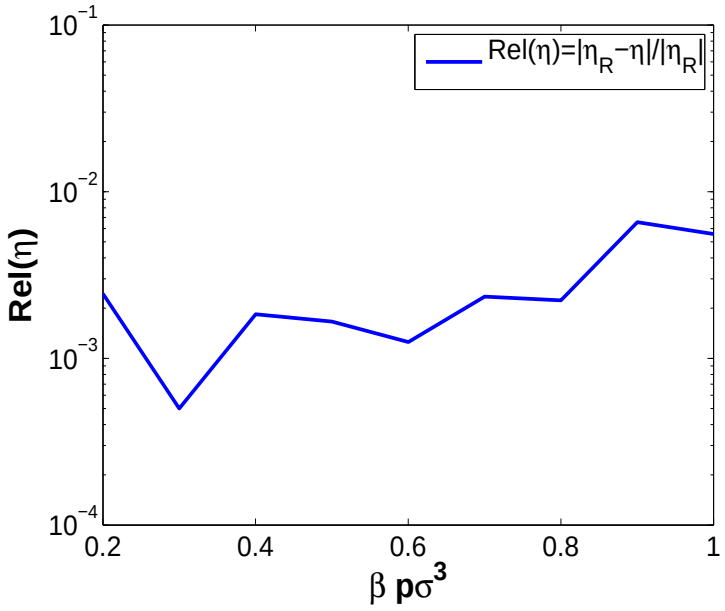

Figure 5.7: We compared the difference between the EOS determined by our simulations and the simulations from Ran. We compared our data for superballs $|x|^{2p} + |y|^{2p} + |z|^{2p} \leq 1$ with $p = 1.5$, by calculating the relative error in Rel(η). Rel($\eta = |\eta_R - \eta|/|\eta_R|$), where η is the packing fraction from our simulations and η_R is the packing fraction from Ran's simulation.

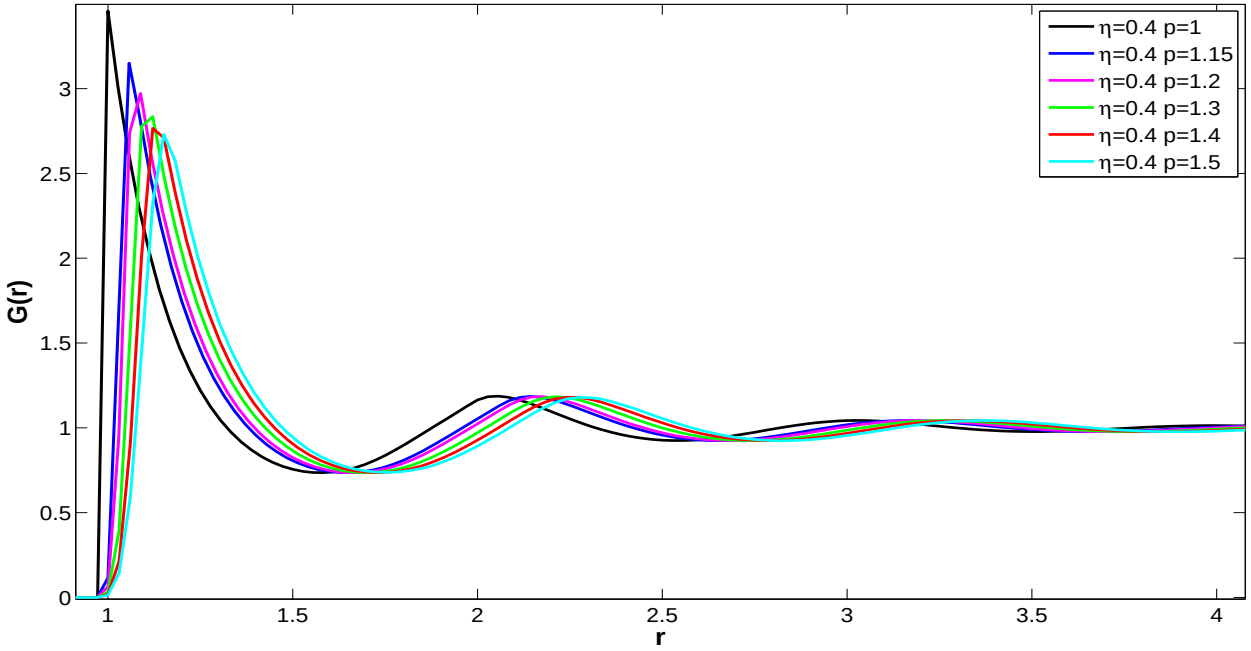

Figure 5.8: The radial distribution function $g(r)$ is constructed during NPT simulations with $\eta = 0.4$ for superballs with $1 \leq p \leq 1.5$. We compared the results with the $g(r)$ function for hard spheres (black line). The first peek of the $g(r)$ function is getting smaller when p is getting smaller and the peeks are getting closer together. The other peeks have the same in hight, but only with a slight offset. Eventually all $g(r)$ function converge to one.

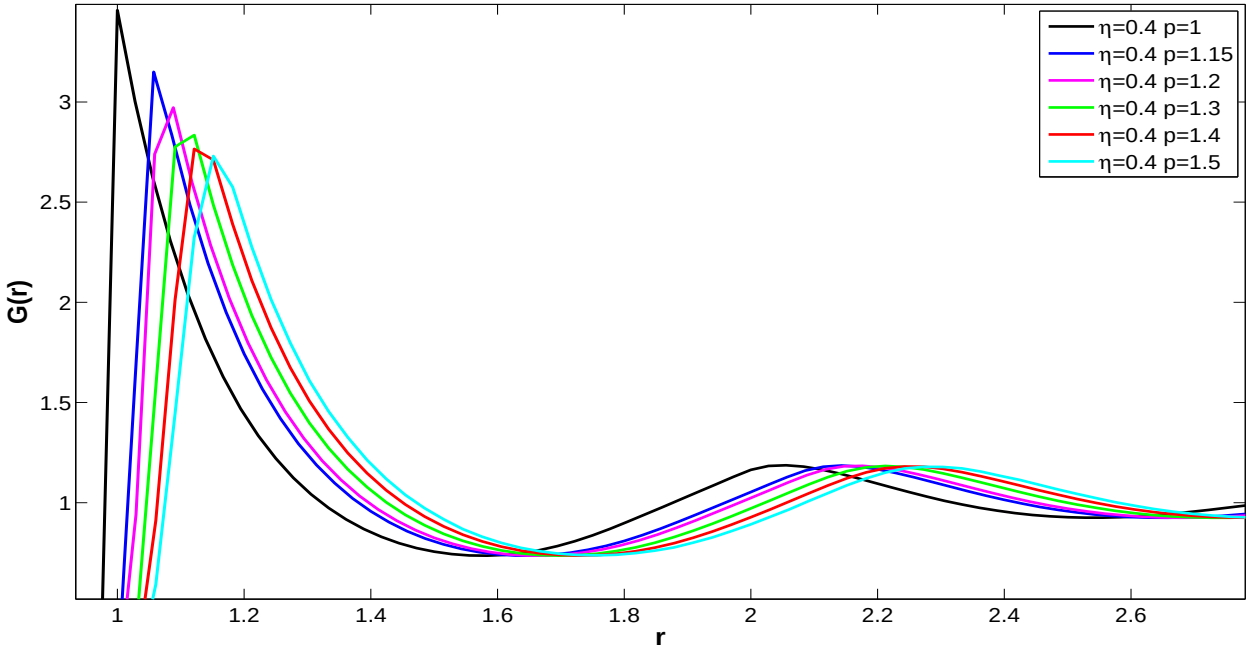

Figure 5.9: A part of the radial distribution function $g(r)$ is constructed during NPT simulations with $\eta = 0.4$ for superballs with $1 \leq p \leq 1.5$. We compared the results with the $g(r)$ function for hard spheres (black line). The first peek of the $g(r)$ function is getting smaller when p is getting smaller and the peeks are getting closer together. The other peeks have the same in hight, but only with a slight offset. Eventually all $g(r)$ function converge to one.

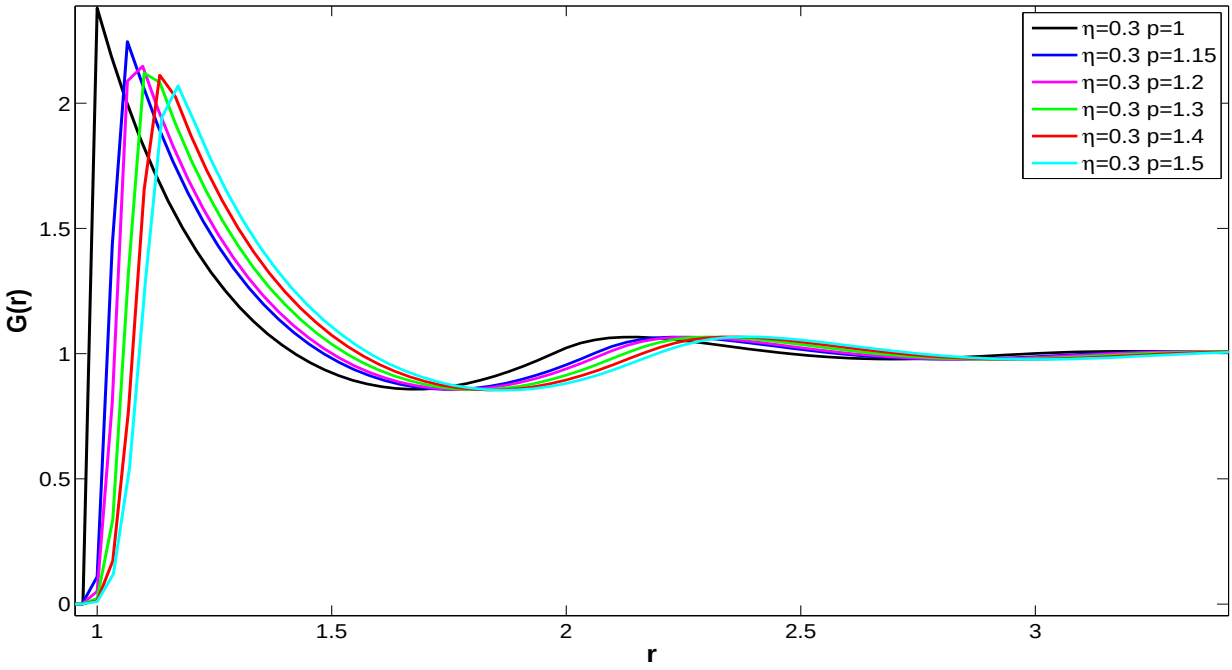

Figure 5.10: The radial distribution function $g(r)$ is constructed during NPT simulations with $\eta = 0.3$ for superballs with $1 \leq p \leq 1.5$. We compared the results with the $g(r)$ function for hard spheres (black line). The first peek of the $g(r)$ function is getting smaller when p is getting smaller and the peeks are getting closer together. The other peeks have the same in hight, but only with a slight offset. Eventually all $g(r)$ function converge to one.

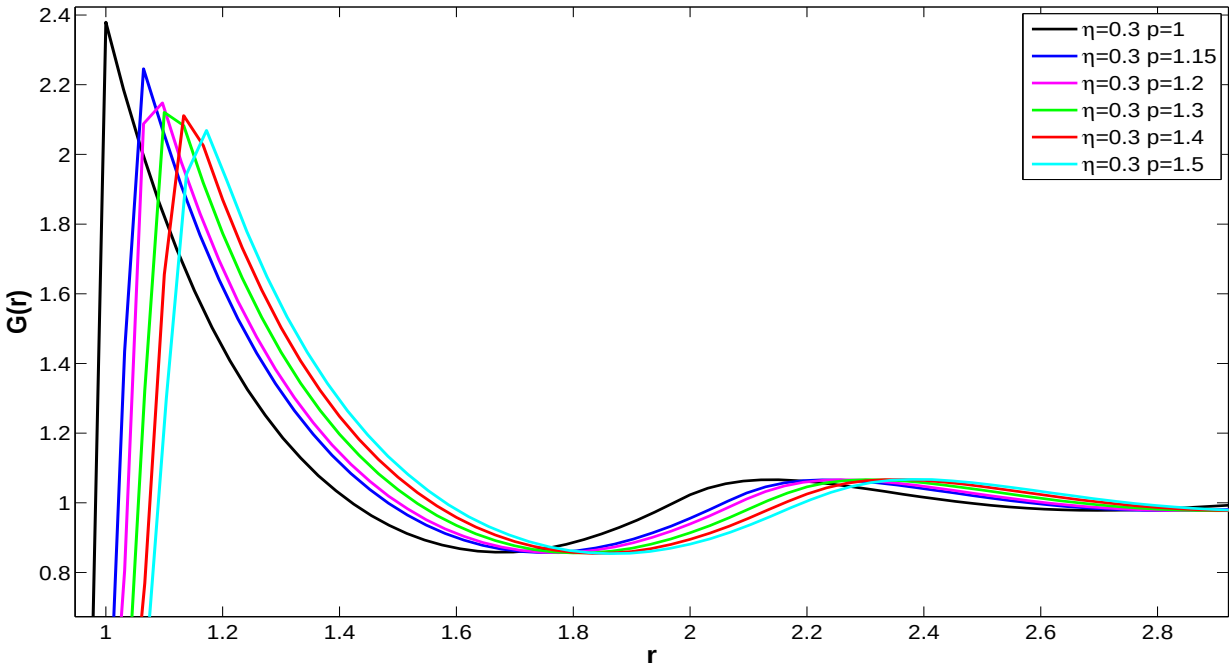

Figure 5.11: A part of the radial distribution function $g(r)$ is constructed during NPT simulations with $\eta = 0.3$ for superballs with $1 \leq p \leq 1.5$. We compared the results with the $g(r)$ function for hard spheres (black line). The first peek of the $g(r)$ function is getting smaller when p is getting smaller and the peeks are getting closer together. Eventually all $g(r)$ function converge to one.

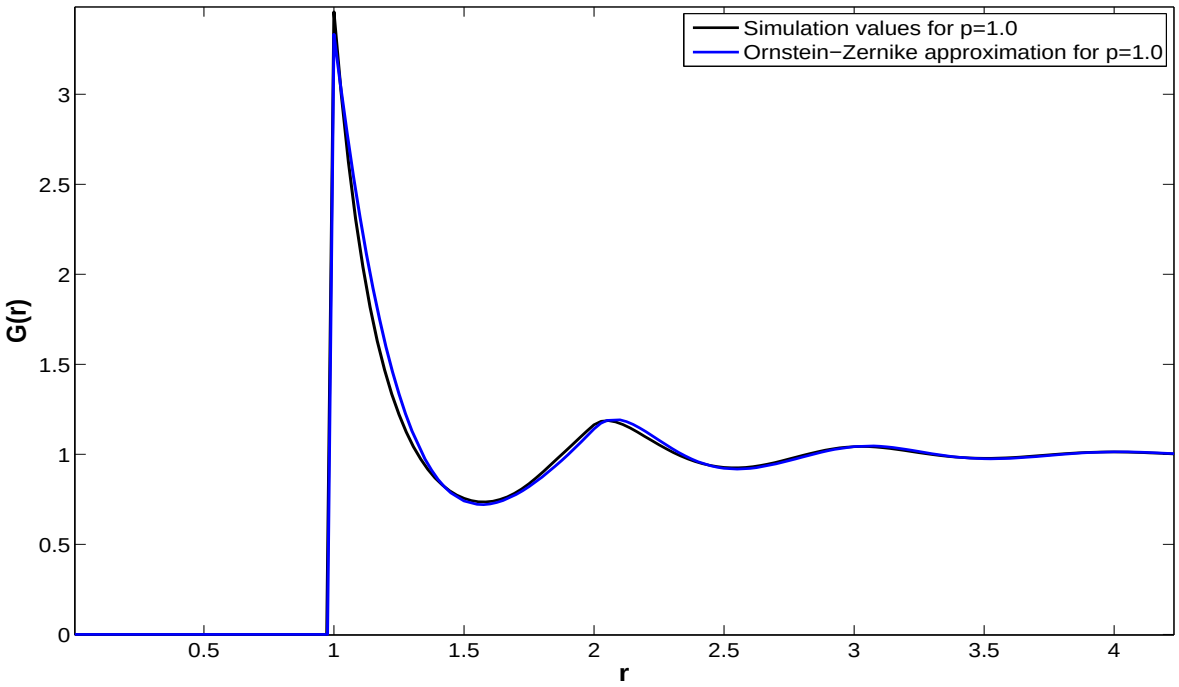

Figure 5.12: The radial distribution function $g(r)$ is constructed during NPT simulations with $\eta = 0.4$ for superballs with $p = 1.0$ (hard spheres). We compared the $g(r)$ function with the theoretical $g(r)$ based on Ornstein-Zernike theory (Ref.[8]). Both graphs are similar and convince us in the correctness of out simulation data.

Chapter 6

Conclusions

6.1 Conclusions

We determined the EOS for superballs for $1.0 \lesssim p <\lesssim 1.5$ and saw that if the pressure βp is fixed, the packing fraction η increases when ϵ increases. In other words, systems with cubic particles are denser than systems with spherical particles at the same pressure. Furthermore We verified that the simulation we wrote, functions well within the numerical tolerances required for physical studies. We compared our EOS for superballs for $\epsilon = 3.0$ with the data of Ran's simulations, we concluded that our data had a difference of order 10^{-3}, which gave us confidence in the correctness of our simulations. With the data of our simulations for the EOS, we tried to make a fit, which suffers from large statistical errors. To get a fit, which satisfies the required numerical precision, we need more data points or data points with a larger precision. For superballs we where also limited by stability of the overlap routine, which prevented us from simulating cubic crystals structure that can parameterised by a large value of ϵ. The overlap routine for superballs or general ellipsoids is only stable for $0.7 \lesssim \epsilon \lesssim 1.8$. Despite of this, we where able to generate crystal structures based on one particle in the box for larger values of ϵ.
Furthermore we where able to determine the Radial Distribution function successfully for some points in the liquid phase and compared it with the theoretical one for spheres based on Ornstein-Zernike theory. We conclude that our Radial Distribution function for spheres is the same as the theoretical function from Ornstein-Zernike. We compared the Radial Distribution function for spheres with superballs with $1.15 \leq \epsilon \leq 1.2$ for differen values of η. The difference between EOS for spheres and superballs differs mainly in the heights of the first peeks and a little shift of all peeks in probably the direction of the outer radius. Besides all that, our Radial distribution functions where quit similar, due to fact that the superballs where still close to spheres.

6.2 Outlook

With the EOS for superballs a lookup table can be made for the parameters A, B, C and D of the model function by making a fit with sufficient precision. Our method may be easily extended upon to determine a phase diagram for different superballs, for which the particles may interact with soft potentials in additions to their hard particle overlap. In some experimental studies more cubic shaped particles are used as a starting point for syntheses, therefore it is useful to do simulations with superballs with $\epsilon \gg 1.8$. Due to the instability of the algorithm for $\epsilon > 1.8$ an improved algorithm is needed.
Furthermore we did FBMC and NPT simulations with a relative small amount of particles in the simulation box to get an approximation of the EOS and the radial distribution function. To get a better precision, more particle and longer simulations are needed. Some of the simulations we did, took already days to finish. Especially systems with high pressure where particles are packed closer together, which results in more overlap checks. We used a cell-list construction to reduce the simulation time, but this is still quit time consuming. To reduce more simulation time, the cell-list construction may be a starting point of parallelising the simulation program.

References

[1] M. Dijkstra. *Computational Materials Science*, pages 71–73. Soft Condensed Matter Group (Utrecht University), 2009.

[2] A. Donev. *PhD Thesis: Jammed Packings of Hard Particles*, pages 18–36. Princeton University, 2006.

[3] L. Filion, M. Marechal, B. Oorschot, D. Pelt, F. Smallenburg, and M. Dijkstra. Efficient method for predicting crystal structures at finite temperature: Variable box shape simulations. *Physical Review Letters*, 103:188302, 2009.

[4] D.E. Frenkel and B. Smit. *Molecular Simulation from algorithms to applications*, pages 9–32,48–49,111–132. Academic Press, San Francisco, CA, 2002.

[5] D. Gottwald, G. Kahl, and C.N. Likos. Predicting equilibrium strutures in freezing processes. *The Journal of Chemical Physics*, 122:204503, 2005.

[6] Y. Jiao, F. Stillinger, and S. Torquato. Optimal packings of superballs. *Physical Review E*, 79:041309, 2009.

[7] B. van Oorschot. *Master thesis Crystal prediction of hard dumbbell particles through Monte Carlo Simulations*, pages 13–30. 2009.

[8] R. van Roij. *Lecture Notes : Soft Condensed Matter Theory*, pages 1–24,34–36. Institute for Theoretical Physics, Utrecht University, 2008.

[9] R. van Roij. *Lecture Notes : Thermische Fysica 1*, pages 51–56. Utrecht University, 2008.

[10] 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*, 7:4139, 2011.

[11] D.J. Tildesley and W.B. Street. An equation of state for hard dumbell fluids. *Molecular Physics*, 41:85–94, 1980.