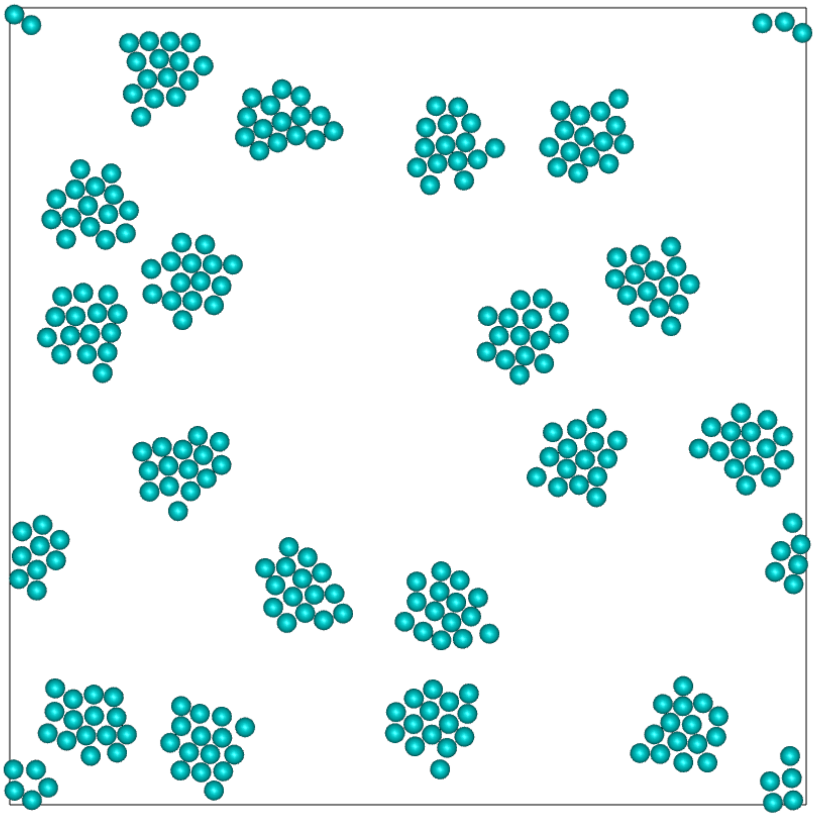
Universiteit Utrecht

Opleiding Natuur- en Sterrenkunde

Inverse design of cluster fluids

Bachelor Thesis

Jannes Vleming



Supervisor:

Prof. dr. ir. Marjolein Dijkstra
Debye Institute for Nanomaterials Science

June, 2021

Abstract

We use Iterative Boltzmann Inversion to discover pair potentials that yield cluster fluids. To do this we need a target radial distribution function, which we sample from a system with a many-body potential that was designed to yield cluster fluids. Prearranged clusters do tend to stay together with the potentials we find, but when starting from a random configuration, clusters do not form properly. When we slowly increase the potential with a random configuration the results are slightly better, at least the radial distribution function matches its target better. Our current simulation technique is perhaps not very well suited for the interaction between clusters, since it is difficult to move clusters using only single particle moves.

Contents

1 Introduction

Everything in our world is made out of materials, materials with vastly different properties: they can be liquid, solid or gaslike, or perhaps something in between. Some materials are very dense and others are not, some solids are fragile and others are flexible, some liquids are very viscous like honey and some are not, like for example gasoline.

What all materials have in common is that they are made out of small particles, like molecules. These particles come in different kinds of shapes and sizes. Sometimes they attract each other and sometimes they repel each other, and sometimes it is a mix of those two at different distances. The interaction of particles can be described by an interaction potential. Much research goes into looking how exactly particles interact with each other and thus finding this interaction potential, like for example in [1].

But what exactly is the relation between these two views: the properties of a material and the properties of the particles it exists of? If we know exactly how certain particles interact with each other, we would like to know how a system of many of those particles together would behave. For this system, we can in theory write down a differential equation which describes the system. One of the simplest potentials is the gravitational potential. Newton was able to solve the differential equation of two particles with a gravitational potential, if we add more particles however this becomes practically impossible. What we would like to do is for a system of many (about 10^{23} or more) particles with an arbitrary complicated but known potential, to describe the properties of the material that forms.

Statistical Mechanics seeks an answer to this question. This has been very successful and has solved many practical problems. This approach is not a general solution, a different problem often requires a different approach, and almost always we have to make approximations, for which it is not always obvious how valid these are. Since the advent of the computer an important technique within statistical mechanics was invented, namely computer simulations, using for instance the Metropolis algorithm. This allows us to use the same approach to describe materials of particles with arbitrary interaction potentials. Classical analytical methods are still useful as they provide more insight and require less computation time. The combination of analytical methods and simulations is very common and can be used to solve a very wide range of problems.

This is however only one side of the story: is it possible to, using only the description of a certain material, discover what particles it is made out of? Thus, given certain material properties, what potential must a particle have so that a system of these particles has those properties? With new experimental techniques to make all different kinds of particles, this question has only become more relevant. This problem is called inverse design, and it is exactly what we will be doing in this thesis. This does not mean we will not need any simulations. In fact, the inverse design technique we will be using needs a lot of simulations, since at every step of the inverse design method we do a simulation with a certain interaction potential, and based on the results of this simulation we change the interaction potential slightly until, hopefully, the simulation gives the right result. We will try to use this to find interaction potentials that will form cluster fluids, see for example the figure on the front page.

2 Theory

2.1 Canonical ensemble

Here we present a short description of the NVT-ensemble, also known as the canonical ensemble, which stands for systems where the number of particles N, the volume V and the absolute temperature T are fixed. Let us say we work in a d-dimensional space and let Λ be a region of space with volume $\int_\Lambda \mathrm{d}\mathbf{r} = V$. The phase space of this system is the set of phase space points $\mathbf{\Gamma} = (\mathbf{r}^N, \mathbf{p}^N)$, where $\mathbf{r}^N$ is a n-tuple of all the positions of the N particles in Λ, and $\mathbf{p}^N$ is a n-tuple of all the particle momenta. We also have a interaction potential Φ, which has units of energy, for each configuration of particles $\mathbf{r}^N$. The Hamiltonian of the system is defined as

$$H(\mathbf{r}^N, \mathbf{p}^N) = \Phi(\mathbf{r}^N) + \sum_i \frac{|\mathbf{p}_i|^2}{2m},$$

where m is the mass of an individual particle. Each phase space point $\mathbf{\Gamma}$ has a probability to occur, which is proportional to the Boltzmann factor $\exp(-H(\mathbf{\Gamma})/kT)$, where by definition $k = 1.380649 \cdot 10^{-23} J/K$ is called the Boltzmann constant. The probability of finding the system in a certain positional configuration can easily be shown to be proportional to $\exp(-\Phi(\mathbf{r}^N)/kT)$ by integrating over the momentum part of the phase space.

2.2 Reduced units

For our purposes it is practical to work with reduced units, for this we need a length scale σ, which will be the diameter of the particles. With this we can divide all positions by σ, thus for each position $\mathbf{r} \in V$ we define $\mathbf{x} = \mathbf{r}/\sigma$, which is unitless. We also define $\bar{\Lambda} = \{\mathbf{x} \in \mathbb{R}^d \,|\, \sigma\mathbf{x} \in \Lambda\}$, which is unitless and has a volume $\bar{V} = \sigma^{-d} V$. The reduced interaction potential is defined as

$$U(\mathbf{x}^N) = \frac{\Phi(\sigma\mathbf{x}^N)}{kT},$$

this is a function on $\bar{\Lambda}^N$. Thus, when we have a certain configuration of particles described in reduced coordinates $\mathbf{x}^N$ we can calculate the original coordinates $\mathbf{r}^N$ by multiplying with σ, with this we can calculate the interaction potential Φ, and if we multiply this with $\frac{1}{kT}$ we get a unitless quantity which we call U. Now it is easy to calculate the Boltzmann factor using

$$\exp(-\frac{1}{kT}\Phi(\mathbf{x}^N)) = \exp(-U(\mathbf{x}^N)).$$

We define $f : \bar{\Lambda}^N \to \mathbb{R}$ to be proportional with this Boltzmann factor but normalized, meaning $\int_{\bar{\Lambda}^N} f(\mathbf{x}^N)\mathrm{d}\mathbf{x}^N = 1$. From this point on we will only work with reduced units.

2.3 Pair potentials

In principle, U can be anything, but there is an interesting special subset on which we will focus, namely (isotropic) pair potentials. These are interaction potentials we can write as $U(\mathbf{x}^N) = \sum_{i<j} u(x_{ij})$, where $x_{ij} = |\mathbf{x}_j - \mathbf{x}_i|$ is the inter-particle distance and $u : \mathbb{R}_{\geq 0} \to \mathbb{R}$

is also called the pair potential. Interaction potentials that are not pair potentials are called many-body potentials. The use of this class of interaction potentials greatly reduces the complexity, also note that many potentials in nature are pair potentials, like the gravitational or Coulomb potential. One pair potential we would like to highlight is the hard-sphere potential, $u(x) = \infty$ if $x < 1$ and $u(x) = 0$ if $x \geq 1$, where "∞" is a symbol such that the exponential of a negative number times ∞ is zero.

2.4 Metropolis algorithm

A fundamental problem is to calculate or approximate ensemble averages of quantities. Say that we have a quantity $A : \bar{\Lambda}^N \to \mathbb{R}$, its ensemble average is

$$\langle A \rangle = \int_{\bar{\Lambda}^N} A(\mathbf{x}^N) f(\mathbf{x}^N) \mathrm{d}\mathbf{x}^N.$$

In case there is no analytical solution available, which is practically always, we will have to settle for some sort of numerical integration. A naive approach is to set out a grid over the entire domain of the integral and calculate a Riemann sum. This grid would have to be rather fine because interactions between particles can also be rather fine. The reason this does not end up working is the dimension of the domain of the integral, for N particles in an d dimensional box, the dimension of the domain of the integral is $N \cdot d$, and thus for a grid with in each dimension k steps, the number of grid points is $k^{N \cdot d}$. With a number of particles N of a few hundred this quickly puts astronomical scales to shame.

A better way to do the numerical integration is to use Monte Carlo methods, which involves a random component, to get a better and better approximation over time. One important Monte Carlo method is the Metropolis algorithm. The idea of the Metropolis algorithm is to randomly sample a sequence of configurations such that the probability of sampling a configuration is the probability of finding the system in that configuration. We pick an initial configuration $\mathbf{x}_0^N$, and then randomly generate a sequence $\mathbf{x}_1^N, \mathbf{x}_2^N, \mathbf{x}_3^N, \ldots$ of configurations. This sequence will be a Markov chain, meaning that the probability of finding the sequence at a certain step in a certain configuration only depends on the previous configuration that was reached.

Assume we have a configuration $\mathbf{x}_n^N$ at step n in our Markov chain, we then pick one particle at random and move it to a uniformly random position within a certain box centered around its original position with side lengths of δ to create a new configuration $\mathbf{x}'^N$. We now either accept or reject the change, if we accept the change then we set $\mathbf{x}_{n+1}^N = \mathbf{x}'^N$, and if we reject then we set $\mathbf{x}_{n+1}^N = \mathbf{x}_n^N$. We accept with a certain probability $\beta(\mathbf{x}_{n+1}^N \to \mathbf{x}'^N)$, this probability is chosen such that the stable probability distribution of our Markov chain reaches the desired probability distribution as dictated by the NVT-ensemble. To find the acceptance probability we use the so called detailed balance condition.

2.5 Detailed balance

Detailed balance is a condition which ensures that, over time, the sequence reaches the right distribution [2]. It requires that for any two configurations a and b the sequence can reach, the probability of finding the system in configuration a times the probability of the chain transitioning from a to b is equal to the probability of finding the system in b times the probability of transitioning from b to a, or in a formula $f(a)\pi(a \to b) = f(b)\pi(b \to a)$. The transition probability $\pi(a \to b)$ is the chance of proposing a trial move to b while in a times the chance of accepting this trial move, in a formula $\pi(a \to b) = \alpha(a \to b)\beta(a \to b)$. In many cases, as in our case, α is symmetric, meaning $\alpha(a \to b) = \alpha(b \to a)$ for all a and b. Because we move one particle in a certain box centered around its old position, and from the new position we can always pick the same particle and move it in the opposite direction to get the old position again, thus all inverses of possible moves are possible. Since we pick the particle uniformly and the position in the box uniformly, all possible moves are equally likely, thus indeed α is symmetric in our case. For a possible move $a \to b$ we thus have

$$\frac{\beta(a \to b)}{\beta(b \to a)} = \frac{f(b)}{f(a)} = \exp[-(U(b) - U(a))].$$

We are now free to choose β such that this equality is satisfied, we will do so by maximizing β with $\beta(a \to b) = \min(1, \exp[-(U(b) - U(a))])$, such that our sequence moves more quickly through phase space. Now that we have a sequence, we can sample from it using

$$\langle A(\mathbf{x}^N)\rangle = \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} A(\mathbf{x}_i^N).$$

An advantage of the Metropolis algorithm is that we only need the quotient $\frac{f(a)}{f(b)}$ and never the actual value of f, and we know this quotient very well from our treatment of the NVT-ensemble, thus saving us the step of normalizing f.

2.6 Practical considerations in the Metropolis algorithm

We have not specified what the initial configuration $\mathbf{x}_0^N$ should be and if it matters. In general, it does matter and our ensemble averages might depend on this choice. If the sequence can move through the entire phase space, meaning that from every configuration, all possible other configurations can be reached with a non-zero probability in a finite number of steps, the initial configuration does not matter for an infinite sampling. For low enough densities this should be the case, but we will just assume it for this thesis.

For an infinite number of steps the initial configuration $\mathbf{x}_0^N$ might not matter and the ensemble averages will be the same. But since we work with a finite number of steps (being able to read this means this thesis has in fact ended), choosing it right can give a better result. Our sequence will spend most of its time in configurations that are likely to occur, thus it pays to choose an initial configuration with a low U value. This way the first few or many steps it does not only reach very unlikely configurations. One should at least avoid picking an initial configuration with vanishing probability.

The choice of the side lengths of the box within which we perform the particle move, δ, is just like our initial position not important in case of an infinite sequence, but the right choice can give a better result in fewer steps. Most of the phase space is, especially in systems with a high density, extremely unlikely, if there are for example two particles that overlap. Therefore if we move particles far away from their original position by choosing δ too high, the acceptance rate, that is the fraction of moves that are accepted, will be very low, wasting computational time on generating moves that are in the end rejected. If δ is too low on the other hand, particles are barely moved and the phase space is not explored as fast as possible, this is characterized by a very high acceptance rate. We therefore need a value for δ which keeps the acceptance rate close to one half. We do this by either fixing the right δ in advance, or fine-tuning it during the simulation based on the acceptance rate. This continuously fine-tuning based on the result of the sequence breaks some of the mathematics, for example the Markov property, thus we will have to use a fixed δ when sampling the ensemble averages.

2.7 Periodic boundary conditions

The Metropolis algorithm helps us to simulate a system of particles, with the purpose of calculating ensemble averages. As is usually the case with computer simulations, the greatest constraint we face is simulation time. For the Metropolis algorithm, this time depends on the number of particles. This is because we have to make a certain number of attempted moves for each particle before the system reaches equilibrium Also, in equilibrium we again need a certain number of attempted moves per particle before the system is reasonably uncorrelated with the last sampling point, in order to sample ensemble averages. For this reason we can only afford to simulate a system of a limited number of particles, say 2048 in [3], or in our case about 300. This is in sharp contrast with bulk systems in nature in which we are actually interested, a cup of water for example consists of roughly 10^{25} molecules.

We will not worry too much about this, still we should try to do the best we can with our limited resources. One important aspect is the boundary of the system. In a small system a relatively large portion of the particles is located at the boundary of the system, and thus at one side there are no other particles for them to interact with. This can have a serious impact on the characteristics of the system. A trick to overcome this is to use periodic boundary conditions, the idea is to copy the system in all directions to create a grid, and let particles interact between different systems. For this, the simulation volume $\bar{\Lambda}$ has to be a box, in our case $\bar{\Lambda} = [-\frac{1}{2}L, \frac{1}{2}L]^d$, if we use d dimensions. We then use the nearest image convention. To this end we calculate the distance between two particles by minimizing the distance vector while trying to add or subtract the simulation box side length L in each dimension. To make sure particles do not interact with other particles more than once, we impose a maximum interaction distance, which cannot be more than one half times L. If the distance x between two particles is more than this so called cut-off distance, their interaction potential is zero. If a move of the Metropolis algorithm moves the particle outside of the particle box, we place it back inside the box by adding or subtracting multiples of L in each dimension. Periodic boundary conditions help to simulate a relatively small system without having any boundaries.

2.8 Radial distribution function

We will now introduce the Radial Distribution Function (RDF), which is the relative density at a certain distance of a particle compared to that of an ideal gas. The RDF is thus a function $g : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$. If we compare it to the pair correlation function

$$\rho^{(2)} : \bar{\Lambda}^2 \to \mathbb{R}_{\geq 0}, \quad (\mathbf{x}, \mathbf{x}') \mapsto \left\langle \sum_{i=1}^{N} \sum_{j=1, j \neq i}^{N} \delta(\mathbf{x} - \mathbf{x}_i)\delta(\mathbf{x}' - \mathbf{x}_j) \right\rangle$$

we have $\rho^{(2)}(\mathbf{x}, \mathbf{x}') = \rho^2 g(|\mathbf{x} - \mathbf{x}'|)$, with $\rho = N/\bar{V}$ the density. This is because g assumes a fixed particle we have to multiply one time with ρ for the probability of finding a particle at $\mathbf{x}$, and since g is relative to an ideal gas it is divided by ρ which we cancel by multiplying another time by ρ. At large distances, densities are uncorrelated meaning $\rho^{(2)}(\mathbf{x}, \mathbf{x}') = \rho^{(1)}(\mathbf{x})\rho^{(1)}(\mathbf{x}') = \rho^2$ thus $\lim_{x \to \infty} g(x) = 1$. The RDF is useful to characterize inter-particle distances and in a system with an isotropic pair potential the thermodynamics follow completely from the RDF. In practice we obtain the RDF by making a histogram of inter-particle distances with a simulation and normalize it by dividing it by a similar histogram for an ideal gas.

2.9 Iterative Boltzmann Inversion

For each pair potential $u : \mathbb{R}_{\geq 0} \to \mathbb{R}$ we now know how to approximate the RDF $g : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$. An interesting problem is if there is a way to inverse the process, thus for a given $g(x)$ at a certain fixed density find a pair potential $u(x)$ such that if we simulate the system with u we again get back g. This problem is further motivated by the importance of the RDF, since it encodes a lot of properties of the system we might be tempted to think that systems with similar RDFs also behave similar in a lot of other ways. Thus, if we have a system with desired properties, we could measure its $g(x)$ and if we could find a $u(x)$ with approximately the same $g(x)$, our hope at least is that a system with u would behave similarly as the original system, even while the original system might have had a many-body potential. Not all RDFs have a corresponding pair potential, but if it exists, Iterative Boltzmann Inversion [3] is a simple method to find it.

Let $g_{tgt}(x)$ be the targeted radial distribution function. We have to make an initial guess for the pair potential, for example we can take what we will call the inverse Boltzmann potential

$$u_0(x) = -\log(g_{tgt}(x)).$$

The idea is for each pair potential u_i to simulate it at density ρ and measure its RDF g_i, and then alter the pair potential in order to bring g_i closer to g_{tgt}, as in

$$u_{i+1}(x) = u_i(x) + \alpha \log\left(\frac{g_i(x)}{g_{tgt}(x)}\right),$$

where α is a small positive constant. If at a certain distance r from a particle $g_i(x)$ is too high, that is, higher than $g_{tgt}(x)$, then the pair potential at that distance is increased, making it less likely there will be particles there and thus decreasing the density there, and vice versa.

3 Approach

Our goal is to find a pair potential which at a certain density yields a cluster fluid of a certain cluster size. We define an ideal cluster fluid to be a system of particles that consists of clusters of a fixed number of particles that are packed closely together like a liquid, while the clusters themselves behave like a hard-sphere fluid, thus their only interaction is short-ranged and repulsive. Our approach to finding such a pair potential has two steps: the first being the simulation of a system with an artificial many-body potential specifically designed to promote an ideal cluster fluid in order to measure its radial distribution function (RDF), and the second being the previously described Iterative Boltzmann Inversion (IBI) with the aim of finding a pair potential with the same RDF, which will be our final result. This rests on the assumption that there exists a pair potential with a similar enough RDF as the one of the many-body potential, and the assumption that their RDFs being reasonably similar would mean that other properties would also be reasonably similar.

Our simulations will be two-dimensional, according to the principle that in education one should use the simplest example, be it the lowest dimension, of the problem that still has all the complexity of the general problem, and also to save simulation time. There are two main parameters for our cluster fluid, namely the cluster size n (number of particles in each cluster) and the density $\rho = N/\bar{V}$. For each choice of n and ρ, we expect to need a different pair potential and also a different RDF. For n we try three different values, namely $n = 5, 15, 25$, in order to get a view of a broad range of different cluster sizes and to roughly see in what region our approach works. We let our simulation box be

$$\bar{\Lambda} = \left[-\frac{1}{2}L, \frac{1}{2}L\right]^2,$$

and for the side length of the simulation box we choose $L = 40, 80, 120$, again to have a broad spectrum of possible densities, note that L is unitless, and this has the meaning of number of particle diameters. This gives us nine different combinations and thus we look for nine different pair potentials. The density is $\rho = N/L^2$, where for N, the total number of particles, we pick $N = 300$, which is divisible by all values of n. The value of N is an important factor in the simulation time; that said, in general, we try to simulate bulk systems, thus values of N that are too small might give wrong results, we therefore pick N as high as our computational resources reasonably allow us to. This general approach is adapted from a paper [3] where they look into three dimensional cluster fluids and compare the properties of the pair potential they found using IBI with a more classical model for a pair potential that also yields cluster fluids.

3.1 Many-body potential

The result of this step should be a RDF of an ideal cluster fluid, but especially it should be a function that is smooth, thus without sharp peaks. This requirement is supposed to help the reverse engineering process we will do later on the function. We can only hope to find a pair potential with roughly the same RDF as the one we find here, therefore the RDF should be smooth such that small details do not affect its general shape. Also, from a more practical point of view, our RDF will have a finite resolution, even more so, the lower the resolution, the less simulation is time required to sufficiently sample the RDF, and thus for this reason we want to avoid having sharp peaks in our target RDF.

We design a many-body potential and then simulate the system with the Metropolis algorithm. The first component of our many-body potential is a hard-sphere pair potential, such that the particles do not overlap.

3.2 Keeping clusters together

Before the simulation starts, we assign the particles to clusters and then one component of the many-body potential is designed to keep the cluster together. We thus need a measure for how compact our cluster is. To reward more compact clusters, we define

$$R = \sum_{i<j\in\text{cluster}} x_{ij}^2,$$

thus R is the sum of the squares of all inter-particle distances within one cluster. We select two constants, A and R_{tgt}, and for each cluster add to our many-body potential the term $A(R-R_{tgt})^2$. The constant R_{tgt} describes the targeted, or desired, value of R, such that each value of R that is different from R_{tgt} costs energy and thus is less likely. The constant A is positive and describes how strongly wrong values of R are punished.

For each choice of n and ρ we have to make a choice for A and R_{tgt}. We could simply pick A very high and R_{tgt} very low to ensure that the clusters are as compact as they can be, but the chances of the resulting RDF being reproducible with a pair potential would decline since the particles would not have a lot of space to move and thus the RDF would have very sharp peaks. In general our approach will be to choose the lowest potential that works. To this end we want to choose A as low as possible and R_{tgt} as close to the resulting real value of R as possible, while still keeping the particles in the clusters close enough together. Ideally one would want to choose A and R_{tgt} just so that R_{tgt} is slightly less than R, such that there will be a balance between the higher entropy of a less compact cluster and the lower energy of a more compact cluster, for the most realistic cluster fluid. For practical reasons, we have made our choice slightly different, we chose R_{tgt} as low as possible under the constraint that in the simulation the resulting R becomes equal to R_{tgt}. In this case we have a compact cluster that has no sharp peaks in its RDF, but with the advantage that the precise value of A is not as important, and we can just choose one value that seems reasonable and use it for all cluster sizes and densities, we choose $A = 250$. This leads to values of $R_{tgt} = 4.5, 24, 52$ for the respective cluster sizes n.

3.3 Separation between clusters

The third component we add to the many-body potential is a term to keep the clusters from touching each other. We could for example keep track of the centers of mass and add increase the many-body potential if the distance between the centers of mass becomes too small. Within periodic boundary conditions, it is not trivial and in some cases not even possible to determine the center of mass of a cluster. One nice way to calculate the center of mass in one dimension is to transform the line segment the particles are on to a circle, then calculate the center of mass in two dimensions and then intersect the line segment between this point and the center of the circle with the circle to finally transform this intersection back from the circle to the original line segment. It is however not necessary to use such a complicated method, and the trigonometric functions that would be needed could take up a lot of simulation time.

Most of the time the cluster is already quite compact and the rest of space quite big, thus we make an approximation. We calculate the center of mass in the normal way, but in case a cluster is on the edge of the simulation box, we move all particles that have crossed the edge over to the rest of the cluster before calculating the center of mass in the normal way. In practice, we pick one pivot particle for each cluster, and for each dimension if a particle is further away from the pivot than half the box side length we move it over with one box side length and then calculate the center of mass. We calculate distances between two centers of mass the same way as with particles within periodic boundary conditions. To the many-body potential we add for each pair of clusters with a distance between them of x_{COM} the term

$$\frac{B}{x_{COM}} \exp(-x_{COM}/z),$$

where B and z are positive constants. For low values of z and corresponding values of B this potential approaches a hard-sphere potential. We pick $z = 0.12$ for all situations. For this z, in each situation, we choose B as low as possible while keeping the clusters from touching each other, yielding values of $B = 10^{12}, 10^{21}, 10^{27}$ for the respective values of n.

We do not only label the particles to be in clusters, in the initial configuration we also place the particles physically in clusters. This is to save time simulating the process of the particles assembling themselves in the predefined clusters, which could take a long time if the particles in the clusters were not already close together. Also we do not allow the particles to overlap in the initial configuration (and also not later on). We place the first particle of the cluster randomly in the simulation box without overlap with other particles. For the other particles we keep trying random positions close to the first particle until no overlap occurs.

We then simulate the system with the Metropolis algorithm, each step we propose a new position that differs slightly from the previous one and based on the energy difference it causes accept or reject it with a certain chance. We make two different kinds of moves, 90% of our moves are particle moves and 10% are cluster moves This is because we have more particles than clusters, but the exact ratio is not important. Particle moves, as described before, are moves where we pick one particle randomly and move it randomly in a small box around its original position, for the side length of this box we choose 0.01 in order to keep the acceptance rate close to 0.5. For a cluster move we pick one cluster at random, pick a random displacement in a box centered around the origin and give every particle in the cluster that displacement. In our case we let this box be the size of the simulation box, but this could also be a smaller box.

In theory, after each step, rejected or accepted, we sample the quantities we want to measure, in practice however, we do this sampling after every few steps. Sampling can be computationally expensive, and after a single step most particles are still in the same place thus doing the same computations multiple times. Therefore in our case we sample every $N = 300$ steps, where every step is either a particle or cluster move that is either accepted or rejected. The quantity we sample is the RDF, but we also keep track of the energy and average value of R over time, to monitor if we have reached equilibrium.

3.4 Measuring the RDF

To measure the RDF we make a histogram of the inter-particle distances and then normalize it by dividing it by a similar histogram of an ideal gas. We need a range $[a, b]$ and also a resolution n_{res}, for this range we take $[1, x_{cut}]$ where x_{cut} is a parameter we will describe in the next section. The lower bound is 1 because below this value, due to the hard-sphere potential, $g(x)$ is 0. The resolution should be high enough to smoothly describe the RDF and all its features in detail. There is also significant virtue in choosing it low enough, since a lower resolution means a larger interval size to count particle pairs and thus we need fewer simulation steps to get an accurate estimate of the function value. Thus, the resolution is inversely proportional to the simulation time to sample the RDF.

Then we divide $[a, b]$ in n_{res} equally sized subintervals. We measure the average number of particle pairs whose mutual distance is in such a subinterval. When we multiply this number with 2, the number of particles in a pair, and divide it by the total number of particles, we get the average number of particles that is at a certain distance of a fixed particle. We then divide it by the expected number of particles at a certain distance of an ideal gas, which is the density $\rho = N/\bar{V}$ times the volume around a particle where a particle would have the correct distance, that is $\frac{4}{3}\pi(r_{up}^3 - r_{low}^3)$ with r_{up} and r_{low} the upper and lower bound of the subinterval. We consider the obtained value for a subinterval the function value of $g(x)$ at the center of the subinterval.

3.5 Iterative Boltzmann Inversion

Now that we have a RDF the next step is to find a pair potential with the same RDF. For this we use Iterative Boltzmann Inversion, for which we already gave a short description in the theory section. One thing to notice is that our many-body potential had a hard-sphere component, this has the effect on the RDF that $g(x) = 0$ for all $x < 1$. To make sure our pair potential has the same RDF in this region of x, we add a hard-sphere potential to our pair potential before doing anything else. Thus from now on we are only concerned with $g(x)$ for $x \geq 1$. We determine a maximum range x_{cut} such that for $x \geq x_{cut}$ we have $g(x) = 0$. We have to do this because of the periodic boundary conditions To make sure a particle can only interact with other particles once, we have to choose $x_{cut} \leq L/2$ with L the side length of the simulation box. There are also other reasons to make this adjustment, we want the clusters in the cluster fluid to only have short range interactions. Perhaps the most important reason is to save on simulation time, if x_{cut} is smaller fewer interactions have to be calculated for each particle move.

To determine what x_{cut} should be we look at the diameter of the clusters, if we assume a circular perfectly packed cluster of n particles, the cluster has a diameter of $\sqrt{\frac{3\sqrt{2}}{\pi}n}$ in terms of particle diameters. For our largest cluster size $n = 25$ this gives a cluster diameter of 5.8. To allow particles on one side of the cluster to interact with a neighbouring cluster on the other side we pick a x_{cut} of 9 for all cluster sizes. In principle we only update the pair potential in the range $x \leq x_{cut}$. We want to avoid sharp discontinuities in the pair potential, therefore every time after we update the potential we shift it vertically such that $u(x_{cut})$ is zero, just like values just above x_{cut}.

To start the algorithm we need an initial guess for the pair potential. The simplest guess one can imagine is $u(x) = 0$, thus we will try this. Another try is to use make use of the Boltzmann distribution directly. If we consider one fixed particle which interacts with the other particles around it with a pair potential $u(x)$, but the other particles do not interact with each other, we can calculate the distribution of particles around this one particle, and it would be a Boltzmann distribution $g(x) = c \cdot \exp(-u(x))$ with some constant c. Thus, if we choose $u(x) = -\log(g_{tgt}(x))$ for a targeted RDF g_{tgt}, this pair potential will yield the targeted radial distribution function g_{tgt}. Of course the other particles do interact with each other, thus it is not an exact solution but we will use it as an initial guess.

Using this initial guess u_0, we will now improve it in the subsequent steps $u_1, u_2, \ldots,$ but to do this we need to measure the RDF for each pair potential $g_0, g_1, g_2 \ldots$. We take a system of $N = 300$ particles as before. For $u_0(x) \equiv 0$ we initialize the particles in our system randomly with the constraint that the particles do not overlap, IBI with this initial pair potential and initial configuration will be called IBI0. For $u_0(x) \equiv -\log(g_{tgt}(x))$, we initialize the particles in clusters, in fact, we use the final configuration of our simulation of the many-body potential, IBI with this initial pair potential and initial configuration will be called IBI1. The choice of this initial configuration should be made such that the system with the pair potential u_0 is already close to its equilibrium in the initial configuration. This way we do not have to simulate this process of reaching the equilibrium. In later steps we use as initial configuration the final configuration of the previous step. This is because we only alter the pair potential slightly, and thus if the last step had reached its equilibrium, this one will almost be in its equilibrium.

The targeted RDF is only known at points (the centers of certain intervals), and thus we limit ourselves to describe the pair potential u at those same points. To find the potential in between two points we use linear interpolation. Below the lowest interval, the potential is infinite and above the highest interval the potential is zero. In the Metropolis algorithm, the maximum distance δ with which we make proposed particle moves is a parameter we continuously update depending on the acceptance rate of the moves, if too few moves get accepted we decrease the parameter and if too many moves are accepted we increase it.

We can now simulate the system and measure its RDF, we then use this to update the pair potential as described in the theory, with a parameter α. This α determines the speed at which u evolves. A higher speed does not always work, because small errors in sampling the RDF would cause large errors in u if α is too large, but also for other reasons that are more specific for the system. The best approach is to simply try out some values of α and pick the highest that works, we have settled on 0.15. We do 200 consecutive steps. Apart from the RDF and the energy over time we also measure the cluster size distribution.

3.6 Cluster classification

To automatically classify clusters we need a formal definition of what we consider to be a cluster of particles. Two particles are directly connected if and only if they are close enough together, we take the upper limit for this to be a distance of 1.3 times the particle-diameter between the centers of the particles. Two particles are indirectly connected if and only if they are either directly connected, or there exists a particle to which they are both indirectly connected. We call the equivalence classes of indirect connection clusters.

3.7 Last step

Once we have obtained a pair potential with the IBI, we use it to simulate a system that starts either with all particles at random positions, the final configuration of the system with the many-body potential or the final configuration of the system of the IBI. We simulate for more cycles than in the IBI to make sure the system is in equilibrium and the RDF is sampled accurately. We use these results to see how well our approach has worked.

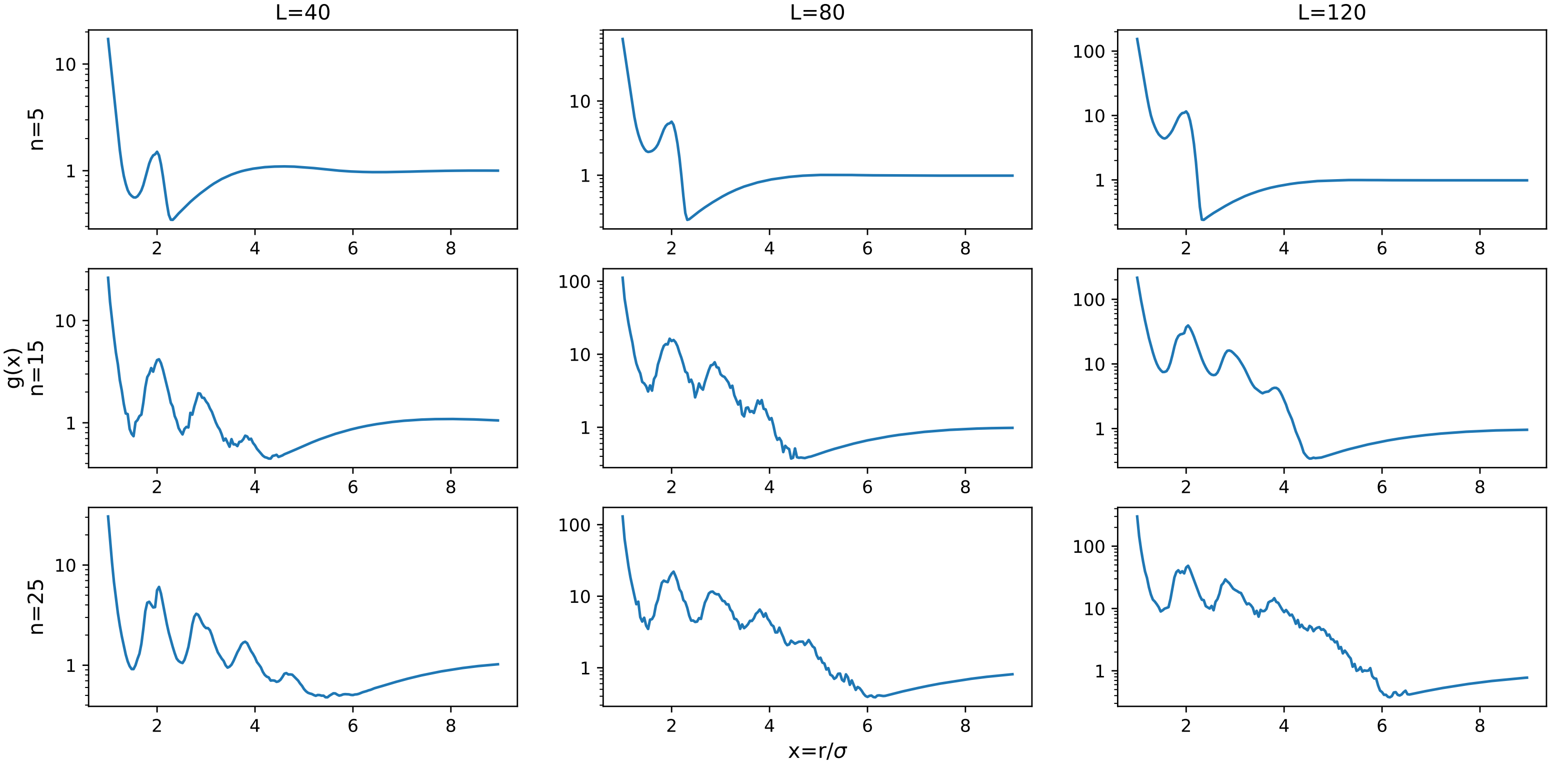


Figure 1: Radial distribution function g as a function of x of cluster fluids as they occur in a system with the many-body potential, for several values of the cluster size n and simulation box length L. These will be our targets during the Iterative Boltzmann Inversion.

4 Results

In figure 1 we can see the results of measuring the RDF of systems with many-body potentials, which we will later try to reproduce with IBI. For large $x = r/\sigma$ we see $g(x)$ approaching 1, as should always be the case with the RDF. For large cluster sizes this is not very clear since we do not measure the RFD over a large enough range. We have not included the part $x \in [0, 1]$, because of the hard-sphere component of the particle here $g(x)$ is zero.

The general shape of the graph is smooth, meaning we have sampled the RDF long enough, although for $n > 5$, some graphs have some small peaks, which could be solved by sampling for an even longer time. The absence of very sharp and very high peaks means our many-body potential is at least not too high with the current parameters, since the particles have enough space to move. It seems like the resolution of the measurement was high enough to capture all the important information about shape of the graph.

Close to the particle (for small x) the $g(x)$ is high, specifically more than one, meaning that particles are often in clusters and one is more likely to find another particle right next to a particle than at a random point in space. When we move to higher values of x we see a few peaks, which are the layers of particles around one particle. These peaks get lower meaning

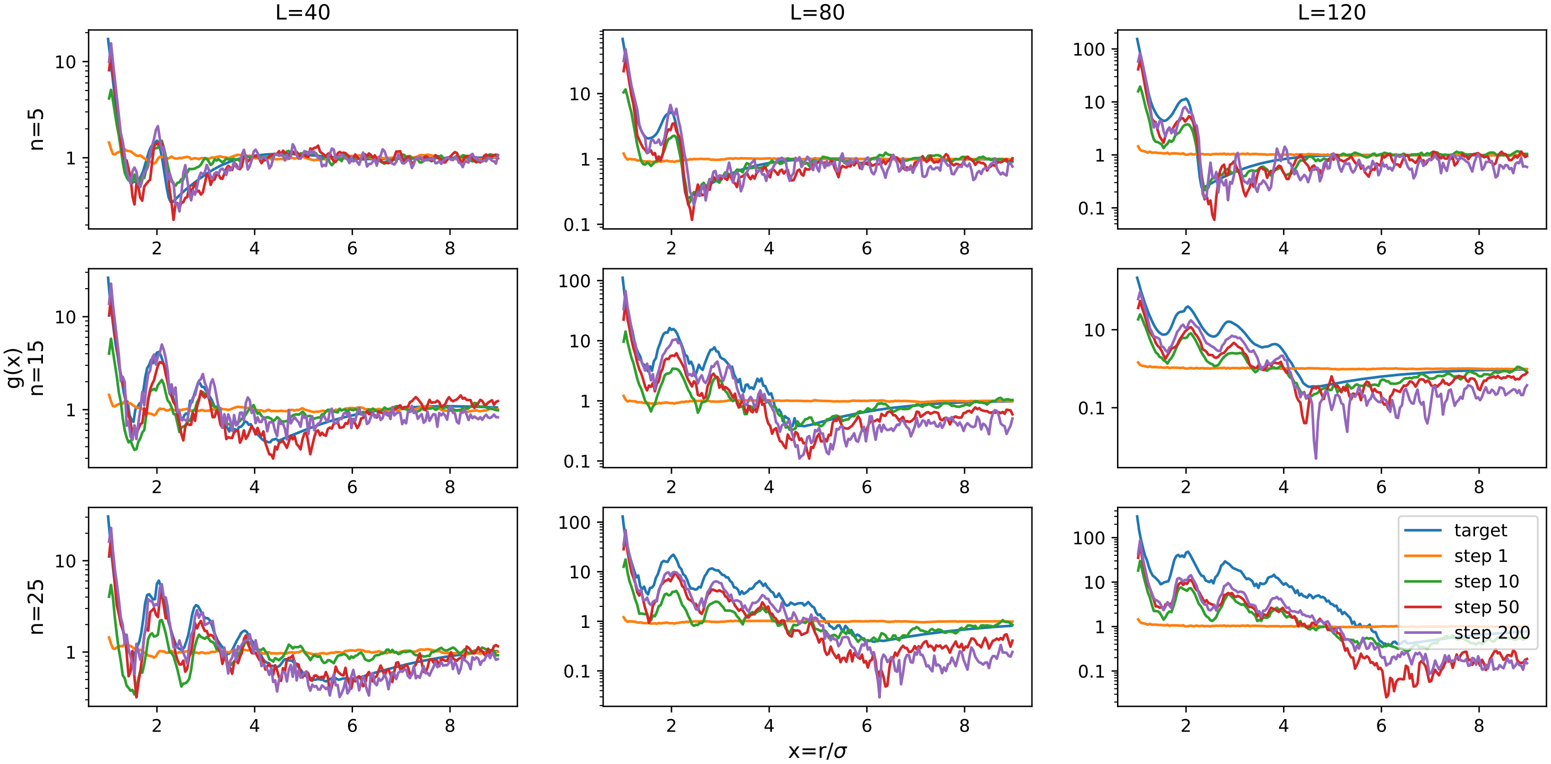


Figure 2: Radial distribution functions g as functions of x of the many-body potential (target) and several steps of the Iterative Boltzmann Inversion starting with a zero potential and all particles starting at random positions, for several values of the cluster size n and simulation box length L.

larger layers are less likely to occur, since some particles are on the edge of the cluster and thus these layers are not complete for them. The lowest point in the graph is at the diameter of the cluster, which matches the approximation $\sqrt{\frac{3\sqrt{2}}{\pi}n}$ we made in the Approach quite well. This point is the lowest point of the graph since at this x no particles occur that are in the same cluster, but other clusters are repelled slightly and thus are also unlikely to occur there. At even larger x other clusters do start to appear and the graph goes to one, which corresponds to the average density.

In figure 2 we have shown the RDF of a few intermediate steps in the Iterative Boltzmann Inversion which started with the zero potential (IBI0). The first thing to notice is that the shape of the graphs is quite rough, and thus a longer sampling could be useful. However, this is partly planned since we only update the pair potential slightly each step and thus one step we might move the potential slightly in the wrong direction because of this noise, but the next few steps will correct this, on average the potential will move in the right direction.

As more steps are made the RDF starts to match the targeted RDF more and more. For high densities and small cluster sizes, the end result is quite good. In some cases the RDF

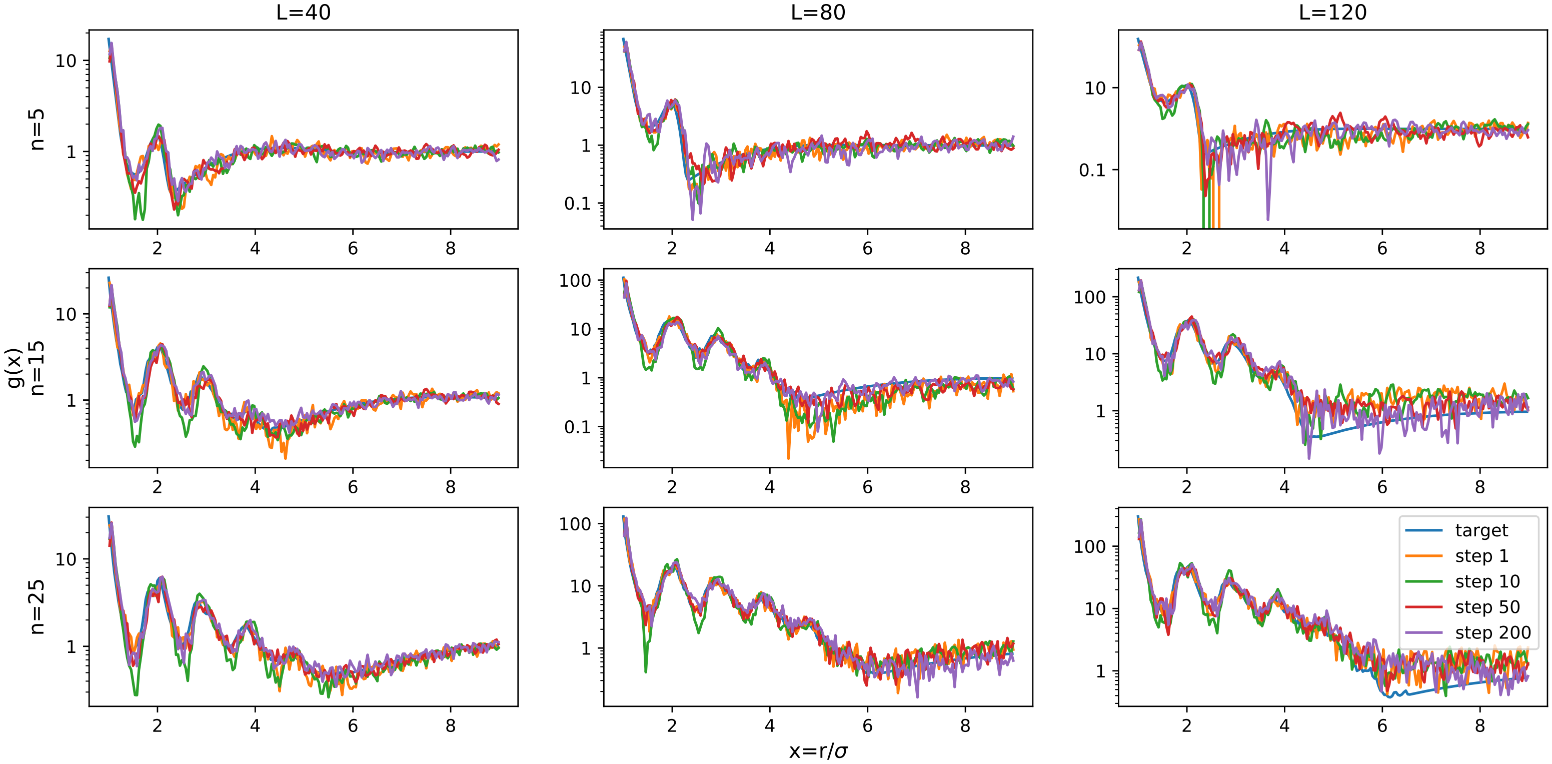


Figure 3: Radial distribution functions g as functions of x of the many-body potential (target) and several steps of the Iterative Boltzmann Inversion starting with a nonzero pair potential and the particles starting in clusters, for several values of the cluster size n and simulation box length L.

stays well below the targeted RDF. In the region where x is less than the diameter of the cluster, the minimum of the target graph, the shape of the graph is quite good, but it is shifted down. This means there probably are clusters and they are roughly the right size, but there are not enough of them. Thus, a lot of particles are not in clusters and are floating freely or perhaps are part of a very small cluster. For the region where x is larger than the cluster diameter, the low RDF tells us that the clusters probably repel each other more than is the case in the many-body potential system.

In figure 3 we have shown the RDF of a few intermediate steps in the Iterative Boltzmann Inversion which in this case started with the inverse Bolzmann distribution (IBI1). We again see a lot of noise in the graphs. Since the particles start in clusters, the RDF matches the targeted RDF from the start, and this remains to be the case for later steps as well. In some cases there is some deviation from the targeted RDF in the region where x is greater than the minimum of the graph. If we look carefully we can see that this deviation is already present in the first step but gets less over time. This deviation is thus a result of the initial configuration we chose, we only chose one configuration and in this particular configuration the clusters were on average slightly too close or slightly too far apart. The effect is visible

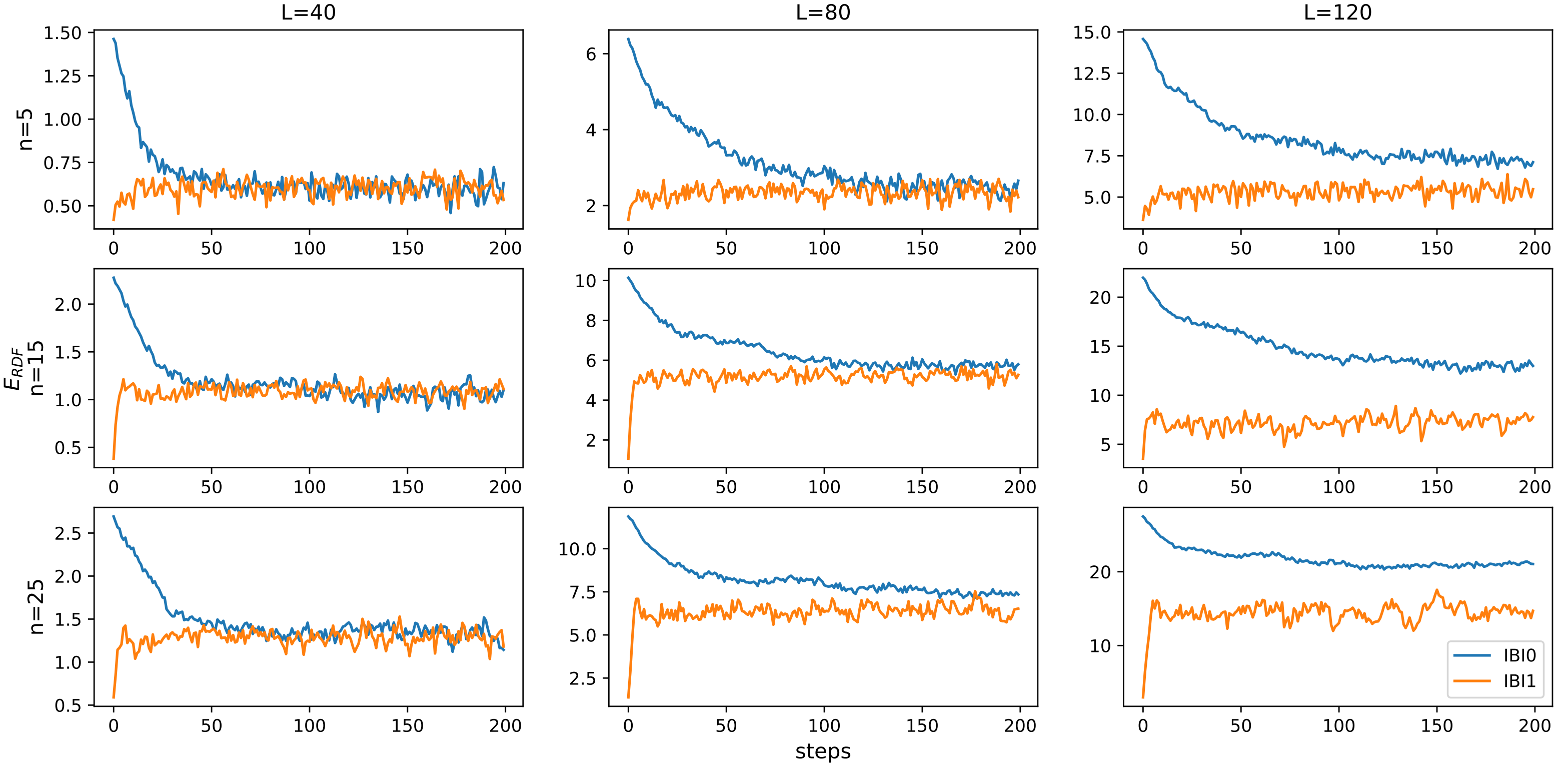


Figure 4: Root mean square difference between the target radial distribution function and the radial distribution functions of the steps of the Iterative Boltzmann Inversion, starting with a zero pair potential (IBI0) and a nonzero pair potential (IBI1), for several values of the cluster size n and simulation box length L.

because we do not have a lot of clusters, and thus there can be some random errors.

In figure 4 we measured the distance between the targeted RDF and the RDFs of the steps of the IBI. We measured this distance according to

$$E_{RDF} = \sqrt{\frac{1}{|x_{cut} - 1|} \int_1^{x_{cut}} (g(x) - g_{tgt}(x))^2 \mathrm{d}x}.$$

For IBI1, we have already seen that the RDF matches the target RDF from the start. In this figure however we see that there is some nonzero distance, this is because of the noise in the measurements of the RDF during the IBI. We see that this distance does not change over time, thus almost no progress is made in matching the target RDF. Despite this noise it is still possible to see if the graph of IBI0 matches the target, since the error the noise causes in the distance is more or less added to the real distance. To see if the IBI0 matches the targeted RDF, we have to look for where the graphs of IBI0 and IBI1 match each other in this figure, since at that point the distance between the graph of IBI0 and the target is as large as the distance between two RDFs that do match each other. We see that for

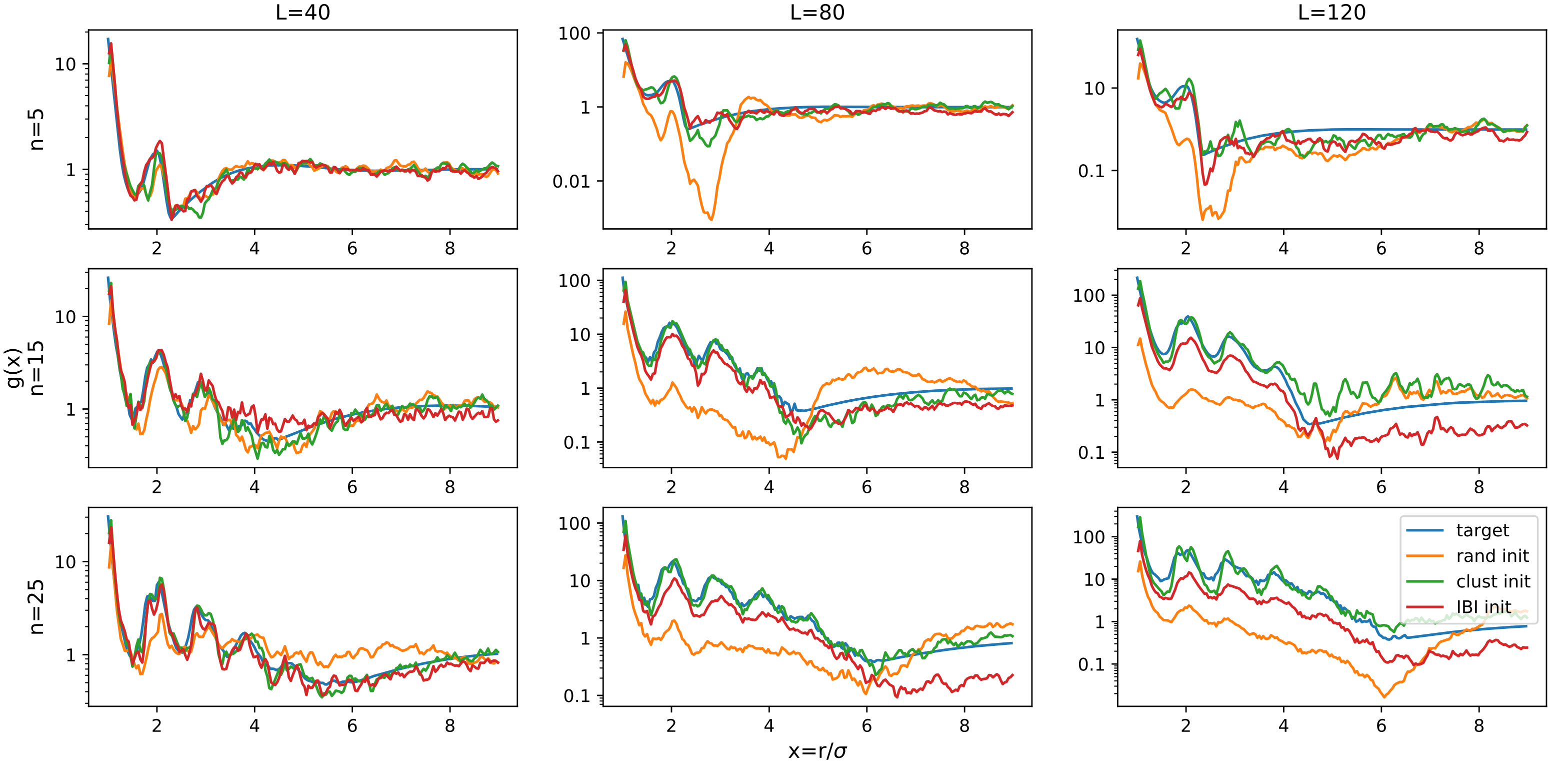


Figure 5: Radial distribution functions g as functions of x of the many-body potential (target), and the pair potential that resulted from Iterative Boltzmann Inversion that started with a zero potential, starting with all particles at random positions (rand init), particles in clusters (clust init) and the final configuration of the Iterative Boltzmann Inversion process (IBI init), for several values of the cluster size n and simulation box length L.

high densities the RDF quite quickly reaches the targeted RDF, for intermediate densities we could perhaps do with a few more IBI steps and for low densities the distance stays large and will probably never vanish.

With the pair potential we obtained from IBI0, we ran simulations for a few initial configurations and measured the RDF, as can be seen in figure 5. We also made some visualisations of the final configurations, as can be seen in the appendix. In all cases when the particles start in clusters the RDF matches the targeted RDF quite well, and the deviations that do exist can mostly be accounted for by the initial configuration. This means that the clusters persist under this pair potential, at least for as long as we simulated. When we start from the final position of the IBI in some cases the RDF stays too low, which, as indicated before, probably means not all particles are part of a cluster. In other cases the RDF does match up, which is quite remarkable, considering that the initial configuration of the IBI itself was completely random. When we start from a random configuration the RDF matches in very few cases and even then not very well, in most cases it is completely wrong.

We do the same for the resulting pair potential of IBI1, as can be seen in figure 6.

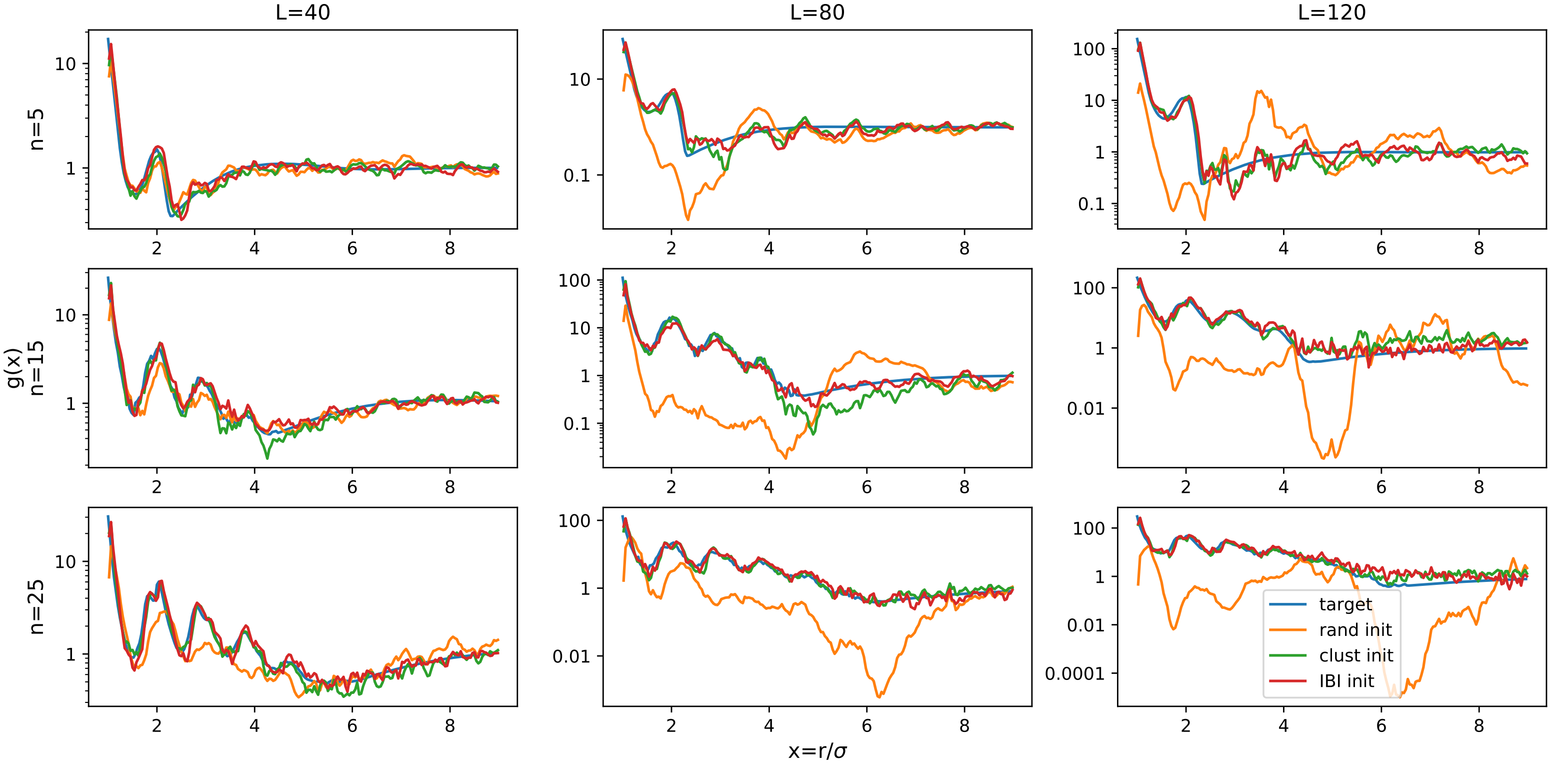


Figure 6: Radial distribution functions g as functions of x of the many-body potential (target), and the pair potential that resulted from Iterative Boltzmann Inversion that started with the nonzero potential, starting with all particles at random positions (rand init), particles in clusters (clust init) and the final configuration of the Iterative Boltzmann Inversion process (IBI init), for several values of the cluster size n and simulation box length L.

When we start in clusters or the final configuration of the IBI, the RDF matches that of the targeted RDF really well, meaning the clusters persist under this pair potential as well as under the previous one. Note that the IBI in this case itself started from clusters, thus it is perhaps unsurprising that the RDF matches in this case. When we start from a random configuration in almost all cases the RDF does not match the targeted RDF, and usually is even more spectacularly wrong than in IBI0.

To see how well the particles group in clusters of the right size, we classified the clusters and measured the frequency of each cluster size occurrence. To give a score of how well the particles are placed in clusters of the right size, we came up with

$$E_{clust} = \frac{1}{N} \sum_{i \text{ cluster}} n_i \cdot |n_i - n_{tgt}|$$

where we sum over the clusters, n_i is the size of cluster i and n_{tgt} is the targeted cluster size. This is equivalent to the average distance between the cluster size it is in and the targeted cluster size, averaged over the particles. A plot over time during the IBI can be seen in

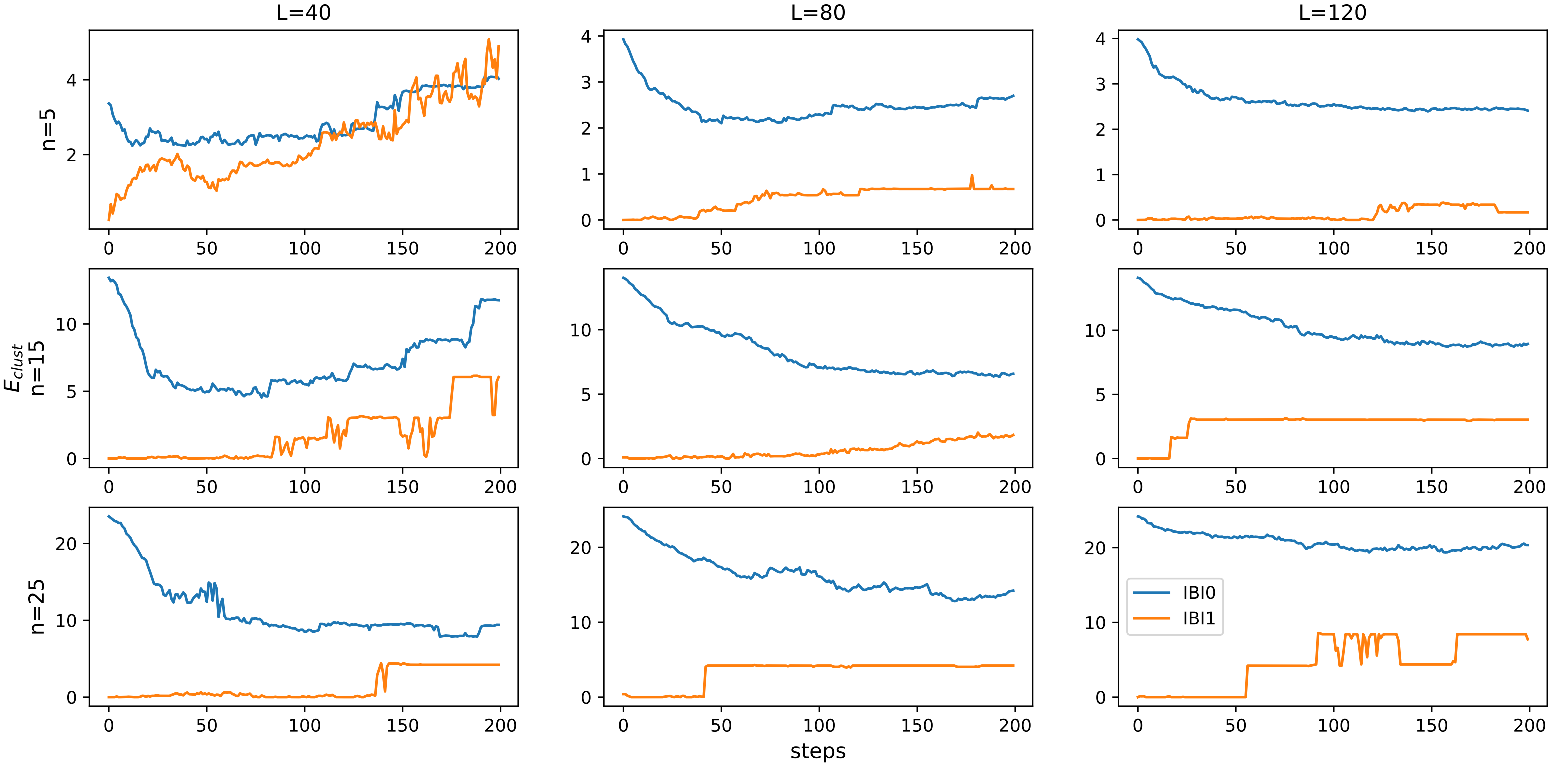


Figure 7: Average distance between the cluster size particles are in and the targeted cluster size during the steps of the Iterative Boltmann Inversion that started with a zero (IBI0) and nonzero (IBI1) pair potential, for several values of the cluster size n and simulation box length L.

figure 7. When we start the IBI in clusters of the right size, as in IBI1, this error starts at zero, and usually it stays quite low. In some cases we see a spontaneous jump up or down, this means a cluster has split up or two clusters have merged. Only for $n = 5$ and $L = 40$ the error increases dramatically, to a level above that of the error of a completely random configuration, this is because the clusters get too big. If we start from a random configuration and a zero pair potential, the error starts quite high, but quickly starts to decrease. However, the error stays quite high and in some cases even goes up again after a while, even in cases where the RDF starts to match the target RDF quite well. When the error goes up again the clusters formed probably get too large.

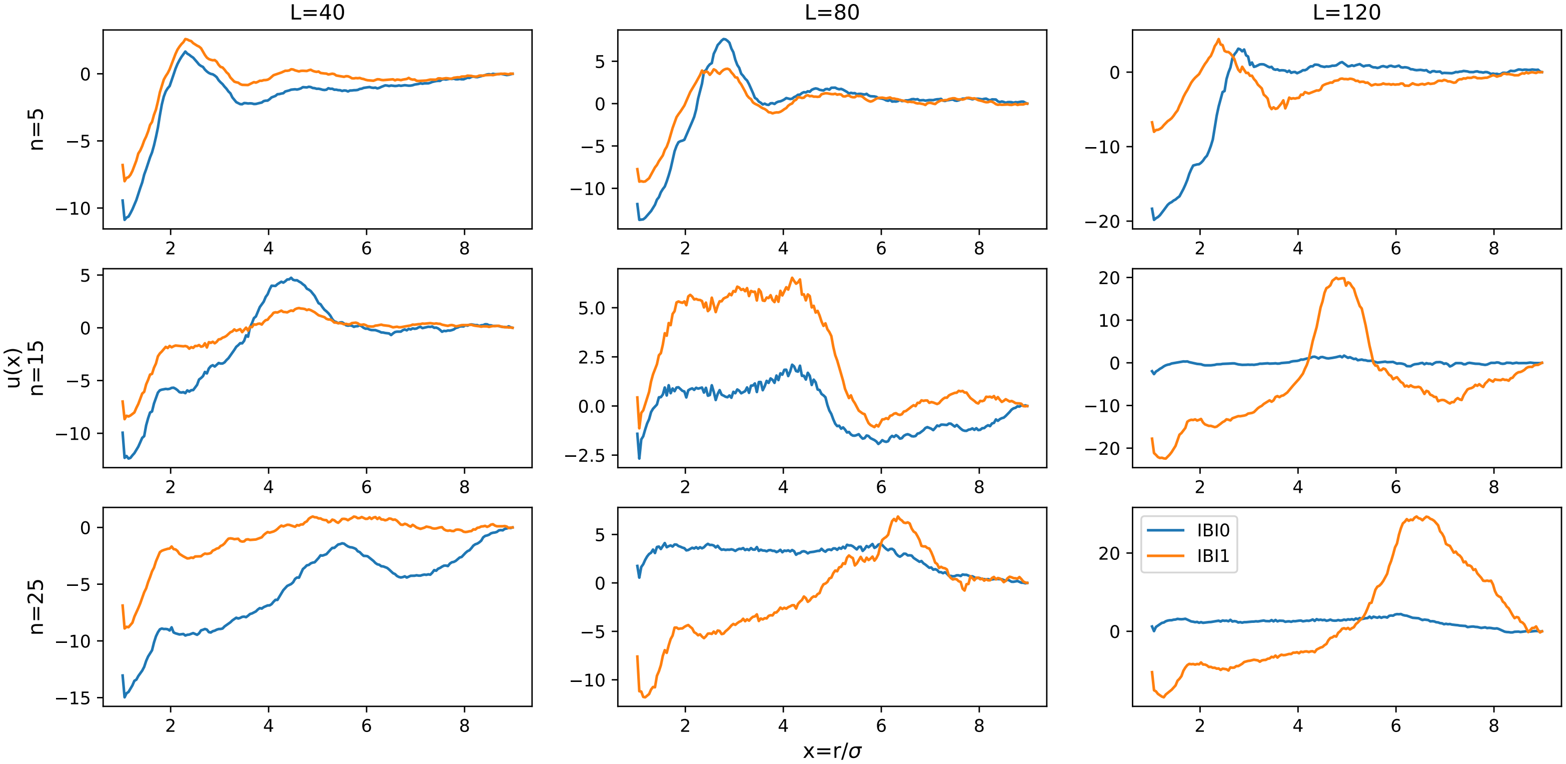


Figure 8: Pair potentials u as a function of x from the Iterative Boltzmann Inversion starting with a zero potential (IBI0) and a nonzero potential (IBI1) for several values of the cluster size n and simulation box length L.

In figure 8 we see the pair potentials that were obtained from IBI. In general at a short range the potential is negative and increasing, this means it is attractive, then hits a maximum, decreases in a small region, which means there is repulsion, and then stays more or less constant.

Some variations of this also occur, in some cases IBI1 has a much higher peak than IBI0, because of the starting potential of the IBI.

5 Conclusion

We created an artificial many-body potential to get the radial distribution function (RDF) of a typical cluster fluid, this graph turned out to look quite reasonable, having the shape we would expect and being smooth. We then used Iterative Boltzmann Inversion (IBI) to try to reproduce this RDF with a pair potential, starting from a zero potential in a random configuration (IBI0), and an inverse Boltzmann potential in clusters (IBI1). For IBI0, the RDF of the pair potential ended up matching the target quite well, although for low densities and large target clusters the RDF stayed structurally too low close to the particle, meaning some particles do not form clusters. For IBI1, the particles start in clusters and thus the RDF matches the target from the beginning, and this remains to be the case and no visible progress is made.

Using the final pair potential we find with IBI on an initial configuration of clusters, the RDF stays the same and keeps matching the target, meaning the clusters persist. If we use this pair potential on a random configuration, for high densities the RDF matches the target but not very well, but for lower densities gives a completely different RDF not resembling the target, with IBI0 being slightly better. One interesting result is that when using the final pair potential of IBI0 on the final configuration of IBI0, the RDF matches the target quite well, even though IBI0 started from a random configuration. When we use the pair potential of IBI0 directly on a random configuration the RDF is a lot worse than when we slowly evolve the pair potential while simulating.

In some cases we can find a pair potential that produces a RDF that matches the target quite well, but this does not mean clusters of the right size are produced, as our original goal was. For high densities and small target clusters in almost all cases the RDF matches really well, but the clusters are not the right size. Other situations are slightly better, but never get close to systems that were initialized in clusters.

6 Discussion and Outlook

In all cases we have set $x_{cut} = 9$. However for the larger cluster sizes, as we can see in figure 1, at this distance the RDF has not reached one, which is the limit for $x \to \infty$. It would be advisable to choose x_{cut} higher such that this is the case. This way, even without shifting the pair potential after every step there will be no discontinuity in the pair potential at this distance since the RDF will probably match the targeted RDF from the start at this distance. We would advice to use a x_{cut} of about two times the cluster diameter.

An important constraint of this research was computation time and computation power, which we probably should have tried to obtain more of, even though it is inherent to this kind of research that there is never enough of it. Looking back, the sampling during the steps of the IBI should have been longer, at least to confirm that it does not influence the results. With the current parameters there was a lot of noise in the RDFs, which probably has limited the accuracy with which we could optimize the pair potential. Perhaps with a more accurate pair potential and a more accurate match in the RDF the clusters would have formed better.

One constraint, for which more simulation time is only a partial solution, is the simulation of cluster fluids, and then specifically the movement of clusters. Remember that when we simulate the many-body potential, we make both particle moves and cluster moves. Later, during the IBI we cannot make cluster moves since the clusters are not well defined, we could classify the clusters like before and try to move those, but it would break the detailed balance condition since this way we could merge two clusters with a cluster move but we cannot split it the next step. Using single particle moves to move clusters is highly inefficient. In order to move a cluster, we would need to do so one by one. We would need to move the first half of the particles away from the majority of the particles before starting to move the particles towards the majority of the particles again, thus the energy threshold would be very high.

With single particle moves, the phase space of a cluster fluid cannot be explored properly within a reasonable time. If we look at the IBI starting in clusters, at the end of the IBI the clusters are still more or less in the same place. This means our research does not tell us anything about the interaction between clusters with confidence. That being said, our results of how the clusters themselves behave should still be valid. This constraint is probably the most important one to be addressed in future research. One idea is to pick a particle and a radius r, and move all the particles within this radius around the particle with the same displacement. But only if afterwards no new particles are within a distance of r from the new position of the particle. This should in theory satisfy the detailed balance condition, but we are not sure if this would work in practise.

One important parameter is α, but we have not given a very good argumentation for its current value. We suggest to try a few values and then pick the one that works best, the problem is that simulating the IBI is the most computationally expensive step of this research, and we cannot just try a lot of values for α. Perhaps its value should change during the IBI. We could try to measure if the RDF is moving in the right direction, and if so how quickly. If this speed is then too low we could increase α, and if we move in the wrong direction on some parts of the graph or perhaps too fast, we could decrease α.

Self assembly is the process of forming a certain material or structure from a random unordered configuration. This has implicitly been a part of this research, in our case this target structure was a cluster fluid. Starting from a random configuration, as we have seen when we continuously update the potential during the IBI we get a better result than when we use the same potential during the entire simulation. It should be noted that the former approach ended up having a lot more simulation steps, thus this can be one reason why the result is better. However, if we look at for example the energy during the latter simulation, it seems to almost have reached equilibrium, since not a lot of change is visible. It would therefore be very interesting to see if the longer simulation time really is the reason for the better result, or perhaps the approach of starting with a very small potential and increasing it during the simulation helps the process. This last approach resembles a technique called Simulated Annealing, like described in [4]. The difference being that we do not continue to zero temperature (infinite potential), and the decreasing of the temperature (increasing the potential) is a implicit result (almost a side effect) of the way IBI works.

When one uses IBI, one finds a pair potential with hopefully the right RDF. In practice however one still needs to physically create a particle with this pair potential, which can be a challenge. A more realistic scenario than being able to create an arbitrary potential is that one is able to modify a handful of parameters which have an influence on on the shape of the potential. To determine the value of these parameters one has too choose some sort of measure of how well the system behaves, in our case how well the clusters form, and then one can apply one out of numerous optimization schemes.

References

[1] Crocker, J. C., & Grier, D. G. (1994). *Microscopic measurement of the pair interaction potential of charge-stabilized colloid.* Physical Review Letters, 73(2), 352–355. https://doi.org/10.1103/physrevlett.73.352

[2] Frenkel, D., & Smit, B. (2012). *Understanding molecular simulation: from algorithms to applications.* Academic Press.

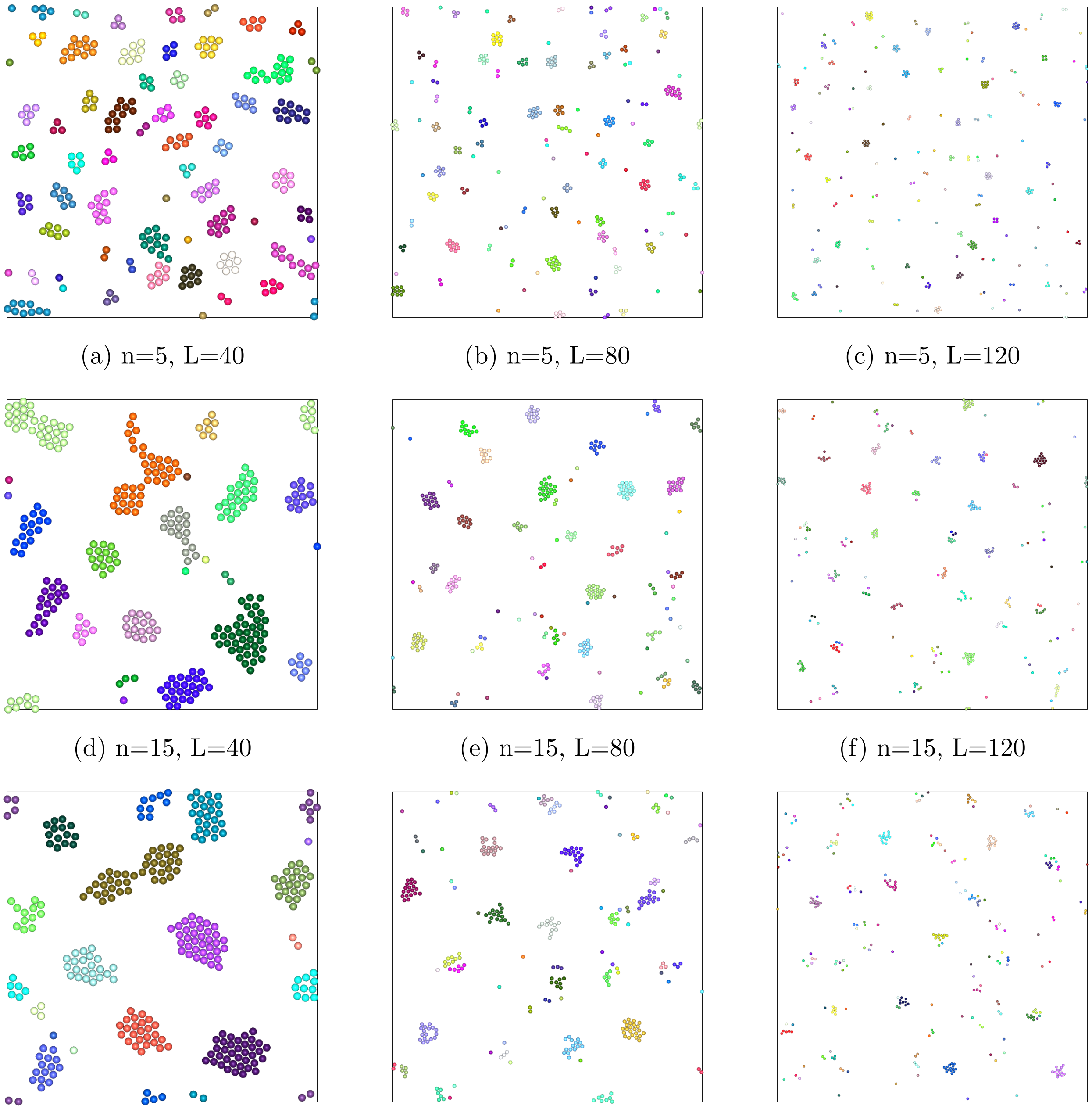
[3] Jadrich, R. B., Bollinger, J. A., Lindquist, B. A., & Truskett, T. M. (2015). *Equilibrium cluster fluids: pair interactions via inverse design.* Soft Matter, 11(48), 9342–9354. https://doi.org/10.1039/c5sm01832c

[4] A. Corana, M. Marchesi, C. Martini, & S. Ridella *Minimizing Multimodal Functions of Continuous Variables with the 'Simulated Annealing" Algorithm.* ACM Transactions on Mathematical Software, Vol. 13, No. 3, September 1987, Pages 262-280.

[5] Hermes, M. *Visualizing colloids.* https://michielhermes.github.io/viscol/.

Appendix A Snapshots

In this chapter we give some visualisations of a few snapshots of our simulation. We do this for the final configurations of the ”Last Step” simulations we described in the Approach. We also classified the clusters, as described in the Approach, and gave each cluster an unique color. All these visualisations, including the one on the cover of this thesis, were made using [5].



(a) n=5, L=40

(b) n=5, L=80

(c) n=5, L=120

(d) n=15, L=40

(e) n=15, L=80

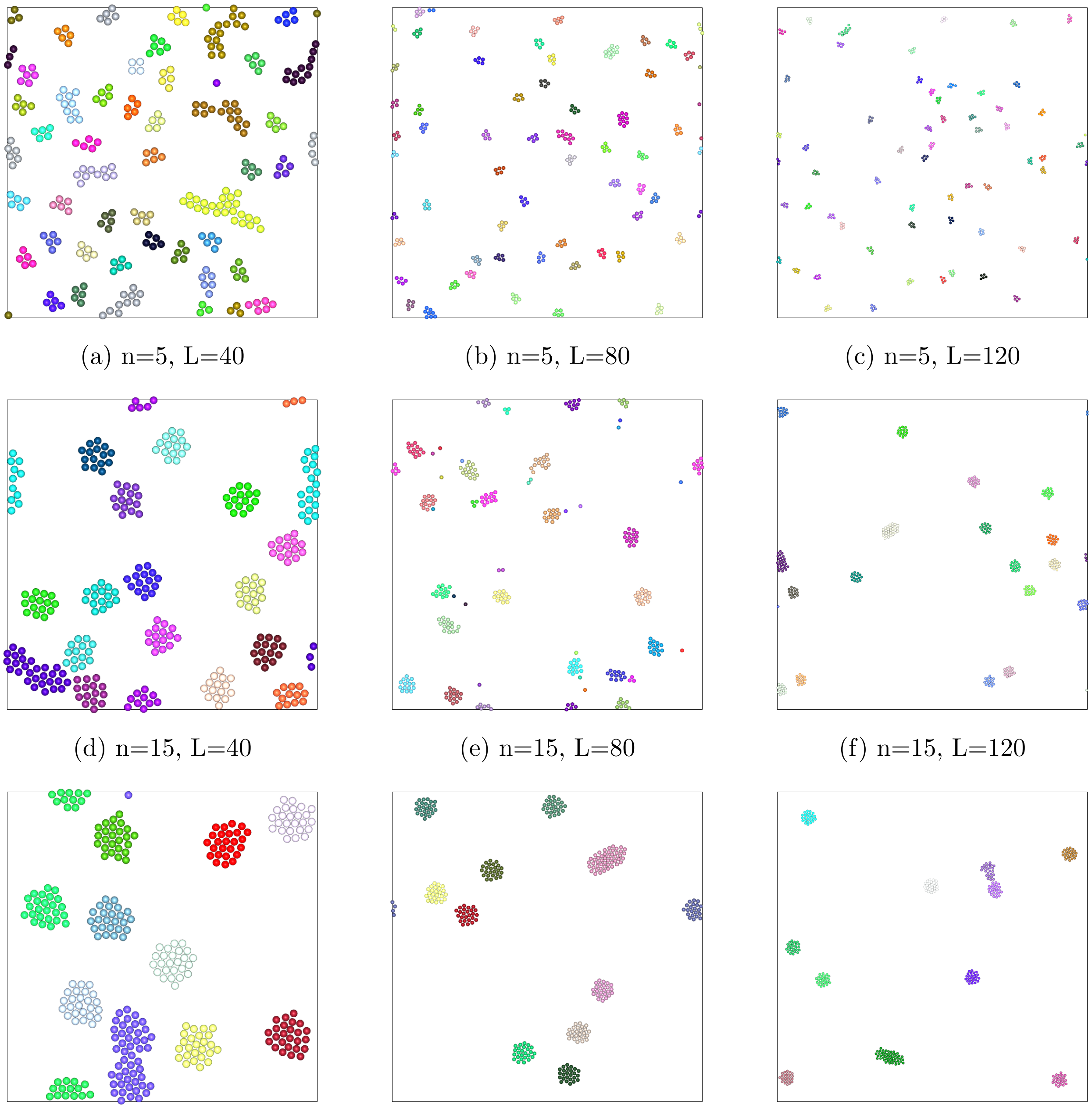
(f) n=15, L=120

(g) n=25, L=40

(h) n=25, L=80

(i) n=25, L=120

Figure 9: Visualisations of the final configuration after simulating with the pair potential that resulted from Iterative Boltzmann Inversion that started with a zero potential, starting in the final configuration of the Iterative Boltzmann Inversion process for several values of the cluster size n and simulation box length L. (IBI0 - IBI)


(a) n=5, L=40

(b) n=5, L=80

(c) n=5, L=120

(d) n=15, L=40

(e) n=15, L=80

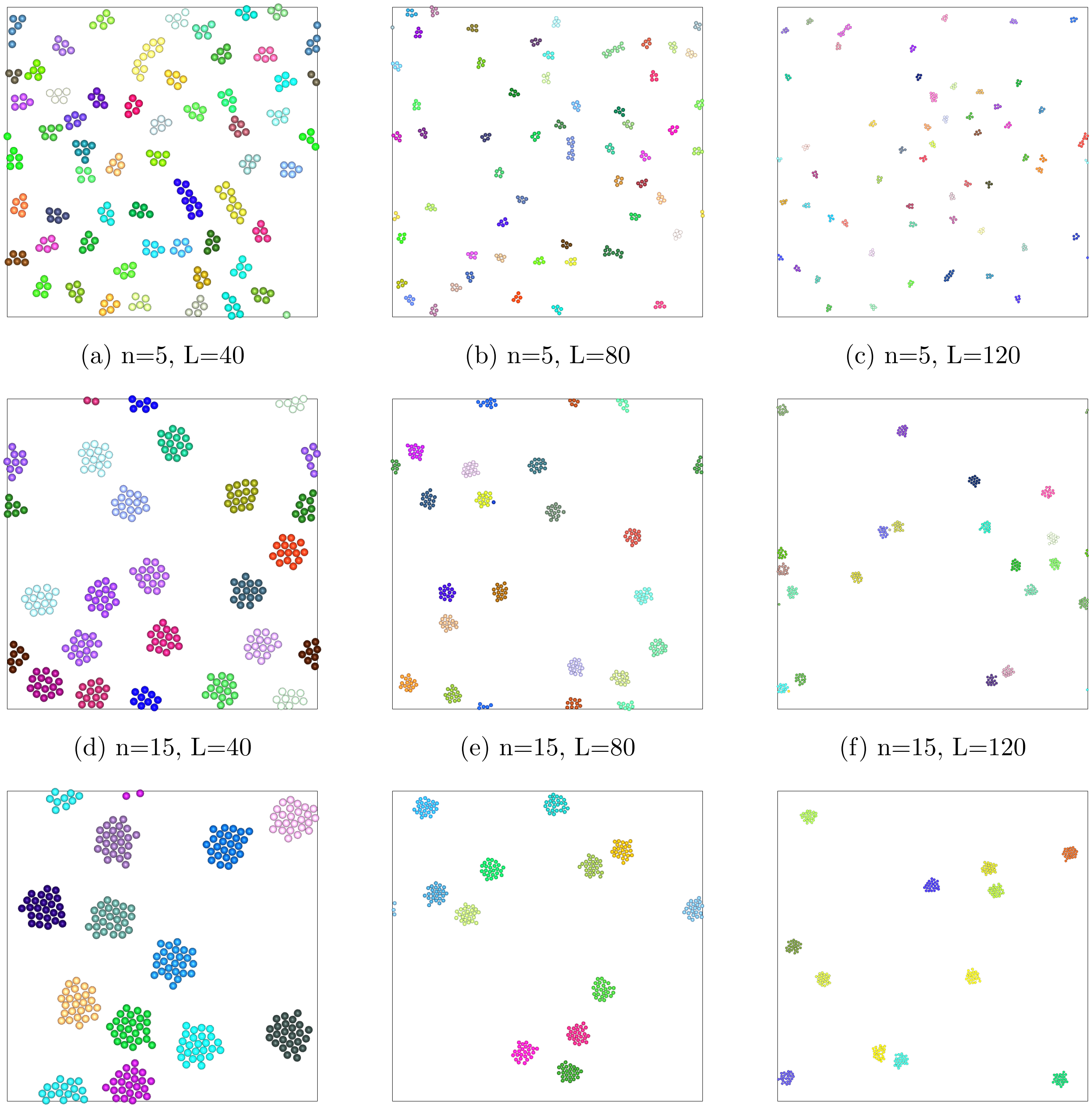
(f) n=15, L=120

(g) n=25, L=40

(h) n=25, L=80

(i) n=25, L=120

Figure 10: Visualisations of the final configuration after simulating with the pair potential that resulted from Iterative Boltzmann Inversion that started with the nonzero potential, starting in the final configuration of the Iterative Boltzmann Inversion process for several values of the cluster size n and simulation box length L. (IBI1 - IBI)



(a) n=5, L=40

(b) n=5, L=80

(c) n=5, L=120

(d) n=15, L=40

(e) n=15, L=80

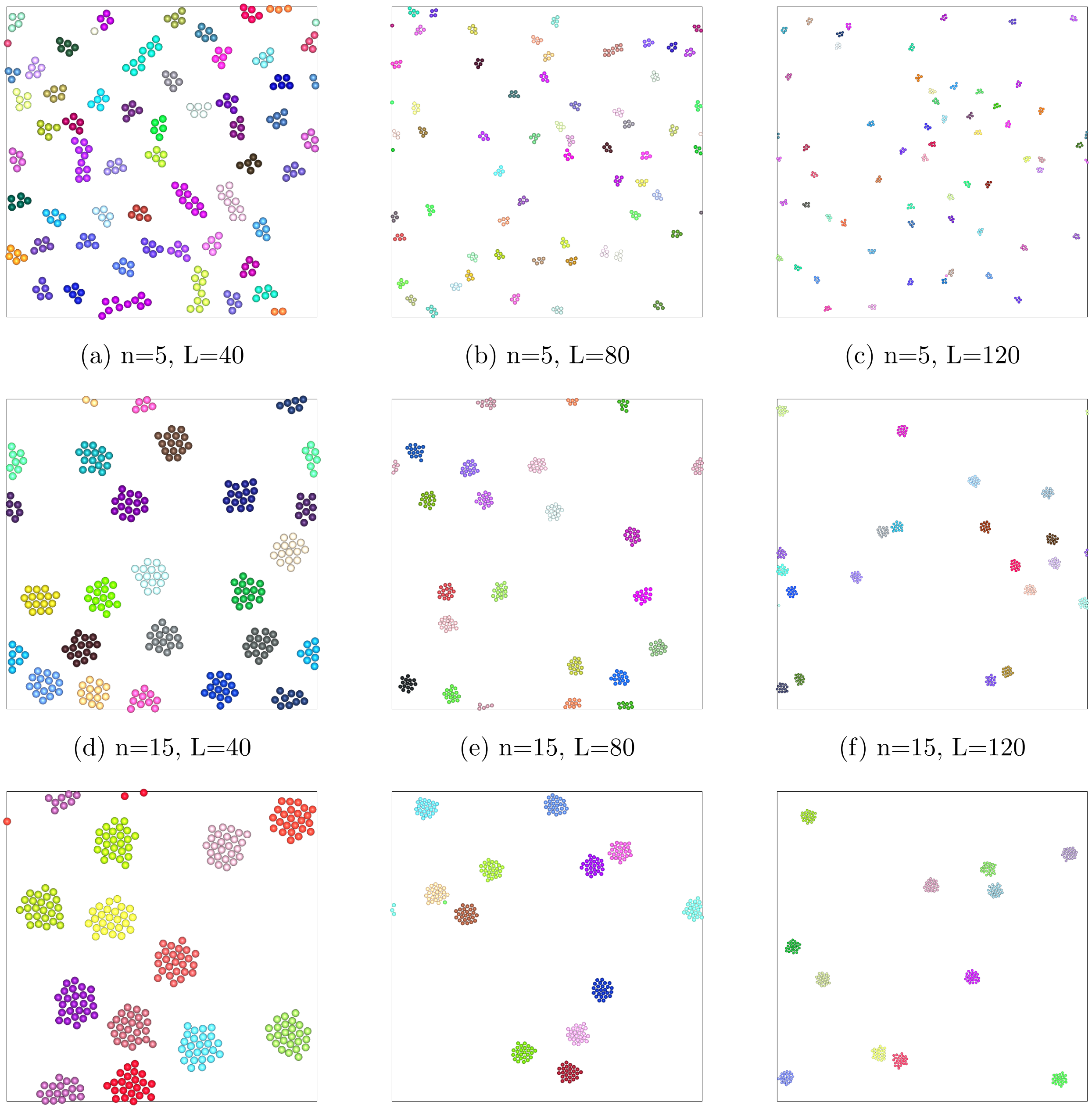
(f) n=15, L=120

(g) n=25, L=40

(h) n=25, L=80

(i) n=25, L=120

Figure 11: Visualisations of the final configuration after simulating with the pair potential that resulted from Iterative Boltzmann Inversion that started with a zero potential, starting in clusters for several values of the cluster size n and simulation box length L. (IBI0 - clust)



(a) n=5, L=40

(b) n=5, L=80

(c) n=5, L=120

(d) n=15, L=40

(e) n=15, L=80

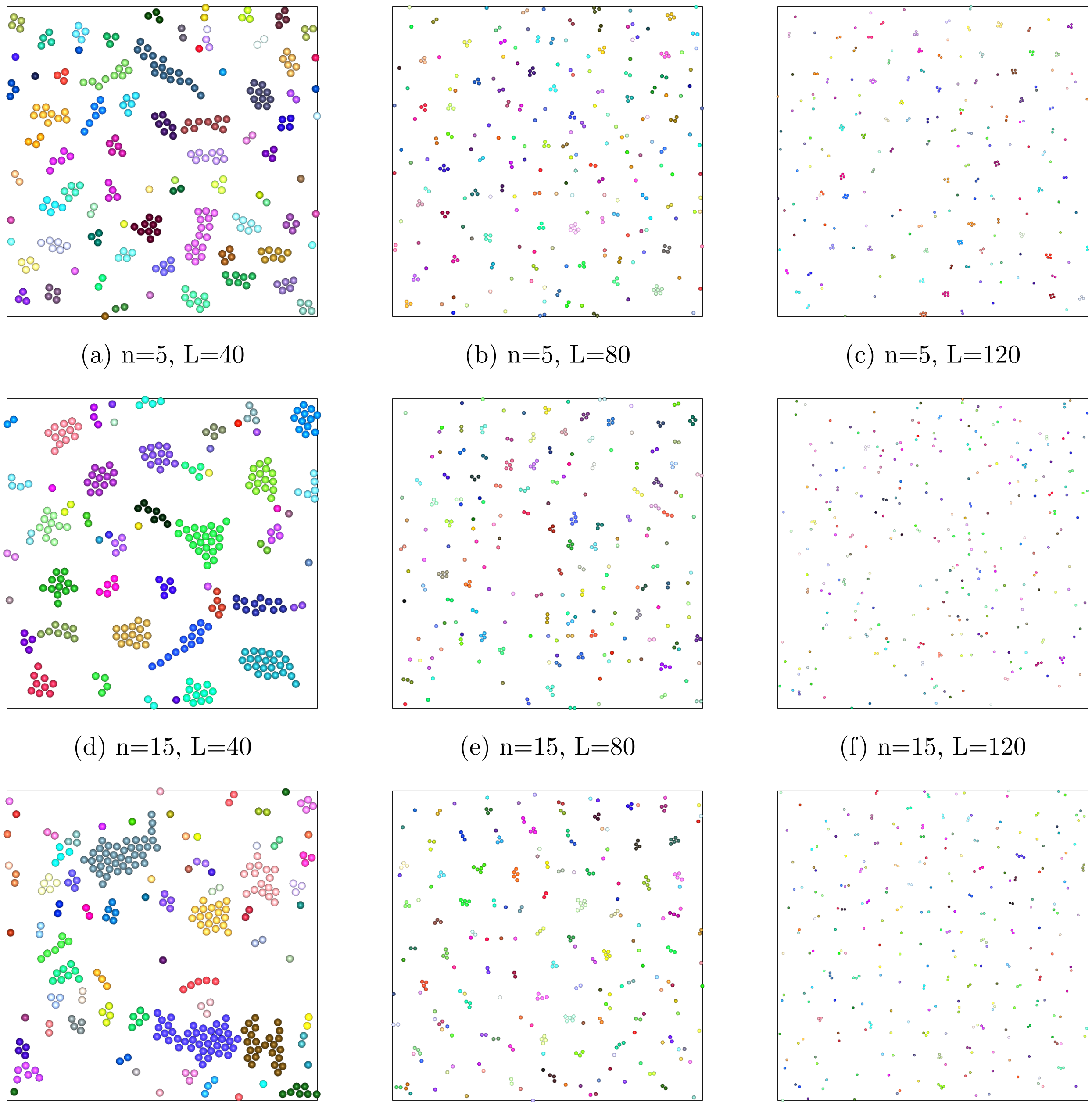
(f) n=15, L=120

(g) n=25, L=40

(h) n=25, L=80

(i) n=25, L=120

Figure 12: Visualisations of the final configuration after simulating with the pair potential that resulted from Iterative Boltzmann Inversion that started with the nonzero potential, starting in clusters for several values of the cluster size n and simulation box length L. (IBI1 - clust)



(a) n=5, L=40

(b) n=5, L=80

(c) n=5, L=120

(d) n=15, L=40

(e) n=15, L=80

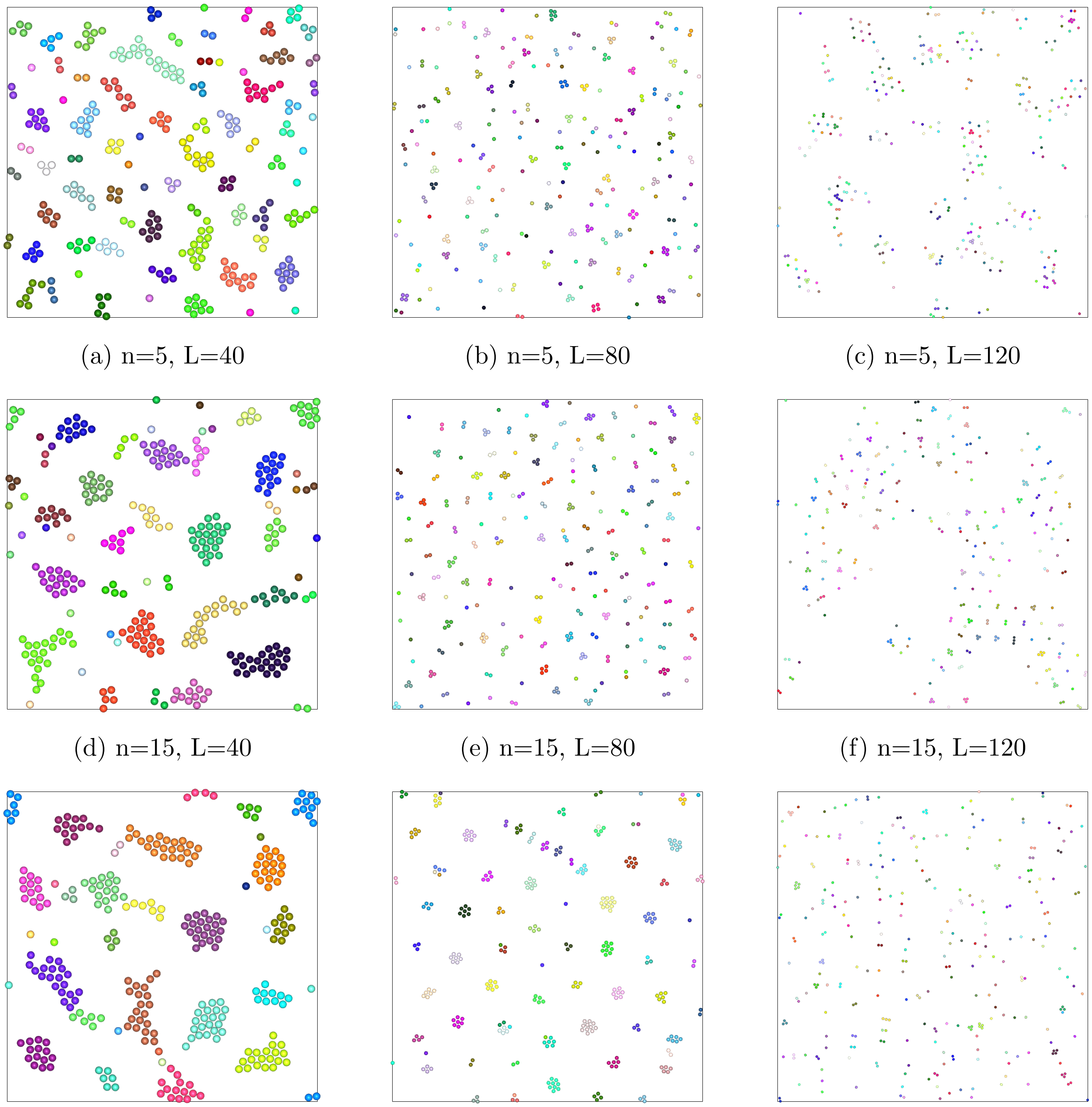
(f) n=15, L=120

(g) n=25, L=40

(h) n=25, L=80

(i) n=25, L=120

Figure 13: Visualisations of the final configuration after simulating with the pair potential that resulted from Iterative Boltzmann Inversion that started with a zero potential, starting in a random configuration for several values of the cluster size n and simulation box length L. (IBI0 - rand)



(a) n=5, L=40

(b) n=5, L=80

(c) n=5, L=120

(d) n=15, L=40

(e) n=15, L=80

(f) n=15, L=120

(g) n=25, L=40

(h) n=25, L=80

(i) n=25, L=120

Figure 14: Visualisations of the final configuration after simulating with the pair potential that resulted from Iterative Boltzmann Inversion that started with the nonzero potential, starting in a random configuration for several values of the cluster size n and simulation box length L. (IBI1 - rand)