# Monte Carlo Simulations of Liquid Crystal Decorated Nanoparticles

Raz Rivlis

Dept. of Physics, Brown University, Providence, RI, 02912, USA

May 4, 2017

**Abstract**

Nanoparticles are a class of particle between 1 and 100 nm large. They have a number of unique and potentially useful properties. There has been interest in trying to control the alignment of large numbers of these particles. One method that is being explored involves attaching liquid crystal mesogens to the surface of the nanoparticles, and using the self aligning tendencies of liquid crystals to cause the nanoparticles to align. Currently, these systems are poorly understood. This project aims to model these liquid crystal decorated nanoparticles to better understand methods of controlling the alignment of the nanoparticles. A small number of these arrangements have been modeled previously by Silvia Orlandi and Claudio Zannoni [2013]. They modeled two systems, both with the same number of nanoparticles and liquid crystals, with the same particle parameters. The difference between the two systems were the attachment angles of the mesogens. The primary goal of this project is to expand this work to more general forms, and study the effects of different types of mesogens, differing numbers of attached mesogens, and the effects of imperfections in the attachments of particles. This project is still in progress, and this report is a documentation of the steps taken until now.

# 1 Introduction

Nanoparticles are a special class of material that consist of a small cluster of atoms, usually between 1 and 100 nm in size. This size is large enough for nanoparticles to exhibit properties of their bulk material, while being small enough to interact with atomic and molecular structures. The large majority of the interesting properties of nanoparticles are due to their extremely high surface area to volume ratio. They are also frequently small enough to confine their electrons and produce quantum mechanical effects. For example, many nanoparticles fluoresce under UV light, with a resulting light frequency dependent on the size of the particle. Lately there has been much interest in finding new ways to control the organization of nanoparticles. Controlling the orientation and position of these particles could allow for new uses in electronic devices, or medical advances, to name only a few potential applications. One potential method of creating order is using the thermodynamic properties of the system to cause the nanoparticles to self align (Marek Grzelczak et. al. [2010]).

Following the work of Silvia Orlandi and Claudio Zannoni [2013], Xiaobin Mang et. al. [2012] and Wiktor Lewandowski et. al. [2013], we are attempting to take advantage of liquid crystals to induce self assembly of nanoparticles. Liquid crystals are a type of material that flows, like a liquid, but the molecules tend to align to form long range order. More details of liquid crystals and nanoparticles will be discussed later. By attaching liquid crystal molecules, called mesogens, to the surface of nanoparticles, we hope that the self alignment properties of liquid crystals will cause the nanoparticles to align as well.

There has been a small amount experimentation with this method of self assembly in the past. Xiaobin Mang et. al. [2012] have experimented with attaching various mesogenic ligands and alkylthiols (co-ligands) to gold nanoparticles. They found this caused the gold nanoparticles to exhibit signs of long range order, and form different structures dependent on the number of attached mesogens. Using various analysis methods, including x-ray diffraction, they were able to determine the different structure types and uniformity, dependent on the number of attached mesogens, and the presence or absence of the co-ligands. This study gives very useful information about some of the observable effects of these materials; however, attempting to create many such materials and study them in depth is time consuming, and difficult. We intend to computationally model these arrangements of particles to better understand what conditions lead to successful self assembly. These models will hopefully show some insight as how to control the packing structures of the nanoparticles.

There has been some prior modeling of liquid crystal decorated nanoparticles by Silvia Orlandi and Claudio Zannoni [2013]. As far as we are aware, this is the

only other work discussing the modeling of these systems. Their study modeled two specific cases, and each one demonstrated different packing structures for the nanoparticles. We intend to take a closer look at some of their results, and better understand how to control the specific parameters of the nanoparticle structures formed.

# 2 Background

## 2.1 Liquid Crystals

When we consider matter, we usually think about the three most common phases, gases, liquids, and solids. If we think about the defining characteristics of these phases, we think of gases as a collection of widely spaced, quickly moving particles, that rarely interact with each other, and expand to fill their container. Liquids on the other hand, are far denser, allowing their particles to frequently interact. However, they still randomly flow around each other. Furthermore, liquids have a fairly uniform density under varying pressures. Since the particles in liquids flow randomly, there is no long range structure to a liquid. Solids are rigid. Solids have particles fixed in relation to the neighboring particles, which means there is local positional order to the particles. Crystalline solids also have a high degree of long range positional and orientational order. This means that the positions of the particles occur in some kind of predictable pattern.

Liquid crystals are a phase of matter that share characteristics of both liquids and crystals. They have little local positional order. Like liquids, the particles flow around each other semi-randomly; however, they have long range orientational order, like crystalline solids. The particles in liquid crystals are usually shaped like rods or discs, and they must have an anisotropic shape for orientational order to have meaning. There are also many different phases of liquid crystals, described by what form of long range order they have. The simplest form is known as the nematic phase, as shown in figure 1. In this phase, the rod-like particles all align in the same direction, but can still flow around each other.

Two more simple liquid crystal phases are the smectic A and smectic C liquid crystal phases. In these cases, the rod-like particles again align in one direction, only this time, they appear in rows, as shown in figure 2. In these phases, the mesogens are free flowing within each layer. In the smectic A phase the alignment of the mesogens is perpendicular to the layers, while in the smectic C phase, the mesogens align at some other angle. Of note, is that many liquid crystals can exist in more than one

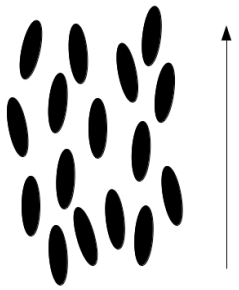

Figure 1: Schematic of a nematic phase liquid crystal. The arrow represents the alignment direction of the mesogens.

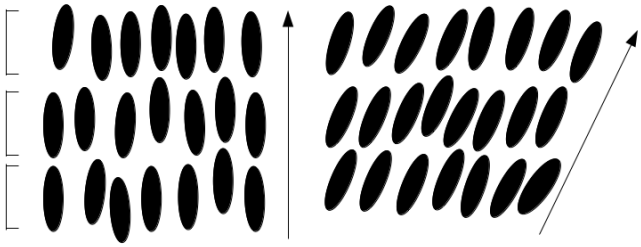

Figure 2: Schematic of smectic A (left) and C (right) liquid crystals. The arrows show the alignment direction for the two phases. Note both phases have the layers propagate in the direction of the central arrow. The brackets highlight the individual layers.

phase, depending on the temperature or pressure (or other stimuli) (Jeffery Billeter and Robert Pelcovits. [1998]). A common example includes rod-like particles in a liquid, or isotropic, phase at high temperatures, that transition into a nematic phase as the temperature decreases. As the material cools off further, a smectic A phase, and then a smectic B phase might be observed, before the material becomes a solid. The smectic B phase is similar to the smectic A phase, but there is more positional order within each layer.

Not all liquid crystals can exist in all of these phases, and some might have more exotic phases, such as the smectic C phase, discotic phases, where the particles are disc-like instead of rod-like, and chiral smectic phases. In chiral phases, the orientation of the rod-like particles twists in a uniform way around a certain direction.

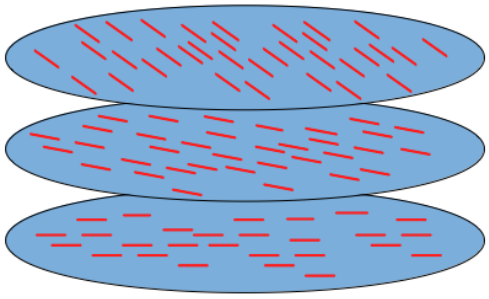

Figure 3: Schematic of a chiral liquid crystal. The blue circles represent specific sections of the fluid at different locations, and the red lines show the orientations of the mesogens at these locations. There are more mesogens between the circles. Note that the alignment of the mesogens changes depending on their location along a specific axis.

(Figure 3).

## 2.2 Nanoparticles

The properties of nanoparticles can vary immensely depending on the materials they are made of, their size, and the environment they're in. The wide range of properties is part of what makes them so potentially useful. Instead of attempting to give a comprehensive description of nanoparticles, we will describe a few of their unique properties, and some of their current applications.

Since nanoparticles have sizes on the same scale as the wavelength of light, they frequently exhibit surprising optical properties. The simplest example is that the color of a nanoparticle solution is frequently very different from the color of the bulk material. Furthermore, nanoparticles are capable of absorbing far more incoming electromagnetic radiation than sheets of the bulk material. This gives them many potential applications in creating better solar cells, or light filters (Taylor et. al. [2012]). Because of this property, Zinc Oxide nanoparticles are sometimes used in sunscreen to increase its UV blocking capabilities.

Nanoparticles have been finding increased uses in biological processes. Due to their high surface area, when various organic molecules are attached to the surface, they can have a large impact on a variety of processes including gene regulation and enzyme inhibition. However, since they are substantially larger than the molecules themselves, the retention time is much larger for the coated nanoparticle, which can

help the treatment of long term illnesses. Nanoparticles can also be used to help target specific organs or areas and reduce the necessary dosage of the medication. (Maria E. Akerman et. al. [2002])

## 2.3 Simulation Methods

To begin simulations of liquid crystals, we must first come up with some kind of model to approximate the particles as well as possible. To model a simple liquid, we could attempt to simulate each particle as a sphere, that generates some kind of potential field around it. In this case, the potential would be radially symmetric. Another way to say this is that the force between two nearby particles, considered in isolation, is dependent only on the distance between the two particles. A well established potential, used for these simple spherical models is known as the Lennard-Jones potential (Lennard-Jones. [1924]):

$$U = 4\epsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right] \tag{1}$$

where U is the potential between two particles, $\epsilon$ is the well depth, $\sigma$ is the distance where the potential is zero, and r is the distance between particles. In this potential, the $r^{-12}$ term describes the repulsive forces due to electron overlap, while the $r^{-6}$ term describes the attractive van der Waals forces. The Lennard-Jones potential is a fairly good approximation, and is computationally simple. Notably, the potential is accurate enough to accurately model phase transitions, as demonstrated by Jean-Pierre Hanson and Loup Verlet. [1969].

To model liquid crystals, we need a more complex potential, as liquid crystal molecules are not spherically symmetric. To model a cylindrically symmetric molecule (symmetric around only once axis), we use a modification of the Lennard-Jones potential, called the Gay-Berne potential (J. G. Gay and B. J. Berne. [1981]):

$$U(\hat{u_i}, \hat{u_j}, \vec{r}) = 4\epsilon(\hat{u_i}, \hat{u_j}, \hat{r}) \times \left[\left(\frac{\sigma_o}{r - \sigma(\hat{u_i}, \hat{u_j}, \hat{r}) + \sigma_o}\right)^{12} - \left(\frac{\sigma_o}{r - \sigma(\hat{u_i}, \hat{u_j}, \hat{r}) + \sigma_o}\right)^{6}\right] \tag{2}$$

where $\vec{r}$ is the vector between the two particles, $r = |r|$, $\hat{r} = \vec{r}/r$, $\sigma_o$ sets a length scale for the potential, $\hat{u_i}$ and $\hat{u_j}$ are unit vectors representing the orientation of the two particles, $\sigma(\hat{u_i}, \hat{u_j}, \hat{r})$ is the intermolecular separation where the potential vanishes, and is given by:

$$\sigma(\hat{u_i}, \hat{u_j}, \hat{r}) = \sigma_o \left[ 1 - \frac{1}{2}\chi_\sigma \left( \frac{(\hat{r} \cdot \hat{u_i} + \hat{r} \cdot \hat{u_j})^2}{1 + \chi_\sigma \hat{u_i} \cdot \hat{u_j}} + \frac{(\hat{r} \cdot \hat{u_i} - \hat{r} \cdot \hat{u_j})^2}{1 - \chi_\sigma \hat{u_i} \cdot \hat{u_j}} \right) \right]^{-1/2} \tag{3}$$

where

$$\chi_\sigma = \frac{\sigma_e/\sigma_s - 1}{\sigma_e/\sigma_s + 1} \tag{4}$$

$\sigma_e$ is the zero potential spacing between two particles arranged end to end, and $\sigma_e$ is the spacing when they are side to side. Note, that when $\sigma_e = \sigma_s$ the equation reduces to the standard Lennard-Jones model. These formulas for $\sigma(\hat{u_i}, \hat{u_j}, \hat{r})$ come from calculating the overlap integral of two multi-dimensional Gaussians, with the parameters of the Gaussian depending on the specific particles.

$\epsilon(\hat{u_i}, \hat{u_j}, \hat{r})$ represents the well depth along various axis and is given by:

$$\epsilon(\hat{u_i}, \hat{u_j}, \hat{r}) = \epsilon_o \epsilon^\nu(\hat{u_i}, \hat{u_j}) g_\epsilon^\mu(\hat{u_i}, \hat{u_j}, \hat{r}) \tag{5}$$

where

$$\epsilon(\hat{u_i}, \hat{u_j}) = [1 - \chi_\sigma^2 (\hat{u_i} \cdot \hat{u_j})^2]^{-1/2} \tag{6}$$

and

$$g_\epsilon(\hat{u_i}, \hat{u_j}, \hat{r}) = 1 - \frac{1}{2}\chi_\epsilon \left( \frac{(\hat{r} \cdot \hat{u_i} + \hat{r} \cdot \hat{u_j})^2}{1 + \chi_\epsilon \hat{u_i} \cdot \hat{u_j}} + \frac{(\hat{r} \cdot \hat{u_i} - \hat{r} \cdot \hat{u_j})^2}{1 - \chi_\epsilon \hat{u_i} \cdot \hat{u_j}} \right) \tag{7}$$

with $\chi_\epsilon$ defined by:

$$\chi_\epsilon = \frac{1 - (\epsilon_e/\epsilon_s)^{1/\mu}}{1 + (\epsilon_e/\epsilon_s)^{1/\mu}} \tag{8}$$

where $\epsilon_e$ is the end to end well depth, and $\epsilon_s$ is the side by side well depth. The introduction of this $g_\epsilon$ term allows the control of the well depth along each axis and orientation. All parameters are adjustable. Some commonly used parameters that have been shown to produce nematic and smectic A phases are $\mu = 2$, $\nu = 1$, $\sigma_e/\sigma_s = 3$, and $\epsilon_s/\epsilon_e = 5$ (Jeffery Billeter and Robert Pelcovits. [1998]). This implies that this potential is a reasonable potential to use for the type of interactions we are interested in. Unfortunately, this potential is still not quite general enough, as we need to model the interactions between spherically symmetric nanoparticles, and uniaxially symmetric mesogens. We used a generalized form of the Gay-Berne potential, developed by Douglas J. Cleaver et. al. [1996]:

$$U = 4\epsilon(\vec{r}, \hat{u_i}, \hat{u_j}) \times \left[ \left( \frac{\sigma_c}{\vec{r} - \sigma(\hat{r}, \hat{u_i}, \hat{u_j}) + \sigma_c} \right)^{12} - \left( \frac{\sigma_c}{\vec{r} - \sigma(\hat{r}, \hat{u_i}, \hat{u_j}) + \sigma_c} \right)^{6} \right] \tag{9}$$

where $\vec{r}$ is the vector between the two particles, $\hat{r}$ is the unit vector in the direction of $\vec{r}$, and $u_i$ and $u_j$ are unit vectors specifying the particles orientation, $\epsilon(\vec{r}, \hat{u_i}, \hat{u_j})$ represents a well depth parameter, $\sigma(\hat{r}, \hat{u_1}, \hat{u_2})$ is a separation term, and $\sigma_c$ is the minimum contact distance. In the case of two different particles $\sigma_c = \sigma_o \sqrt{\frac{\sigma_{ci}^2}{2} + \frac{\sigma_{cj}^2}{2}}$ where $\sigma_o$ is the length scale, and $\sigma_{ci}$ and $\sigma_{cj}$ are the minimum contact distances for each particle. $\sigma(\hat{r}, \hat{u_1}, \hat{u_2})$ is given by

$$\sigma(\hat{r}, \hat{u_1}, \hat{u_2}) = \sigma_o \left[ 1 - \chi \left( \frac{\alpha^2 (\hat{r} \cdot \hat{u_i})^2 + \alpha^{-2} (\hat{r} \cdot \hat{u_j})^2 - 2\chi (\hat{r} \cdot \hat{u_i})(\hat{r} \cdot \hat{u_j})(\hat{u_i} \cdot \hat{u_j})}{1 - \chi^2 (\hat{u_i} \cdot \hat{u_j})^2} \right) \right]^{-1/2} \tag{10}$$

where $\chi$ is:

$$\chi = \left( \frac{(\sigma_{ei}^2 - \sigma_{si}^2)(\sigma_{ej}^2 - \sigma_{sj}^2)}{(\sigma_{ej}^2 + \sigma_{si}^2)(\sigma_{ei}^2 + \sigma_{sj}^2)} \right)^{1/2} \tag{11}$$

and $\alpha^2$ is:

$$\alpha^2 = \left( \frac{(\sigma_{ei}^2 - \sigma_{si}^2)(\sigma_{ej}^2 + \sigma_{sj}^2)}{(\sigma_{ej}^2 - \sigma_{si}^2)(\sigma_{ei}^2 + \sigma_{sj}^2)} \right)^{1/2} \tag{12}$$

where $\sigma_{ei}$ and $\sigma_{ej}$ are the end to end separations for each particle, and $\sigma_{si}$ and $\sigma_{sj}$ are the side to side separations.

$\epsilon(\vec{r}, \hat{u_i}, \hat{u_j})$ is given by

$$\epsilon(\vec{r}, \hat{u_i}, \hat{u_j}) = \epsilon_o \epsilon^{\nu}(\hat{u_i}, \hat{u_j}) g_{\epsilon}^{\mu}(\vec{r}, \hat{u_i}, \hat{u_j}) \tag{13}$$

where $\epsilon_o$ is a constant used to scale the energy, and

$$\epsilon(\hat{u_i}, \hat{u_j}) = [1 - \chi^2 (\hat{u_i} \cdot \hat{u_j})^2]^{-1/2} \tag{14}$$

where $\chi$ is from equation 11.

$$g_{\epsilon}(\hat{r}, \hat{u_1}, \hat{u_2}) = \left[ 1 - \chi' \left( \frac{\alpha'^2 (\hat{r} \cdot \hat{u_i})^2 + \alpha'^{-2} (\hat{r} \cdot \hat{u_j})^2 - 2\chi' (\hat{r} \cdot \hat{u_i})(\hat{r} \cdot \hat{u_j})(\hat{u_i} \cdot \hat{u_j})}{1 - \chi'^2 (\hat{u_i} \cdot \hat{u_j})^2} \right) \right] \tag{15}$$

where $\chi'$ is:

$$\chi' = \left( \frac{(\epsilon_{ei}^{1/\mu} - \epsilon_{si}^{1/\mu})(\epsilon_{ej}^{1/\mu} - \epsilon_{sj}^{1/\mu})}{(\epsilon_{ej}^{1/\mu} + \epsilon_{si}^{1/\mu})(\epsilon_{ei}^{1/\mu} + \epsilon_{sj}^{1/\mu})} \right)^{1/2} \tag{16}$$

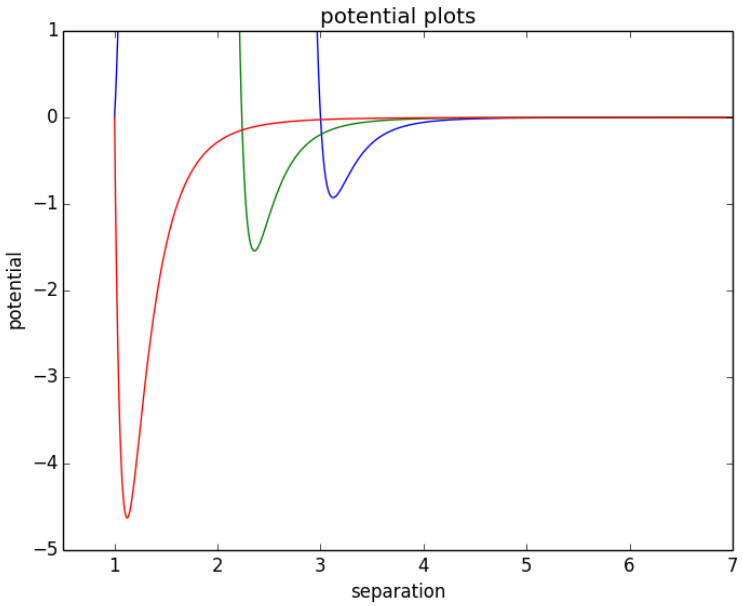


Figure 4: Plots showing the interaction potentials for two mesogens. The potential is in units of $\epsilon_o$ and the separation is in units of $\sigma_o$. The red line is for particles oriented side by side, green is end to side orientation, and blue is end to end orientation. The positive sloped blue line near 1 $\sigma_o$ is an artifact of the potential function used, but is separated from the potentials of interest by an infinite potential barrier, so is never reached in the simulation.

and $\alpha'^2$ is:

$$\alpha'^2 = \left( \frac{(\epsilon_{ei}^{1/\mu} - \epsilon_{si}^{1/\mu})(\epsilon_{ej}^{1/\mu} + \epsilon_{sj}^{1/\mu})}{(\epsilon_{ej}^{1/\mu} - \epsilon_{si}^{1/\mu})(\epsilon_{ei}^{1/\mu} + \epsilon_{sj}^{1/\mu})} \right)^{1/2} \tag{17}$$

where $\epsilon_{ei}$ and $\epsilon_{ej}$ are the end to end separations for each particle, and $\epsilon_{si}$ and $\epsilon_{sj}$ are the side to side separations. Some plots of these potentials are shown in figure 4 and 5.

Now that we have described some possible potentials used, we need a method of progressing a simulation of particles. To start, we only want to simulate a small number of particles, as modeling all the molecules in a liquid is impossible. To accomplish this we use periodic boundary conditions. In these conditions, all of the particles are confined to a box, but when a particle crosses a boundary, it is added

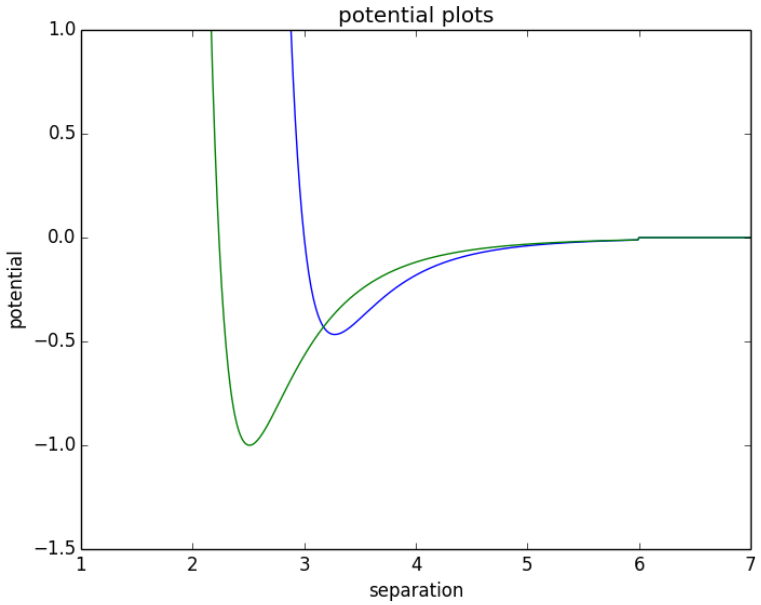


Figure 5: Plots showing the interaction potentials for a mesogen and a nanoparticle. The potential is in units of $\epsilon_o$ and the separation is in units of $\sigma_o$. The green line is when the mesogen is tangentally oriented and the blue line is when the mesogen is radially oriented.

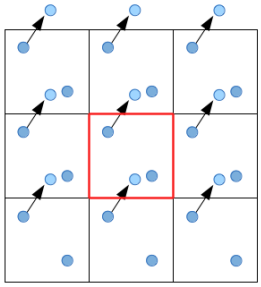

Figure 6: Periodic boundary conditions in two dimensions. These boundary conditions simulate an infinite tiling of the space with identical cells. When a particle leaves the cell over a boundary, it reappears on the other side.

back on the other side. Effectively, this tiles a large space with an infinite number of identical boxes of particles. This technique allows the simulation of bulk properties of materials, while only simulating a small number of particles. This is shown in figure 6.

Now, given some initial conditions, we should, given infinite time, be able to exactly predict the location of the particles in the future using the potentials given. We do not have infinite time. There are two primary methods used to model the progression of molecular simulations. These are molecular dynamics simulations, and Monte-Carlo simulations. In molecular dynamics, basic Newtonian mechanics are used to predict the simulation progression. The potential is used to determine a force on each particle, which is used to update each particle's velocity. The velocity is used to update each particles position over a given time step. The duration of the time step can be adjusted to change the accuracy of the simulation. The shorter the time step, the more accurate, but the longer the runtime will be. There are many methods of improving the efficiency of these time steps, so overall this method can be relatively fast; however, it still is much slower at determining equilibrium conditions than the Monte-Carlo method.

Monte-Carlo methods use statistical mechanics to predict equilibrium conditions of a system. This is accomplished by randomly sampling system states, and using the energy and temperature to seek equilibrium. An algorithm commonly used for this task it the Metropolis-Hastings algorithm, which works in an iterative manner. The algorithm requires some initial condition, a probability density, and a method of suggesting a candidate for the next system state. First, the next candidate is chosen, and the probability of this new state being accepted is calculated from the probability

density of the current and potential states. The new state is accepted according to this probability and the process is repeated until an equilibrium is reached (W.K. Hastings. [1970]).

There are several primary probability densities used for this purpose. The most commonly used in Gay-Berne studies are known as the canonical and grand canonical ensemble. The canonical ensemble gives the probabilities for a system in thermal equilibrium with its surroundings, while the grand canonical ensemble describes a system in thermal and chemical equilibrium with its surroundings. The probability ratio between two states in the canonical ensemble is given by:

$$P = e^{-\Delta U/k_b T} \tag{18}$$

where $\Delta U$ is the energy difference between the two states, $k_b$ is the Boltzmann constant, and T is the temperature. Note, if the energy change is negative (i.e. a favorable energy transition), the probability ratio is greater than one. When using the Metropolis-Hastings Algorithem, this is the probability used to accept each potential system state. When the probability ratio is greater then one, the transition is always accepted.

To pick the next potential state in a simple Gay-Berne system, a single particle is selected at random, and it is then translated or rotated randomly. The new location of the particle is used as the next possible state. This sampling method is preferred as it is easier to calculate the energy change when only one particle moves at a time, and the states can be sampled efficiently.

Each of these methods has advantages and disadvantages. Molecular dynamics are very good at modeling the development of the system, but are rather slow at finding equilibrium conditions. Monte-Carlo methods do not produce accurate system progressions, but do produce accurate, temperature dependent equilibrium conditions relatively quickly. Since our interest is in studying the phase transitions of nanoparticle/liquid-crystal systems, Monte-Carlo methods are better suited to our studies.

# 3 Code Description

We will now go into the specifics of the simulation methods we used. Since we are attempting to model mesogens bonded to nanoparticles, we modify the potential using a simple spring model proposed by Silvia Orlandi and Claudio Zannoni [2013]. The specific arrangement used is shown in figure 7. To model the attachment point, a spring is created attaching two points on the surfaces the nanoparticle and mesogen

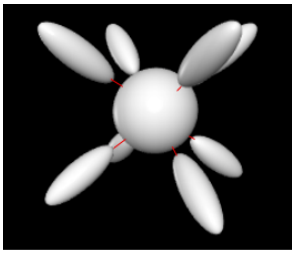

Figure 7: Arrangement of nanoparticles and mesogens with zero potential energy from the springs connecting them. The red lines represent the springs.

Table 1: Parameters Used for each particle type

| Particle Type | $\sigma_s$ | $\sigma_e$ | $\sigma_c$ | $\epsilon_s$ | $\epsilon_e$ |
|---|---|---|---|---|---|
| Nanoparticle | 3 | 3 | 3 | 0.5 | 0.5 |
| Mesogen | 1 | 3 | 1 | 1 | 0.2 |

using the following potential:

$$U_s = \epsilon_o(k_d(r - r_o)^2 + k_\theta(\theta - \theta_o)^2) \tag{19}$$

where $U_s$ is the spring energy, $\epsilon_o$ is a constant scaling the energies, $k_d$ and $k_\theta$ are the linear and angular spring constants, r and $\theta$ are the current separations and angular difference, with respect to fixed points on the surface of each particle, and $\theta_o$ and $r_o$ are the equilibrium spacing and angle of the objects. We used values for $k_d$ and $k_\theta$ of 0.5 and 0.15 respectively.

The parameters for the Gay-Berne potential are given in table 1 and 2.

We apply a potential cutoff at 6 $\sigma_o$, to decrease computation time, where we set the Gay-Berne potential between two particles equal to zero if they are further then 6 $\sigma_o$ apart. This means we don't have to fully compute the complicated Gay-Berne potential if the particles are too far apart. Between the spring potentials,

Table 2: Other Parameters Used

| Parameter | Value used |
|---|---|
| $\sigma_o$ | 1 |
| $\epsilon_o$ | 1 |
| $\mu$ | 1 |
| $\nu$ | 3 |

the generalized Gay-Berne potential, and the potential cutoff, the potential energies in the simulation have been fully described. We must now describe how the system changes between each iteration. The most basic allowed transition is a single particle translating or rotating. Due to the nature of the potential function, calculating the change in energy only involves calculating the potential between the moving particle and all particles within the cutoff radius. The computation time is proportional to the number of particles, and is far more efficient than computing the total energy of each system, which has runtime proportional to the number of particles squared.

We also allow the translation or rotation of a nanoparticle cluster (the nanoparticle and the 8 attached mesogens). While this is not strictly necessary, it helps the simulation sample the possibility states faster. Without this option, whenever a nanoparticle is selected to rotate, the 8 springs attached will cause the probability of acceptance to be near zero. By specifically allowing the entire cluster to move together, the equilibration time can be reduced substantially. The frequency of these transition attempts can be adjusted, and they are currently being attempted every four iterations. The energy change for these transitions is computed the same was as the energy change for one particle, only repeated 9 times, once for each particle in the cluster. However, simply summing the energy change for the 9 particles counts the potential between the particles in the cluster twice. These potential pairs are subtracted from the sum to account for this.

The last allowed transition involves the resizing of the entire simulation box. This occurs every 100 iterations. One of the box axis (selected at random) is scaled by up to $\pm 1\%$, and the particle positions are scaled uniformly by the same amount. The change is accepted with probability $e^{N \ln(\frac{V_2}{V_1}) - \frac{\Delta U}{K_b T} - \frac{P \Delta V}{K_b T}}$ where N is the number

of particles, $V_1$ and $V_2$ are the initial and final volume respectively, and P is the pressure. This is the probability ratio for two states given by the isothermic-isobaric ensemble, instead of the canonical or grand canonical ensemble discussed above. Since every particle has its position updated during these types of transitions, the energy must be entirely recalculated. As a result, this step is slower than the others, but it is also less important for determining equilibrium conditions, which is why it is attempted less frequently than the other transitions. The magnitudes of the translations, rotations, cluster translations, cluster rotations, and box resizes were adjusted to obtain acceptance ratios of around 40-60%.

We also implemented a Verlet-neighbor's list algorithm to further increase computation efficiency. For each particle, we store a list of the nearest neighbor particles, within some outer cutoff distance. We use a value of 8 $\sigma_o$ for this outer cutoff. During any of the translation or rotation system updates, we only need to compute the potential between the updated particle and all particles in its neighbor list, instead of all other particles. This is because any particle not in the neighbor list is outside of the potential cutoff radius. This is a substantially smaller number of particles, usually around 80-100, instead of 1124. This means that we need to compute significantly fewer potentials in each iterative step, and saves noteable computation time. However, since the particles are constantly moving, eventually a particle from outside the outer cutoff radius could move within the potential cutoff. At this point, if we do nothing, the computed energy change for either of the two particles will not account for the potential between them. As such, a method of resetting the neighbor lists is required. The method used involves tracking the total displacement of each individual particle. A list of vectors is stored, one for each particle, and updated each time a transition is accepted. In addition, the two largest displacements are tracked. Each time a particle updates, it's new displacement is checked against the two largest displacements, and they are updated if necessary. If the sum of the magnitudes of the two largest displacement vectors is larger than the difference between the outer cutoff radius and the potential cutoff radius, then there is the possibility that the two particles have moved close enough to be within the potential cutoff, and are not in each others neighbor lists. So, whenever this occurs, the neighbor lists are immediately updated, and the displacements are reset to zero.

This computation of the displacements may sound complex and inefficient; however, it only involves one addition and a few conditional statements each time a particle is updated, so the displacements are processed in constant time each iteration. Meanwhile, the computation time for calculating the potential changes each iteration is reduced by around a factor of 10. The only inefficient part of this method, is that the time resetting the neighbor lists is proportional to the number of particles

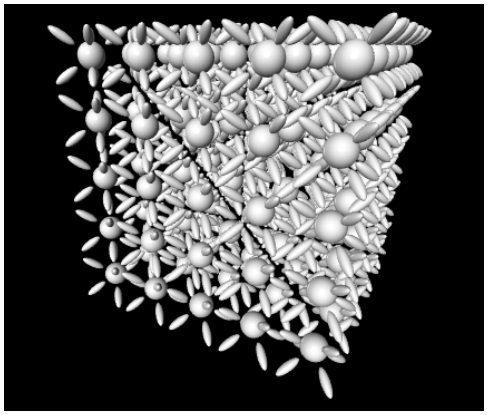

Figure 8: The initial arrangement of all the particles in the system, before any Monte-Carlo steps are performed.

squared. This method gives many iterations of reduced time, for one of increased runtime every few hundred iterations.

We have now described the entire potential field, and how we use the field to progress the simulation, given various environmental conditions. Now we must describe our initial conditions. The simulation starts in a very large box, (35 $\sigma$) and allowed to contract via resizes during the simulation. This is mostly to make room for the particles to avoid the particles accidentally overlapping. The nanoparticles are then arranged in an even grid, with five nanoparticles to a side. The mesogens are then placed in the positions with minimal spring potentials, surrounding each nanoparticle. This setup is shown in figure 8. From this initial setup, we run the simulation at a high temperature (around T = 5) to create a uniform isotropic phase. Using this starting phase, we can slowly reduce the temperature to cause the fluid to go through any phase transitions.

## 3.1 Code Structure

While we have now described all primary details, there is a lot of underlying structure to the code needed to track all the particles and successfully run the simulation. The primary data structures is an array that stores each particle, and an array that stores each spring. The particle data type contains information related to what type of particle it is, its position and orientation, as well as the indexes of any attached springs, and a list containing the indexes of the neighboring particles. It is important to note that the orientation of the particles must be a complete rotation space.

While the particles themselves are cylindrically symmetric, the spring attachment points remove this symmetry. We used quaternions to store these rotations, instead of Euler angles. While Euler angles are simpler conceptually, they require trigonometric functions, which are less efficient computationally. Furthermore, it is easier to generate random rotations efficiently using unit quaternions. A quaternion is a set of four numbers, with one real, and three specifying a vector. To extract an orientation from a quaternion rotation, we convert the orientation to a quaternion by setting the real part to zero, and setting the direction components of the quaternion equal to the orientation vector components. We then conjugate this constructed quaternion by the desired rotation quaternion:

$$p' = qpq^{-1} \tag{20}$$

where $p$ is the constructed quaternion, $q$ is the rotation quaternion, $q^{-1}$ is it's inverse, and $p'$ is the result of the rotation.The spring data type contains the indexes of the two particles it connects to, as well as the location of the attachment point on each particle.

## 3.2 Code Tests

The first test of the simulation is to check that the attempted state changes are reasonable, and performed properly. The magnitude of $\Delta U$ is checked, and if it is too large, then the displacement size is reduced. Another important check tests that the code functions as intended, by tracking the energy throughout the code. The energy is tracked via the computed $\Delta U$ at each iterative step. The total energy of the system is also recomputed at the end of the simulation, and compared to the tracked energy. If the two numbers disagree by more then floating point precision errors, the there is a problem with the implementation. In addition, several system variables are tracked throughout the simulation and can be plotted to help determine the system state during the simulation. This helps track down any problems that might not be immediately apparent. Some of the tracked variables include the system energy, and system volume, and the position and orientation of a random particle. Tracking the position of a random particle helps make sure the simulation is flowing and solidifying as it should. We also can save snapshots of the simulation, either to use for further simulations at a later time, or to view as an image. Figure 9 shows an example of a system snapshot.

Our two primary analysis tools for determining the phase of the simulation is a pair distribution function, and the nematic order parameter. The pair distribution

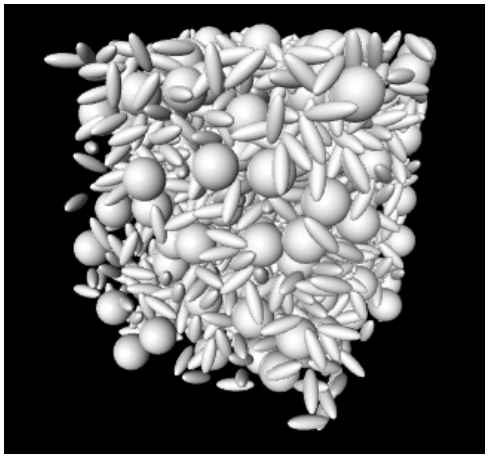

Figure 9: An image of a simulation in progress

function plots the average number of particles, per unit volume, that are within a given radius of another particle. The pair distribution for a single particle is given by

$$p(r) = \frac{V}{N^2} * \frac{n}{(r+dr)^3 - r^3} \tag{21}$$

where $V$ is the volume of the simulation, $N$ is the total number of particles and $n$ is the number of particles within radius $r$ and $r + dr$ of a chosen particle. For the total plot, we calculate $p(r)$ for each particle, and sum the results.

The order parameter is given by the largest eigenvalue of the tensor order paramter:

$$Q_{\alpha\beta} = \frac{1}{N} \sum_i (3u_{i\alpha}u_{i\beta} - \delta_{\alpha\beta})/2 \tag{22}$$

where $\alpha$ and $\beta$ are x,y,z, $u_i$ is the unit vector specifying the orientation of the ith particle, and $\delta$ is the Kronecker delta function. The order parameter gives a measure of how aligned the particles are, where the sum is over all particles.

# 4 Results and Conclusions

Unfortunately there have been some very major issues with getting the simulation to equilibrate properly. An error has been causing the simulation to not form a uniform isotropic phase. This means the simulation is either not properly sampling the space, or there are issues with the potential functions causing an incorrect bias

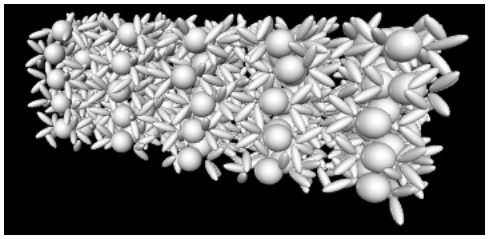

Figure 10: The arrangement of particles after some time in prior simulations, at pressure = 1, temperature = 5.

to the particles. We believe we have fixed this issue; however, more simulations and tests are required to verify this. Until recently, many simulations had the box resize into a long thin rectangular prism (figure 10). This was evidence of a mistake in the potential or sampling causing a bias. Also note, that this effect caused the particles to becomes somewhat locked in their positions, preventing the formation of proper isotropic phases.

Currently, we are unsure if the simulation produces a liquid phase, or if the particles are somewhat frozen in their locations. We require a more simulations to test this. Verifying the accuracy of the produced isotropic phase is essential, as without a correct isotropic phase we cannot make any accurate assessments of any results we might find, as we cannot confirm their accuracy. We have checked the most likely errors, and cannot find any obvious mistakes. All particles are correctly confined to the box via the periodic boundary conditions. The computed and tracked energies are the same, so there are no errors in how the change in energy is calculated. Furthermore, the energy stays within reasonable limits as the system progresses, so there are no obvious errors in the acceptance of new system states. If there is still an error after we run these tests, the most likely explanation is that some mistake is causing the potential to be implemented improperly, causing the system to equilibrate to an unexpected state. However, the plots of the potential function appear as expected along the primary axes. The problem could also be in how the system transitions are picked. If there is a mistake and the transitions do not sample uniformly, the bias could effect the equilibration of the system.

We intend to keep working on these simulations and hope to test and fix any remaining errors over the summer. We then intend to run many simulations using the developed code, and see what control over nanoparticle arrangements we can obtain.

# 5 Acknowledgments

Thanks to my research advisor, Robert Pelcovits, who made this project possible, as well as all the people of the physics and chemistry departments at Brown University. Thank you to anyone and everyone who listened to me talk about this project for hours on end.

# References

S. Orlandi and C. Zannoni., Phase Organization of Mesogen-Decorated Spherical Nanoparticles, *J. Molecular Crystals and Liquid Crystals.*, 2013, vol. 573, Issue 1, pp. 1–9.

R. A. Taylor, T. Otanicar, G. Rosengarten., Nanofluid-based optical filter optimization for PV/T systems, *Light: Science & Applications.*, 2012, Issue 1, e34.

R. A. Taylor, S. Coulombe, T. Otanicar, Tyagi., Small Particles, Big Impacts: A Review of the Diverse Applications of Nanofluids, *J. Applied Physics.*, 2013, vol. 113, Issue 1.

X. Mang, X. Zeng, B. Tang, F. Liu, G. Ungar, R. Zhang, L. Cseh, G. H. Mehl., Control of anisotropic self-assembly of gold nanoparticles coated with mesogens, *Journal of Materials Chemistry.*, 2012, Issue 22.

J. E. Lennard-Jones., On the Determination of Molecular Fields, *Proceedings of the Royal Society.*, 1924, vol. 106, Issue 738.

D. J. Cleaver, C. M. Care, M. P. Allen, M. P. Neal., Extension and generalization of the Gay-Berne potential, *Physical Review E.*, 1996, vol. 54. Issue 1.

Jean-Pierre Hanson and Loup Verlet., Phase Transitions of the Lennard-Jones System, *Physical Review.*, 1969, vol. 184, Issue 1.

W.K. Hasitings., Monte Carlo sampling methods using Markov chains and their applications, *Biometrika.*, 1970, 97-109.

J. Billeter and R. A. Pelcovits., Simulations of Liquid Crystals, *Computers in Physics.*, 1998, vol. 12, No. 5.

M. Grzelczak, J. Vermant, E. M. Furst, L. M. Liz-Marza., Directed Self-Assembly of Nanoparticles, *American Chemical Society.*, 2010, vol. 4. No 4.

W. Lewandowski, D. Constantin, K. Walicka, D. Pociecha, J. Mieczkowski, E. Górecka., Smectic mesophases of functionalized silver and gold nanoparticles with anisotropic plasmonic properties, *Chemical Communications.*, 2013, issue 71.

M. E. Åkerman, W. C. W. Chan, P. Laakkonen, S. N. Bhatia, E. Ruoslahti., Nanocrystal targeting in vivo, *Proceedings of the National Academy of Sciences.*, 2002, vol 99.

J. G. Gay and B. J. Berne., Modification of the overlap potential to mimic a linear site–site potential, *The Journal of Chemical Physics.*, 1981, vol 74.