# COMPUTATIONAL STUDIES OF 2D DIFFUSION AND WATER NUCLEATION

By
Casey H. Williamson B.A.
East Central University
Ada, OK, USA
2015

Submitted to the Faculty of the
Graduate College of
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
Master's of Science
December 2017

# COMPUTATIONAL STUDIES OF 2D DIFFUSION AND WATER NUCLEATION

Thesis Approved:

Dr. Christopher Fennell

Thesis Adviser

Dr. Ashlee Ford Versypt

Dr. Lionel Raff

Dr. Nicholas Materer

## ACKNOWLEDGMENTS

I would like to thank the support system of my family and fiancé who provided the much needed encouragement during my studies. I would like to thank the Chemistry department at East Central University, as well as the faculty for the McNair Scholar's program who helped provide my necessary educational foundation.I would also like to thank Dr. Christopher Fennell for his time, instruction, and support during my stay at Oklahoma State University. As well as the indispensable help of my lab mates Shanaka Paranahewage and Gentry Smith who were always available for conversation and assistance.

Acknowledgments reflect the views of the author and are not endorsed by committee members or Oklahoma State University.

Name: Casey H. Williamson

Date of Degree: December 2017

Title of Study: Computational Studies of 2D Diffusion and Water Nucleation

Major Field: Chemistry

Abstract: Molecular simulations for nearly any molecule can currently be conducted, but often these calculations are costly and can only be accomplished using multiple nodes on a super computer. These topics of research use the new and novel approach of approximating water using Rose Equations to create an accurate 2D model of water molecules. This research was published in a special edition of the Journal of Molecular Liquids dedicated to Vokjko Vlachy, whose work with an analogous 2D model known as the Mercedes Benz Model inspired this project. While the previous Mercedes Benz model of water approximated many properties of real water due to the constraints of Monte-Carlo sampling certain properties, such as coefficients of diffusion, could not be calculated. This new 2D model was created by modifying simple trigonometric (Rose) functions to give an approximation of real water that consists of one central atom and three surrounding Hydrogen bonding arms. Because all of the simulations were done in 2D different properties could be quickly probed at varying temperatures using multiple starting configurations to give proper statistical accuracy. This model was then expanded to give visual insights into the properties of particles when confined to nano-tubes. This visual representation was created using free software known as Pov-Ray. Pov-Ray was then used to help elucidate nucleation events within 3D water environments. This newly created software has the ability to represent every ring within a simulation of particles as determined by simple geometric means

## TABLE OF CONTENTS

Chapter Page

# LIST OF FIGURES

Figure Page

## NOMENCLATURE

| | |
|---|---|
| $\alpha^*$ | Coefficient of thermal expansion |
| $\epsilon_{HB}$ | Defined strength of Hydrogen bond |
| $\kappa^*$ | Isothermal compressibility |
| $\sigma_{\mathrm{LJ}}$ | Lennard-Jones boundary wall |
| $\tau_1$ | Hydrogen bond orientational relaxation time |
| $C_1$ | Self orientational autocorrelation function |
| $C_p^*$ | Heat capacity |
| $D$ | Coefficient of Diffusion |
| $g(r)$ | Radial distribution function |
| $P^*$ | Reduced Pressure |
| $r_l$ | Lower bound limit of Hydrogen bond |
| $r_u$ | Upper bound limit of Hydrogen bond |
| $r_{HB}$ | Ideal length of Hydrogen bond |
| $r_{ij}$ | Distance between particles $i$ and $j$ |
| $s(r_{ij})$ | Distance dependent potential |
| $T^*$ | Reduced Temperature |
| $T_z$ | Torque about the z-axis |
| $V(\Theta_{ij})$ | Orientational potential |
| $V^*$ | Reduced Volume |
| nMol | Number of particles |
| NPT | Constant number of particles; Constant Pressure; Constant Temperature; |
| NPTi | Constant number of particles; Constant Pressure; Constant Temperature; Isotropic scaling |
| NVT | Constant number of particles; Constant Volume; Constant Temperature; |

# CHAPTER 1

# Introduction

## 1.1 Introduction

With the start of modern computing in the 1960s computers were anything but personal. They were industrial sized machines that took up entire rooms and were able to perform relatively few calculations per minute. Thankfully for the entire realm of scientific research modern computers are ever shrinking while still growing in computational power. With our current available processing power computers have made their way into every scientific field in some form or another to push forward the area of scientific knowledge. Chemistry is no exception to this push; Within the field of chemistry we classify the study of the physical phenomena produced by chemical interactions as physical chemistry, this subsection of Chemistry uses laws and theories directly produced as a study of Physics so that we may apply these laws on individual particles and shed light on how they interact. As to be expected, while all of Chemistry relies on modern computing Physical Chemistry relies heavily on this resource.

Physical Chemistry can take many different forms, such as purely mathematical and theoretical work, as well as experimental lab work that helps to confirm or disprove Physical theories, as well as a combination of experimental and theoretical work known as Molecular Dynamics. Molecular dynamics uses theoretical models as well as modern computers to produce simulated molecular systems that can be probed to give information that can be compared to experimental results or that can give information that is not readily deduced through experimental calculations. One benefit of molecular dynamics arises from the ability to craft your own system and view individual molecular interactions on small or large time scales. These extremely small system sizes and time scales are often difficult to obtain in a purely experimental setting. Molecular dynamics simulations use the current processing power brought to us by modern computing to explore the fundamentals of chemistry on the smallest scales of time and size. While mankinds' history of lab chemistry has given us many insights into how our world works, molecular dynamics explores the fundamental properties of our world using computers to simulate lab like situations and directly observe how different particles will interact on an individual, microscopic level. The field of Molecular Dynamics is relatively new and exciting, where we can use the elucidations provided to answer questions like How do particles individual interactions compare to interactions of the same particles in bulk material, and How exactly do nucleation events start and propagate? These are two of the primary questions that my work has revolved around during my research. As expected, this new area of scientific exploration provided by modern day computing can be both time consuming and costly. One of the goals within our research group has been to simplify many of the calculations that our Molecular Dynamics Engines are required to perform.

Many different Molecular Dynamics Engines exist, and different engines excel at different tasks. One of the most common is Gromacs. This engine excels at simulating 3D environments and works quite well as a common language between differing fields of study. [2] While Gromacs is a popular molecular dynamics engine many others still exist, one of which being OpenMD. This molecular dynamics engine was created in 2005 by the Gezelter lab at the University of Notre Dame. [3] While the two engines have many commonalities OpenMD has ability to define ”rigid-body” objects, these objects will appear in chapter 4, which is devoted to diffusion within channel like environments. Another advantage of our group using OpenMD is preexisting knowledge. Accessibility to a member of the group that created this engine meant that modifications to the engine's source code would be a trivial task after the model had been properly crafted.

## 1.2 Mercedes-Benz model

With the help of a newly crafted model and our modified Molecular Dynamics engine we plan on comparing our new model with a previously existing model known as the "Mercedes-Benz" model. Molecular dynamics studies have a rich history with the simulation of water as it is ubiquitous and comprised of simple parts. Water may be comprised of simple parts, but it holds anomolous properties, such as a solid phase having a lower density than the liquid phase and a temperature of maximum density located in the liquid state. These anomolous properties are likely due the the large amount of Hydrogen bonds contained within water systems. A simple model was created by Arieh Ben-Naim to help explain this unique system via a 2-dimensional model. [4] [5] [6] [7] [8] This 2-dimensional model approximated water by turning three of the four bonding sites (two lone pairs and two Hydrogen atoms) towards the plane of the viewer, and removing the single bonding site not in our 2-dimensional viewing plane. The three remaining bonding sites are considered to have equal bonding potential, explaining why figure 1.1 has three identical arms. As shown in figure 1.1 this 2-dimensional model could be used to easily explain the state of frozen water and demonstrate water's propensity for forming six member rings. [4] This model will be approximated by our new "Rose Potential" as discussed in chapter 3, and the Mercedes-Benz model's propensity to form six member rings will appear in chapter 5 where I discuss nucleation events via the distribution of ring sizes.

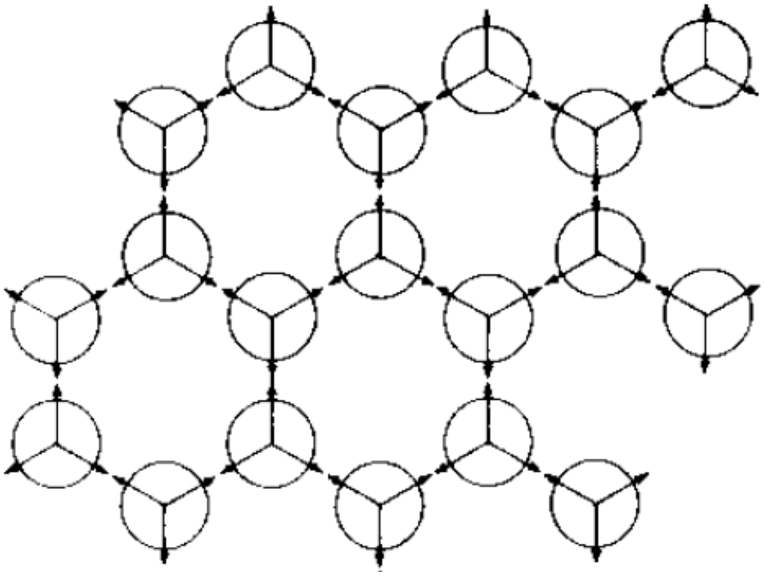

Figure 1.1: 2-dimensional representation of water systems at low temperatures where the particles form a six member ring.

# CHAPTER 2

# Methods

## 2.1 Why Use OpenMD

While the source code for the OpenMD engine is available and numerous force-fields exist for carrying out molecular simulations an original goal of our group was to mimic the previously created Mercedes-Benz model using a newly created mathematical model that could be placed in a molecular dynamics engine. This new model could then be used to obtain previously calculated thermodynamic data as well as dynamical data that can be compared to experimental results. Similarly to the Mercedes-Benz model our mathematical model will be 2-dimensional and hopefully capture the previously seen salient features that the Mercedes-Benz model captured while being modeled in only two dimensions. This reduction in dimension will cut down on numbers of calculations performed, leading to lower processing requirements and shorter calculation times. The engine chosen for this endeavor was knows as OpenMD. [3] This engine was chosen because of our groups access and previous knowledge of the source code as well as its ability to link particles together and classify them as one, rigid-body, structure when desired. While the Mercedes-Benz model 1971 it was mostly used as a teaching instrument to demonstrate to students why bulk water environments behave the way that they do. More recently there has been work published using these teaching models to run Monte-Carlo type simulations. [1] [9] [10] [11] [12] [13] [14]

Monte-Carlo type simulations are a classification of molecular dynamics that statistically test the conditions of your system. These types of simulations can be useful for investigating properties such as Volume as a function of temperature, radial distributions, coefficient of thermal expansion, compressibility, and heat capacity. While Monte-Carlo simulations can shed light on a variety of properties they cannot be used to solve for dynamical data as the particles within a simulation are not physically progressed through a trajectory, but rather statistically sampled to see a probability of positions.

While these simulations did shed some light onto why water holds anomalous features they were not able to give any thermodynamic data about the simulated systems such as coefficients of diffusion and bond relaxation times. In a paper released in 2016 our group built upon these 2D models to create analogous force fields that could be used in conjunction with a MD engine to bridge the gap between the data given by 2D MC simulations and experimental thermodynamic data. [15] This forcefield was conceived by examining the 2D Mercedes-Benz model of water and looking for some mathematical function that could easily and accurately represent this preexisting model. By the work of Dr. Christopher Fennell an analogous trig function was found. [15] This trig function approximates the key characteristics the three armed Mercedes-Benz model.

## 2.2 How to Use OpenMD

After the engine has been successfully built and installed a general simulation requires an executable version of the engine, as well as three other files. The first required file is the .frc file; this file contains information about any atoms that will be in the simulation and defines their properties, such as mass, Leonard-Jones values and moments of inertia. The next required file is the .md file , this is a meta-data file that contains every atom that can be used in the system. This file contains necessary information that allows the engine to properly build a desired molecule. If a molecule has only one member, such as $Cl^{-}$, this item is listed individually within the .md file; while more complex molecules, like our channel particle (to be clearly defined later) will contain explicit locations of all individual particles contained within this larger rigid-body object. This concatenation is used so that our final required file, simulation.md, can list a single object such as channel rather than every particle within that channel individually. This final required file of simulation.md contains all other settings required for the simulation to proceed. The settings listed in the header are: number of molecules of each type listed within this file, ensemble settings, forcefield to be used, target pressure, target temperature, tolerance for the thermostat, tolerance for the pressure, the timestep, the total runtime, as well as information that allows the engine to know if this is an initial run, or if this run has been produced as the result of some other run. Many of these settings are rather intuitive, while others will require more explanation that I will provide.

```
Settings:
Ensemble settings:
ensemble = NPTi;
```

The syntax listed above defines our desired ensemble. While conducting experimental procedures it is necessary to keep as many constant variables as possible while varying one, and only one, variable at a time. The definition of ensemble allows molecular dynamics calculations to mimic these experimental procedures as closely as possible. Within the field of Chemistry there are several ensembles that are often used and therefore given a short-hand name. Our primarily chosen ensemble of NPTi is often referred to as the isothermal-isobaric ensemble. This ensemble attempts to maintain a constant number of molecules, constant Pressure, and constant Temperature, so that a change in volume or other properties can be studied. One reason this ensemble is often chosen is because it closely matches commonly carried out experimental procedures where a constant number of molecules are used in a constant temperature environment with an open flask to maintain constant pressure of atmospheric conditions. The most common notation for the isothermal-isobaric ensemble is NPT; our simulations varied in volume, but were restricted to only scale isotropically, meaning that our 2-dimensional systems had a constraint that the x dimension always equaled the y dimension. In order to achieve this isotropic scaling our ensemble was defined as NPTi, rather than simply NPT. While not used in our simulations another ensemble, knows as Canonical is often chosen. This ensemble maintains the number of molecules, volume, and temperature constant (NVT) so that variations in pressure can be studied as well as other properties. This Canonical ensemble is often chosen because it directly relates to the Helmholtz-free energy for that specific system.

`Header information for .md files:`

`Molecules of each particle:`

```
component{
    type = "ROSE";
      nMol = 225;
}
```

The syntax listed above states that the simulation will be dealing with one type of particle, this ROSE particle is defined in both the water.md and .frc files. The setting of nMol tells the engine that the simulation will be run on 225 particles, each of which will be individually listed in the simulation.md file. If other particles were being used in this system they would be listed here, as well as their nMol value if our system was being run under the constraint of constant number of molecules.

`ForceField settings:`

The forcefield setting within our engine tells the program what .frc file to use to define the model for the particle. As previously listed these settings found in the

.frc file are: mass of particle, moments of inertia for particle, and its Leonard-Jones parameters.

```
Timings:

dt = 2.0;

runTime = 1e6;

sampleTime = 1e3;

statusTime = 1e2;
```

These settings tell the engine how long to allow the simulation to progress as well as how often to calculate and record data, as a note all timings are in units of femtoseconds. The runtime variable tells the engine how long to progress through a simulation. The above syntax example tells the engine to progress through $1 \cdot 10^6$ femtoseconds. The dt variable tells the engine that it is to take a time-step of 2 femtoseconds. Meaning that every particle is allowed to move the distance it would naturally progress towards if it moved 2 femtoseconds. This fine grain time-step helps to ensure that particles do not interfere with one anothers Leonard-Jones boundary walls, which would cause particles to be extremely repulsive towards one another and lead to unreasonably high velocities and an unnatural particle path. While the particles are moved 2 femtoseconds at a time this does not mean that the data is recorded and printed every 2 femtoseconds; this would lead to very large output files that would be difficult to manage. Instead the sampleTime variable tells the engine how often to obtain a sample set of the data, in the above example syntax the engine would record positions and velocities every 100 femtoseconds. Every time this data is recorded the engine prints the position and velocity of each atom to a simulation.dump file. Each time this data is recorded the engine has printed a frame of the trajectory. The status time variable is a feature of OpenMD that is relatively unique. This variable tells the engine how often to print out user selected data or default data to a simulation.stat file. This file is automatically generated when a simulation is ran on OpenMD giving the user essentially free information that the engine is required to calculate in order to progress through a trajectory. All timing variables need to be carefully chosen and tested before running multiple simulations so that the user does not waste valuable computing time on files that have an unreasonable scope.

```
Target Values:
targetTemp: 20K;
targetPressure: 1060 torr;
tauThermostat: 100;
tauBarostat: 100;
```

As defined in the Ensemble settings subsection our systems were commonly ran using constant pressure, temperature, and nMols. While the goal of the MD engine is to keep these values constant that is not entirely possible. As the engine simulates a run of these particles and allows their random movement these constant values will vary due to the engines primary goal of allowing particles to interact as realistically as possible. If an initial run was created using the NPTi ensemble at 20 K and 1060 torr it could be progressed through a simulation, and then after completion have its targetTemp variable edited to a different temperature setting of 10 K. If another simulation was carried out on this system the internal thermostat of the engine would attempt to cool down the system to 10 K. OPENMD uses an internal thermostat derived by the Nosé-Hoover model. This model calculates the heat of our current system and the heat of our imaginary heat-bath so that the two can be solved for an overall system energy that can be conserved from frame to frame. This method is very useful but has the downfall of creating an oscillatory function, meaning that if a temperature was changed from 20 K to 10 K the thermostat would very likely overshoot its first frame simulation meaning that the actual heat of the system would be slightly under 10 K. After further frames are simulated the thermostat would slowly place the system in the desired user defined state. Because in the above example syntax we have given the engine a tauThermostat value of 100 femtoseconds, the engine would force the Nosé-Hoover thermostat to correctly adjust system temperature every 100 femtoseconds. The user defined variables of targetPressure, and tauBarostat behave similarly to our temperature settings in that they both use the Nos-Hoover model to calculate and correct pressures, leading to similarly oscillating pressures.

```
Initial run vs. Equilibrated run:

tempSet = true;

useInitialTime = true;

useInitialExtendedSystemState = "false";
```

When running MD calculations every simulation must come from some initial file. Very often only one initial file is created and then the subsequent files are generated from the final frame of the initial simulation containing all velocity vectors for every atom. This final frame is generated by the OPENMD engine in the simulation.eor (end of run) file. This file contains all necessary data for a new run, including header information as well as velocity vectors. The above syntax is an example of header information for an initial file, meaning one that was directly generated by the user and not the result of some other run. The variable of tempSet tells the engine that all velocities are to be set from the Maxwell-Boltzmann distribution calculated using the targetTemp. This ensures that the initial run has velocities reasonable for the target temperature. The variable of useInititalTime tells the engine that the starting time is equal to the time listed in the final frame of data. If every simulation is run from a simulation.eor file, which contains only one frame of data this setting is arbitrary, but this setting will play an important role in simulations of subsequent runs are created using a previous simulation.dump file. As a general rule this variable should be set to true and new simulations should be generated by using previously calculated simulation.eor files. The variable of useInitialExtendedSystemState tells the engine to save the extended variables of tauThermostat and tauBarostat, as a general rule this setting can be left to true so that subsequent runs can use this information to accurately maintain temperature and pressure.

```
Trajectory Header:
```

```
<Snapshot>
  <FrameData>
    Time: 100
    Hmat: {{ 24.970666, 0, 0 }, { 0, 24.970666, 0 }, { 0, 0, 24.9706665}}
    Thermostat: -0.03884417001 , 3791.589112
    Barostat: {{ 9.2351e-06, 0, 0 }, { 0, 9.2351e-06, 0 }, { 0, 0, 9.2351e-06 }}
```

When any simulation is run a simulation.dump file is created. Depending upon the user defined variables of runTime and sampleTime the number of frames contained within a dump file can range from just a few, to thousands. Every frame within a .dump file is denoted by the syntax of $< Snapshot >$, telling the user this is a snapshot of the data at a given time. Each snapshot of data within a .dump file contains individual header data specific to that snapshot. The variable of Time tells the user that this snapshot was recorded at this specific time, the next Time variable will be the previous time plus the user defined variable of sampleTime, and this will continue until the current Time would be larger than the user defined runTime. The vector variable of Hmat tells the user the size of each dimension of their simulation box boundary, given in the format of (Hmat($x$), Hmat($y$), Hmat($z$)). The variable of Thermostat tells the user the difference between the targetTemp and the Nosé-Hoover calculated temperature as well as the conserved quantity that the Nosé-Hoover model solved for. The variable of Barostat tells the user the difference of targetPressure and Nosé-Hoover calculated pressure in each dimension. As a note all OpenMD simulations were carried out in only 2 dimensions. Meaning that our $z$-axis existed as required by the engine but was not included into any of our models and no calculations were performed using $z$-axis values. These 2D simulations were ran using the NPTi ensemble meaning that box boundaries were forced to isotropically scale, so that any box size change in the $x$ direction was required to match a change in $z$ direction.

```
Trajectory Output:
0    pvqj       -11.37335484       -11.40500272
```

Within any file containing 1 or more snapshots of data there must be a line denoting every molecule within the simulation. These lines will match the syntax above but contain a total of fifteen values. The first value tells the user the identity of the atom within the simulation, these values will always follow numerical, ascending order. The second value tells the user what numerical values are to follow, this value is required to be a string with the minimum requirements of p. The largest possible value for this string is pvqj, meaning that the most values that can follow this string is thirteen. The p within this string denotes position, telling the user the next 3 values will be the atoms position in the x, y, and z direction. The v within the string tells the user that the next 3 values will be velocities, also within the x, y, and z direction. The q within the string tells the user that the next 4 values will be the quaternion associated with that atom. These values help define the particles rotational orientation within space. The final variable of j tells the user the particles jerk, or rate of acceleration for that particle. As a note all of our OPENMD calculations were carried out in only 2 dimensions, causing many of these z-related variables to drop to 0, and therefore increasing the speed of our calculations.

# CHAPTER 3

# Rose Potentials

## 3.1 Creation and Modification of the Model

This new three-petaled model, shown in figure 3.1 is based on a trigonometric function commonly referred to as a rose function was relatively simple. It could be easily manipulated by altering the leading coefficients of the equation $r = a\sin(n)$ . Where increasing the coefficient would lead to larger wells on the model and decreasing the coefficient below $a = 1$ would lead to smaller wells. Preliminary work was done by Dr. Christopher Fennell and Oklahoma State University student Joshua Hall to manipulate this model so that it mimicked the Mercedes-Benz water model shown in figure 3.2, which would hopefully lead to it being also analogous to experimental water. [15] This was done by not only changing the leading a coefficient but by also removing a portion of the positive region of the function by subtraction of an $r = a\sin^2(n\theta)$ term. These modifications are shown in figure 3.3 with a dark dashed blue line denoting an unadulterated $r = a\sin(n\theta)$ function and more refined $r = a\sin(n\theta)$ model in subsequent lighter shades of blue. The final, solid, line was chosen as our final model to be compared to both the Mercedes-Benz model as well as experimental data for real water.

The removal of the positive portion of the function leads to a model where the areas of repulsive interaction are smaller than the areas of attraction leading to a model that forms Hydrogen bonds more easily than an unrefined sine model. With this refined model calculations were carried out to derive the necessary forces and torques to be placed within our MD engine. Because our simple 2D model has only a

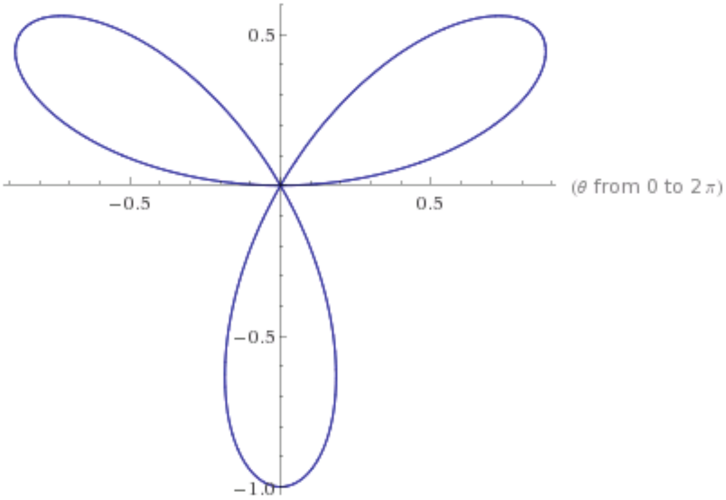

Figure 3.1: 2D Rose Potential shown through polar plot, mimicing bonding sites of Mercedes-Benz model.

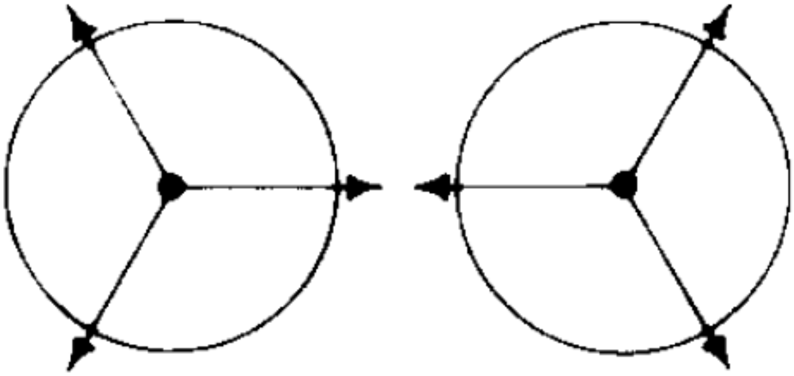

Figure 3.2: 2D Mercedes-Benz model with three potential bonding sites. [1]

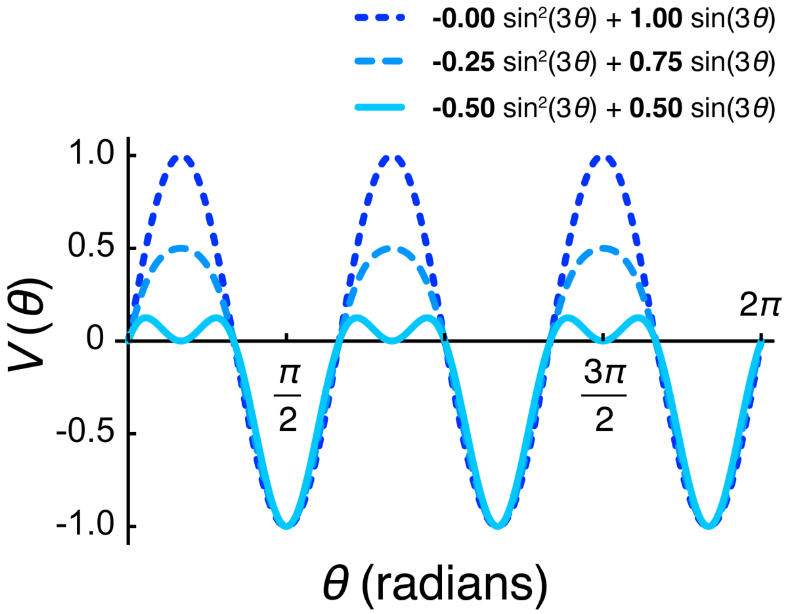

Figure 3.3: Original Rose Potential plot of $r = -a_2 \sin^2(3\theta_{ij}) + a_1 \sin(3\theta_{ij})$ shown with varying degrees of the positive (repulsive) wells removed.

single angular degree of freedom all angular potentials can be calculated by viewing any particle through the body frame of its interacting partner. Leading to an angular potential equation with respect to particles $i$ and $j$ of:

$$V(\theta_{ij}) = a_2 \sin^2(3\theta_{ij}) + a_1 \sin(3\theta_{ij}) \tag{3.1}$$

Where $V(\theta_{ij})$ is our total potential; $a_1$, and $a_2$ are our parameters to manipulate the attractiveness of the model; And $\theta_{ij}$ is the angle of particle $j$ when viewed from the frame of particle $i$. As is required in molecular dynamic calculations particle interactions are not only defined by angular interactions, but also by distance between the particles. This distance dependent Gaussian potential used by the Mercedes-Benz model was approximated by using a double-sided cubic switching function [16]:

$$s(r_{ij}) = \begin{cases} 0, & r_{ij} < r_l \\ \frac{(r_l+2r_{ij}-3r_{HB})(r_l-r_{ij})^2)}{(r_l-r_{HB})^3}, & r_l \leq r_{ij} < r_{HB} \\ \frac{(r_u+2r_{ij}-3r_{HB})(r_u-r_{ij})^2)}{(r_u-r_{HB})^3}, & r_{HB} \leq r_{ij} < r_u \\ 0, & r_u \leq r_{ij} \end{cases} \tag{3.2}$$

Where $r_{ij}$ is the distance between particles $i$ and $j$; $r_l$ are the user-defined parameter of lowest distance at which a bond can exist and not interfere with the Leonard-Jones boundary, similarly $r_u$ is the upper limit at which a bond can begin to form; And $r_{HB}$ is the ideal distance of a Hydrogen bond. This switching function modeled particles potential when the radial distance between two particles was too small, leading to the top-most line where the $r_{ij}$ distance was less than the defined lower limit of $r_l$; as well as behavior where the $r_{ij}$ distance was too large, leading to the bottom-most line. Both potentials in this case would be zero meaning the particles were not attracted to one another. The two middle cases in our switching function lead to potentials that will cause particles to attractively interact forming at least a partial Hydrogen bond. Luckily the simplicity of our model leads to trivial calculations for this portion of the potential, where $r_{ij}$ is an angular distance while $r_l$, $r_u$, and $r_{HB}$ are all user defined values that realistically match the bonding behaviors of water while cutting down on computational costs by doing no more calculations than are necessary. This distance-dependent function is multiplied by the angular-dependent function to give a sum pairwise potential of the form:

$$V_{HB}(r_{ij}) = \frac{\epsilon_{HB}}{2} \cdot s(r_{ij}) \cdot V(\theta_{ij}) \tag{3.3}$$

Where $\epsilon_{HB}$ is the strength of a Hydrogen bond. This additional term of $\epsilon_{HB}/2$ is in the above equation to account for the fact that this potential is the potential of one particles interaction with another, and not the two interactions combined; another thesis of this paper was that our results could be compared to previously calculated thermodynamic data using the Mercedes-Benz model. These previously calculated results were given in terms of the strength of a Hydrogen bond, therefore our calculations were scaled similarly. For sake of simplicity our angular-dependent potential is converted back into Cartesian coordinates so that forces and torques can be calculated more easily. Shown below is the Cartesian representation of our original rose function of $r = \sin(3\theta)$

$$\sin(3\theta_{ij}) = \frac{3x_{ij}^2 y_{ij} - y_{ij}^3}{r_{ij}^3} \tag{3.4}$$

This leads to a total rose potential of:

$$V_{HB}(r_{ij}) = \frac{\epsilon_{HB}}{2} \cdot s(r_{ij}) \cdot \left( a_2 \cdot \frac{(3x_{ij}^2 y_{ij} - y_{ij}^3}{r_{ij}^5} + a_1 \cdot \frac{3x_{ij}^2 y_{ij} - y_{ij}^3}{r_{ij}^3} \right) \tag{3.5}$$

This total potential is applied to a cartesian version of our original sine model as a function the distance dependent cubic switching function. In order to calculate the angular potential between particles in a 2D plane we need forces in the $x$ and $y$ direction, as well as a torque about the unused $z$-axis. The forces in the $x$ direction were calculated by taking the first derivative of our angular momentum with respect to $x$. Similarly forces in the y direction were calculated giving the equations of:

$$F_x = -\left( 2a_2 \cdot \frac{3x_{ij}^2 y_{ij} - y_{ij}^3}{r_{ij}^3} + a_1 \right) \cdot \left( \frac{6x_{ij} y_{ij}}{r_{ij}^3} - \frac{9x_{ij}^3 y_{ij} - 3x_{ij} y_{ij}^3}{r_{ij}^5} \right) \tag{3.6}$$

$$F_y = -\left( 2a_2 \cdot \frac{3x_{ij}^2 y_{ij} - y_{ij}^3}{r_{ij}^3} + a_1 \right) \cdot \left( \frac{3x_{ij}^2 - 3y_{ij}^2}{r_{ij}^3} - \frac{9x_{ij}^2 y_{ij}^2 - 3y_{ij}^4}{r_{ij}^5} \right) \tag{3.7}$$

The torque was calculated by converting the base rose function to unit vector form on the body frame of particle $i$.

$$\sin(3\theta_{ij}) = \frac{3(\hat{x}_i \cdot r_{ij})^2(\hat{y}_i \cdot r_{ij}) - (\hat{y}_i \cdot r_{ij})^3}{r_{ij}^3} \tag{3.8}$$

Giving the finalized torque element about the vector normal to the $xy$-plane as:

$$T_z = \left(2a_2 \cdot \frac{3x_{ij}^2 y_{ij} - y_{ij}^3}{r_{ij}^3} + a_1\right) \cdot \left(\frac{3x_{ij}^2 - 9x_{ij}y_{ij}^2}{r_{ij}^3}\right) \tag{3.9}$$

## 3.2 Computational Methods

With this new overall potential equation, we are able to modify the way in which OpenMD calculates energy potentials to force this previously existing MD engine to run 2D simulations on the previously mentioned rose model. In order to compare our new model to previously calculated thermodynamic data obtained by Monte-Carlo simulations on the Mercedes-Benz model many of the same values were calculated and cast in similar units, by scaling all newly calculated thermodynamic data with respect to the length of the Hydrogen bond. While OpenMD supports many different ensembles most of our calculations were carried out using the NPT ensemble, meaning that our simulations contained a constant number of molecules, as well as constant pressure and temperature.

Preliminary molecular simulations were carried out to fine tune our model so that the anomalous feature of a volume minimum after melting could match closely with already calculated non-thermodynamical data found for the MC simulations of MB water. It should be noted for the reader that when referring to volume we are approximating this quantity of volume based on our actual 2-dimensional area. [1] This fine tuning was achieved by altering the leading coefficients shown in figure 3.3. This fine tuning of the sine function lead to a model with an appropriate balance of repulsive and attractive features. After the appropriate coefficients were chosen and the force equations were calculated they were added into the source code for OpenMD. Meaning that our modified version of OpenMD could now be used to carry out 2D simulations of bulk water environments. Because one goal of this work was to bridge the gap between previous 2D work and experimental data the chosen areas of investigation were volume as a function of temperature, a radial distribution function, coefficient of thermal expansion, isothermal compressibility, and heat capacity. Many of the thermodynamic data sets were calculated using the NPT ensemble. In order to maintain easily comparable data to previous Mercedes-Benz model Monte-Carlo calculations both thermodynamic data and statistical data are given in reduced units.

$$T^* = \frac{k_B T}{|\epsilon_{HB}|} \tag{3.10}$$

$$V^* = \frac{V}{r_{HB}^2} \tag{3.11}$$

$$P^* = \frac{PV}{V^*|\epsilon_{HB}|} \tag{3.12}$$

Where $T^*$ is our desired reduced Temperature; $k_B$ is the Boltzmann constant; $V^*$ is our desired reduced Volume; And $P^*$ is our desired reduced Pressure, where all constants are scaled by the ideal length of a Hydrogen bond $r_{HB}$, or the defined strength of a Hydrogen bond $\epsilon_{HB}$.

### 3.3 New, Dynamic Data

The value for coefficient of diffusion was obtained using five equilibrated independent 1 nanosecond trajectories sampled every 1 picosecond. For calculating a coefficient of diffusion these five independent runs ranged in sizes containing 144 particles to 576 particles. This was done so that any artifacts left by our finite system size could be used to extrapolate to an infinite system size where the system would approximate the behavior of bulk water. [17] Using these five independent trajectories a coefficient of diffusion was calculated via a regression fit to the linear region of the plot generated by means of the Stokes-Einstein relation.

$$D = \lim_{x \to \infty} \frac{\langle |r_i(t) - r_i(0)|^2 \rangle}{4t} \tag{3.13}$$

Where $r_i(t)$ is the position of particle $i$ at time $t$, and $r_i(0)$ is the initial position of particle i. Unlike the standard Einstein relation numerator is divided by a scaling factor of 4 rather than 6 due to the reduced dimensionality of our system. A value of Hydrogen bond orientation relaxation time was determined for our system as well. This value was calculated via integration of the self orientational autocorrelation function, $C_1(t)$, where $C_1(t)$ was found using:

$$C_1(t) = \langle P_1[\omega(t) \cdot \omega(0)] \rangle \tag{3.14}$$

Where $P_1$ was the first Legendre polynomial; And $\omega$ represents a bodies orientation vector fixed about the lab frame at a given time. The radial distribution function was calculated for comparison with previous Mercedes-Benz data. This distribution was calculated by binning separation distances between particles.

$$g(r) = \frac{1}{N\langle \rho \rangle} \sum_{i}^{N} \sum_{j \neq i}^{N} \frac{\delta(r_{ij} - r)}{\pi r} \tag{3.15}$$

Where $g(r)$ is the desired radial distribution function; $N$ is the total number of particles; $\rho$ is the bulk density of the simulation.

### 3.4 Comparable Thermodynamic Data

In order to compare our new model to the older, Mercedes-Benz model previously calculated data was also found for our system. The values: coefficient of thermal expansion, heat capacity, and isothermal compressibility were found.

$$\alpha^* = \frac{\langle V^* H^* \rangle - \langle V^* \rangle \langle H^* \rangle}{T^{*2} \langle V^* \rangle} \tag{3.16}$$

$$C_p^* = \frac{\langle H^{*2} \rangle - \langle H^* \rangle^2}{N T^{*2}} \tag{3.17}$$

$$\kappa^* = \frac{\langle V^* 2 \rangle - \langle V^* \rangle^2}{T^* \langle V^* \rangle} \tag{3.18}$$

$$H^* = H / |\epsilon_{HB}| \tag{3.19}$$

While these values are useful for comparison to previous models and validation the calculations were rather trivial thanks to the previously mentioned .stat file created when running a simulation in OpenMD. This file provides commonly needed data such as volume, total energy, temperature, and pressure.

### 3.5 Findings and Conclusions

All of the previously mentioned data was collected and compared to the Mercedes-Benz where possible, and experimentally obtained data from actual water when comparison to Mercedes-Benz was not possible. Findings confirmed that our 2D model was over-all a reasonable representation of actual water, and properties that did not closely match actual water could be fine-tuned within our model to give more realistic results if desired. Although our model was a close fit to the Mercedes-Benz model the underlying property of our model containing only a single potential placed on the Leonard-Jones site lead to both more efficient performance and a slight variation in properties such as the radial distribution function. As shown in figure 3.4 our model as crafted by the trigonometric function mentioned in figure 3.3 has a single Leonard-Jones wall represented by the dark blue wall, three attractive sites shown in light blue, three repulsive sites shown in orange, as well as an ideal Hydrogen bonding distance

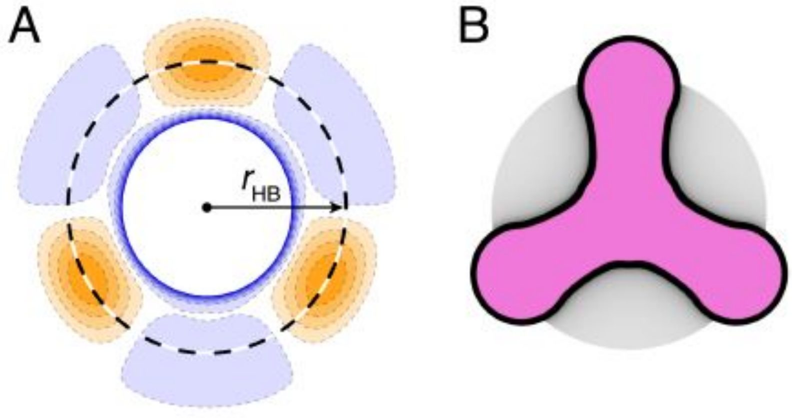

Figure 3.4: Shown is a contour plot of our finalized rose potential. The larger blue shaded regions are areas of positive interaction, while the smaller orange regions represent areas of repulsive interaction. The label of $r_{HB}$ shows the ideal length of a Hydrogen bond. The innermost solid blue line shown represents our Leonard-Jones boundry wall. On the right is the same contour plot rendered using the program "POV-Ray" where the larger pink lobes are attractive and the smaller gray areas are repulsive.

denoted by $r_{HB}$. The figure shown in pink accompanying this potential contour plot was rendered using a set of free image tools knows as POV-Ray. This all pink model was chosen to be the representation of our rose model and will be shown as such in all following images. When working with molecular dynamics large trajectory files are created describing every particle's path throughout the simulation. These trajectory files lend themselves to a relatively simple conversion into POV-Ray readable files commonly called ".pov" files. These files contain necessary header information such as camera, and light source settings as well as a list of every rendered object in some coordinate form. Our 2D trajectories already contain full listings of every particle's position in the x and y direction meaning their conversion to a 2D image is relatively simple.

One of the primary goals of this project was to capture the anomalous feature of experimental waters tendency to reach a density maximum after melting. This feature is unique to water and does not appear in any other known liquids. Being one of the main goals of this project much of the preliminary model calibrations were determined by this metric. As shown below rose potentials were created with varying $a_1$, and $a_2$ coefficients as mentioned in figure 3.3. These values were held to the constraint of:

$$|a_1| + |a_2| = 1 \tag{3.20}$$

This constraint was chosen to preserve the $\epsilon_{HB}$ value at 1.2 in order to maintain the Mercedes-Benz characteristic of having a Hydrogen bonding strength that is ten times the dispersion strength while not altering the Leonard-Jones parameters. While $\epsilon_{HB}$ was used to adjust the depth of the trigonometric equation and thus the attractiveness of the function the full width at half maximum parameter was used to alter the radial shape of the models wells. This previously mentioned $r_{FWHM}$ parameter is defined by the length of the ideal Hydrogen bond and the upper/lower limits of attractiveness mentioned in switching function for the distance dependent potential. In order to closely approximate this anomalous density maximum, the aforementioned parameter of $r_{FWHM}$ was probed by crafting independent models with varying values for $r_{FWHM}$. As shown in figure 3.5 a final value of $6/12 \cdot r_{HB}$ was chosen as it was the closest approximation to the Mercedes-Benz model with respect to the density maximum, represented by a dip in the volume per particle as a function of temperature.

While the desired dip in volume does occur at similar temperature in our model near the temperature where the MB model experienced a dip our model appears to have an overall lower melting temperature. This was likely an artifact of the scanning process due to runs with a higher leading coefficient being generated by the final frame of a previously run simulation. If, for example, the simulation for $2/12 \cdot r_{HB}$ was generated from the final frame of the run for $1/12 \cdot r_{HB}$ the starting state of the 2/12 run would be in a configuration matching higher temperatures leading to a disordered state. This higher temperature configuration was then equilibrated to a lower, initial temperature and supercooled. Leading to our apparent lower melting temperature. As shown in figure 3.6 the melting temperature of our model does

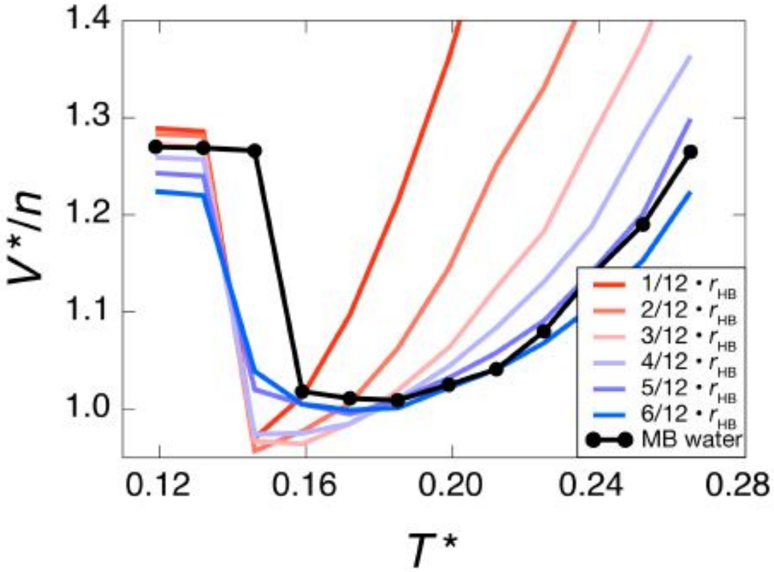

Figure 3.5: Shown are a sequence of simulations used to help determine the appropriate settings for the $r_{HB}$ parameter that modifies the radial shape of our model. The parameter setting of $6/12 \cdot r_{HB}$ was chosen as it most closely matched the volume dip in Mercedes-Benz water.

match the melting temperature of the MB model when a simulation is created from an initial, crystalline, state and the temperature is increased.

With the help of POV-Ray simulations of our simple 2D model can be used to create images that allow the user to quickly see the behavior of their system by generating a jpeg file for each frame, and even looping those jpeg files together to create a gif if desired. As shown in figure 3.6 a final model was ran in a simulation with increasing temperature using the NPT ensemble for each simulation. This model was crafted to approximate the temperature of maximum density by setting the Leonard-Jones minimum equal to $r_{HB}$. This figure shows our system in three visually distinct states. An initial, crystalline structure at a reduced temperature of 0.12. A collapsed, aqueous system at the temperature of maximum density at 0.17, and fully gaseous system at a reduced temperature of 0.26.

As shown in figure 3.7, the radial distribution function of our Rose model was similar to the MB model in that the peaks and troughs appeared at similar distances from the original particle. Meaning that the two systems organize in similar manners

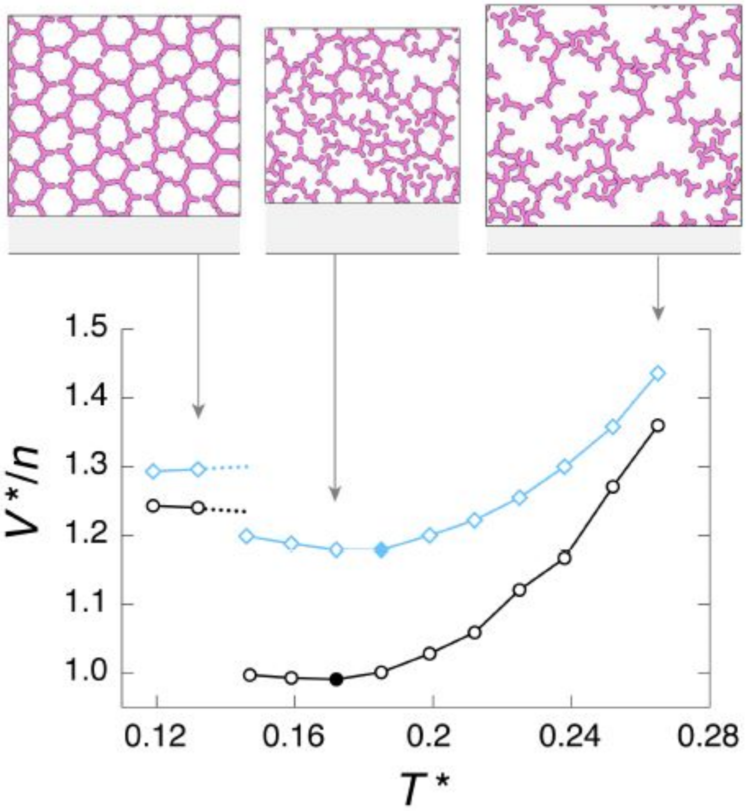

Figure 3.6: Shown is a plot of Volume per particle of the finalized Rose Potential model and how it compares to the same data for the Mercedes-Benz model.

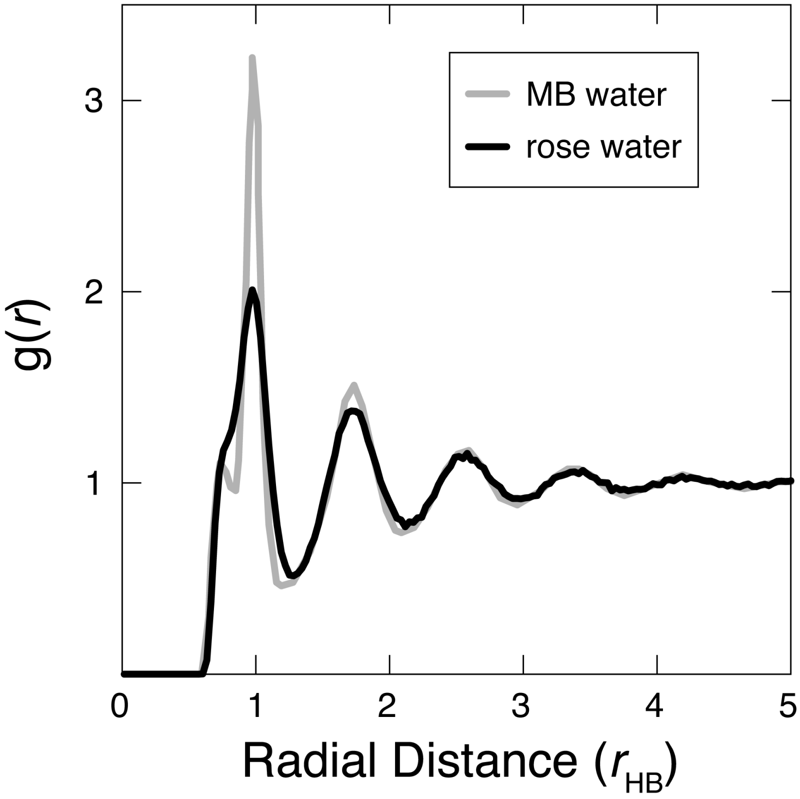

Figure 3.7: Shown is the radial distribution function for both the Rose model as well as the Mercedes-Benz model. These two models behaved simularly, but our final Rose model had less intense peaks due to its ability to form Hydrogen bonds at angles that the Mercedes-Benz model would not allow, leading to a system with more disorder.

while in a disordered state. A key difference between the two models is that our model had less intense peaks over-all, coming from our models tendency to make Hydrogen bonds at angles that the MB model would not accept. This promiscuity between Hydrogen bonds allows for a more disordered system where the shells of atoms are less defined when viewed from an original point, leading to less intense peaks in the radial distribution function.

### 3.6 Thermodynamic Data

Because our model possessed similar structural and energetic properties of MB water it was a likely to match previously obtained thermodynamic data from MB calculations as well. Shown in figure 3.8 are plots of thermal expansion, isothermal compressibility, and heat capacity. When comparing the thermal expansion both models show an initial negative slope upon heating, meaning that as previously mentioned the systems initially contract upon heating until a maximum density is reached. Once both systems reach a maximum density they then slowly expand with a positive slope until a fully gaseous state is reached. This final positive portion of thermal expansion does vary slightly in our model near the end of heating where the slope becomes negative. This is due to having a $\sigma_{\mathrm{LJ}}$ value that is smaller than $r_{HB}$. The graphs of isothermal compressibility and heat capacity do show agreement with that of the MB model. However, when viewing heat capacity, the rose model possesses a higher capacity once it reaches the solid state. This is likely due to the models ability to allow hydrogen bonds at greater angle differences than its MB counterpart. Allowing for this greater degree of freedom will enable the system to accept more energy per degree of temperature increase.

### 3.7 Dynamical Data

With this new model embedded within our molecular dynamics engine we could not only calculate the previously mentioned thermodynamic data, but also dynamic data derived from the simulated path taken by each particle. The two dynamic properties chosen were coefficient of diffusion and orientational-relaxation times. Both of these properties were size dependent and as such were calculated at five different system sizes: 144, 225, 324, 441, and 576 particles per system. Each diffusion and relaxation value from these systems were then used to extrapolate to infinite system sizes. Shown in figure 3.9 are the data points for coefficient of diffusion and orientational relaxation time at infinite system sizes as a function of temperature. While acquiring these data sets was not a primary goal of this project this data can be used to confirm or reject our models approximation of experimental water. Like previous data the coefficient of diffusion was recast in terms of Hydrogen bond length, or $r_{HB} = 3.8$ Å. When

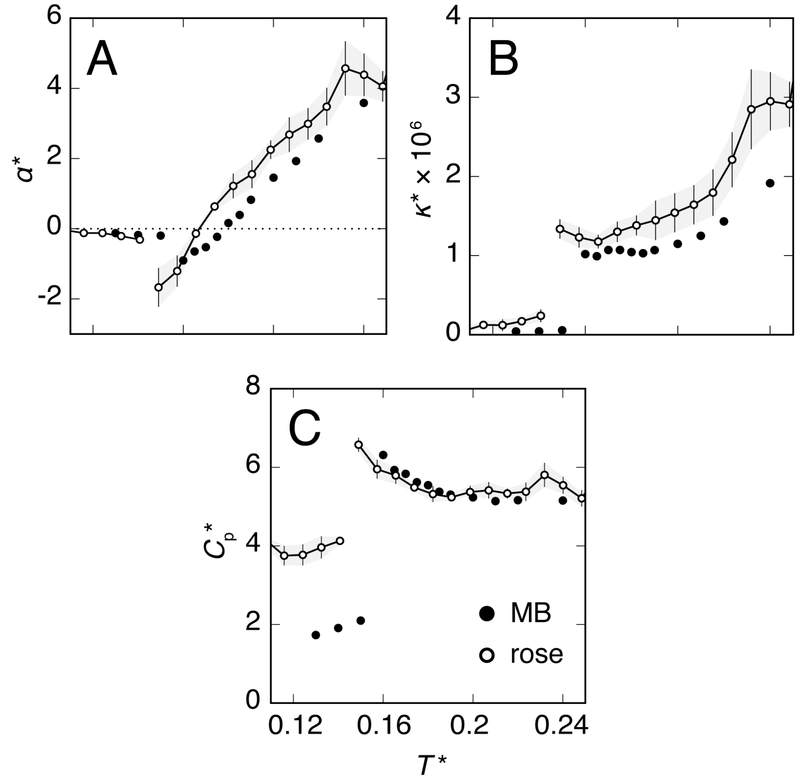

Figure 3.8: This figure shows plots of: (A) coefficient of thermal expansion, (B) isothermal compressibility, and (C) heat capacity. This data shows that our model is comparable to the Mercedes-Benz model in most areas, with the exception of a drop off in thermal expansion at high temperatures, and an initially higher heat capacity. When compared to the Mercedes-Benz model our model had a significantly lower Leonard-Jones value, which leads to the difference upon thermal expansion. As well as the ability to form partial Hydrogen bonds which leads to our model being able to absorb more energy per degree of temperature increase in the system. This ability to absorb more energy leads to having an apparent heat capacity that is higher than the original Mercedes-Benz model.

comparing diffusion constants given by our model with experimental results of water at the same temperature our model shows a higher coefficient of diffusion, likely due to its reduced dimensionality and therefore higher amount of free volume. The free volume for a simulation of our final Rose model at a reduced temperature of 0.183 was calculated to be near 1.0 while the free volume for experimental water would be 0.73 for the same temperature. This higher free volume would lead to more space for the particles to move through and therefore a higher coefficient of diffusion. The second calculated dynamical property was Hydrogen-bond relaxation time. This property quantifies how quickly a Hydrogen bond can be reformed after it has been broken. In simulations using our model the relaxation time was found to be 21.7 ps at the reduced temperature of 0.183, this is markedly higher than the known experimental value of 4.8 ps for the same temperature. This longer relaxation time is likely due to the stickiness of our models Hydrogen bonds which can also be seen in the previous figure 3.6 where the rose particles tend to form clusters upon their crystalline collapse.

## 3.8 Final Conclusions

While our 2D approximation of the Mercedes-Benz model was not a perfect match for all thermodynamic properties it did show reasonable similarities to its Monte-Carlo counterpart. This new model is simple, versatile, and can be run efficiently using a freely available molecular dynamics engine. This model could be easily tuned to accurately represent known dynamical properties of experimental water so that further probing can be done by running new simulations. Future plans for this project included using this model to represent larger systems with unique features such as a channel that will allow accelerated diffusion, and the probing of four and five petal systems.

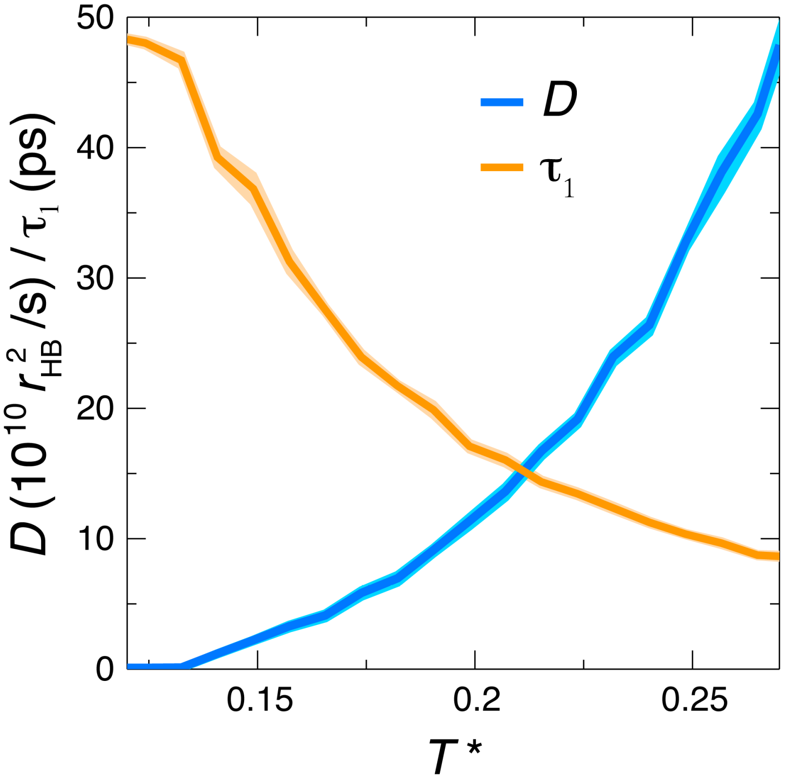

Figure 3.9: This figure shows a combined plot of Coefficient of Diffusion and Hydrogen-Bond relaxation time. Our model has a considerably higher coefficient of diffusion when compared with experimental water due to its excess of free volume. Our model also has a longer relaxation time due to the ”stickiness” of our modeled Hydrogen bonds.

# CHAPTER 4

# Diffusion within Channels

## 4.1 Introduction and Literature Review

As mentioned in the introduction one goal of our research was to use this simple 2D system to investigate accelerated diffusion within channels. This follows as a simple off-shoot of the previous work due to the availability and validity of our 2D model and previously created systems. The topic of accelerated diffusion within channels is a commonly discussed topic in the scientific community partly due to the new found feasibility of the construction of the micro channel environments by application of carbon nanotubes. As discussed by Hummer in "letters in nature" water particles seem to counter-intuitively fill the space within nano-tubes. [18] In their simulations water particles entered a channel with a diameter of 8.1 Å , and quickly passed through the opposite end of the nanotube. These simulations were carried out in three dimensions leading to the channel size 8.1 Å being large enough to only allowed a single chain of Hydrogen bonded particles. Although energy is lost through the breaking of chemical bonds as particles enter the channel, and little to no energy is gained due to the bonding of polar water particles to the non-polar carbon nanotube the particles spontaneously filled the nanotube and passed through its confines. These particles were then shown to have accelerated diffusion in the axis of the length of the cylinder. In order to see how the nanotubes would fill if the carbon-water attractive forces were weaker the depth of the van der Waals potential was lowered by 0.05 $\mathrm{kcal\,mol^{-1}}$. Although no hard data was given as to how this change effected the rate of diffusion the rate at which the nanotube fluctuated between an empty state (containing no particles) and a filled state (containing a number of particles) did sharply increase. With this knowledge we sought out a way to build upon these studies in our simplified 2-dimensional systems and obtain diffusion data as a function of channel size as well as repulsiveness of our ”channel” object. With this large span of variables we will hopefully be able to steer experimental work in the appropriate direction with the knowledge of most efficient channel size, and repulsiveness.

### 4.2 Methodologies and Representations

As described by Hummer the goal of their, and similarly our simulations, would be to closely represent how these carbon nanotubes behave in laboratory situations using molecular dynamics. With the goal of obtaining diffusion data for particles within a channel, we were required to simulate larger systems for longer run times. This is due to the fact that there are relatively few particles contained within the channel at any given time, meaning that it would be difficult to obtain statistical data for these particles. The systems dimensions were not directly given to OpenMD, but rather the total number of particles per simulation were chosen so that the engine could chose and maintain an appropriate box size that allowed the simulations be carried out using the NPT ensemble. Systems were created having the numbers of particles: 625, 1225, 1600, and 2025. This allowed the engine to craft a box size for each simulation, where the average the box sizes were approximately 100 nm. With appropriately sized systems our next goal was to insert unfilled channels into these systems and allow the simulations to progress as naturally as possible so that a coefficient of diffusion could be found. Before any work could be done on creating these systems I first needed to create an immovable channel like object that could be fine-tuned to be attractive or repulsive to the water particles. As previously mentioned in the introduction one of the motivating factors for our group using the OpenMD engine was its ability to define rigid-body objects. These objects can be made of many particles that are inserted into the initial trajectory once, and are then defined by the engines metadata files to behave like a single, large particle. The ability to define these rigid body objects was taken advantage of in this project by defining a molecule of type WALL that consisted of several atoms known to OpenMD as the Heavy type. Meaning that once our wall was defined it could be placed in the trajectory a single time, and act as a type of channel due to the relatively low momentum of the surrounding water particles. The chosen method of defining the channel length was to dynamically chosen channel size to be 80% of the total simulated box length in the $x$ dimension. Because the simulations were run using periodic boundary conditions we were not able to simply set the channel length equal to the simulation box length in the $x$ dimension as this would lead to an apparent channel size of infinity and no particles would be able to enter the channel. While maximum channel length would lead to more particles within the channel and therefore more statistics with which to calculate the diffusion it did lead to problems when setting up directories to carry out these simulations.

With the channel length being set dynamically many extra meta data files had to be edited, leading to more possible errors and greater difficulty in scaling up these simulations to obtain useful data. Another solution to crafting these systems would be to place a channel, of a set length, in already populated simulations, directly over existing water particles. In order to keep the repulsive forces of our overlapped particles from shooting the particles in a direction with a very high velocity the particles would be progressed through very fine grain time steps and their velocities reset to zero until all particles were a sufficient distance away from the newly placed channel object. With a properly insert wall object simulations can be run using the NPT ensemble with increasing temperatures. In order to properly probe the attractive/repulsive characteristic of the channel the simulations would also need to be ran with varying Lennard-Jones parameters for the channel like particle. It should be noted, at this point there is no numerical data available for diffusion within channels, but both solutions previously mentioned are readily available to any future calculations. As well as a simple python program that can be used to calculate diffusion in both the $x$ and $y$ axes independently. While no numerical data is available for this project several gifs have been made to visually explore what happens in these channel like environments. In figure 4.1 and 4.2 images shown from gifs of these channel systems with a channel of 7 Å and 16 Å allowing for a single chain of Hydrogen-bonded particles as well as small clusters of tightly packed particles.

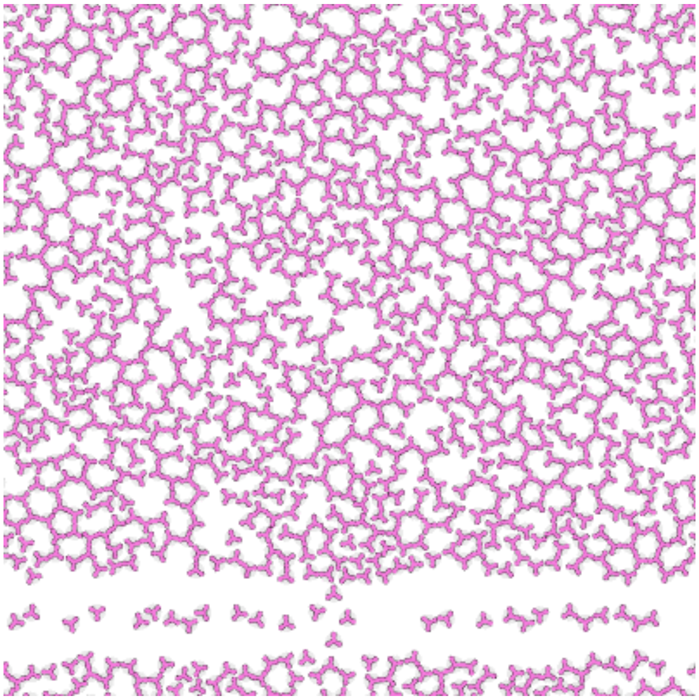

Figure 4.1: Shown is a 7 Å wide channel containing water molecules undergoing apparent accelerated diffusion. Due to the relatively small channel size particles are only allowed to flow through the channel in chain like structures.

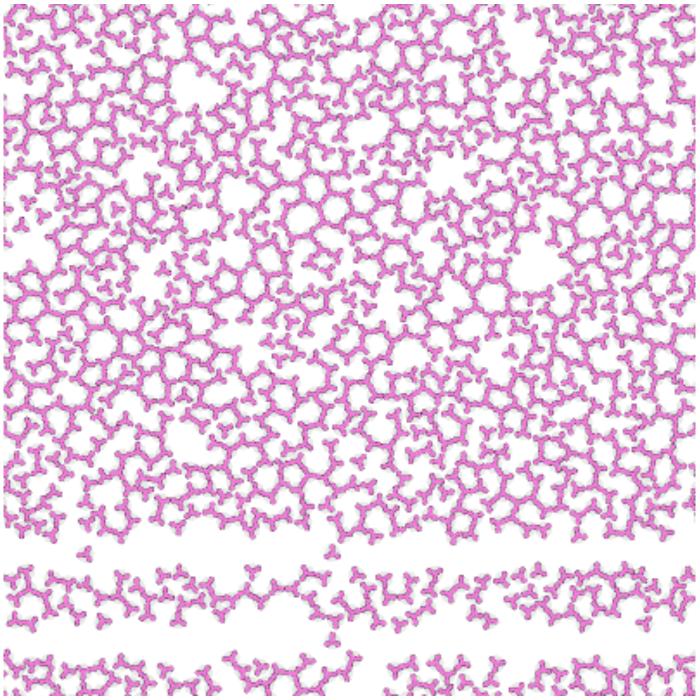

Figure 4.2: Shown is a 16 Å wide channel containing water molecules undergoing apparent accelerated diffusion. Due to this larger channel size particles are allowed to "clump" together while in the channel, because of this the apparent accelerated diffusion is lessened.

# CHAPTER 5

# Nucleation Events

## 5.1 Introduction and Literature Review

While in some respects nucleation is an easily observable physical phenomenon it is also a minutely detailed chemical phenomenon that is often controversially discussed in scientific circles. As discussed by Amir Haji-Akbari and Pablo Debenedetti the process of nucleation is broken up into two categories, one being the nucleation that occurs on a free interface in a clear crystallization process. [19] This form of nucleation can be proven experimentally by verifying the existence of a crystalline region on the interface in systems where the bulk temperature is just above the freezing temperature of that system. The other form of nucleation is identified by an event that is driven by purely kinetic forces at or near the interface that does not leave a clearly defined crystalline region due to an inconsistent interface such as one that can be found at a moving liquid-vapor interface of water. This second, not easily observable, nucleation event leads to it being a phenomenon that is a great candidate for molecular dynamics simulations. Work done by Imre Bako investigated nucleation through the viewing of hydrogen bond path lengths. [20] Specifically, these simulations dealt with liquid-methanol mixtures in attempts to classify the branching of Hydrogen bonds. It would follow that a longer a Hydrogen bond path length, indicates more organized Hydrogen bonds are being formed. With the creation of organized Hydrogen bonds we can infer that a crystalline state has started to form meaning that we have started a nucleation process. While these simulations did shed light on the overall path length of the longest Hydrogen bond network, as well as a ring size distribution at temperatures of 250, 300, and 350 K it did not give a full classification of ring sizes which could be used to shed more light onto the topic of nucleation. In a simulation being run from a high temperature to a temperature below the freezing point of water it can clearly be seen that these systems tend to form six-membered rings as shown in the 2D approximation of figure 1.1. While we know that some form of nucleation event has occurred to produce these six-membered rings we do not know exactly how this process is carried out. This vacuum of information lead to the final project of identification of nucleation events via the shift in the distribution of ring sizes.

## 5.2 Methodologies and Findings

Our primary metric with which to judge nucleation events is the formation of multiple bonds causing a ring of three or more water molecules. A preliminary study was done with the previously crafted rose model to help identify ring forming events within a 2-dimensional environment. Although these calculations were carried out and yielded some precursory data the formation of rings in our simple 2D models was difficult to compare with more realistic 3D simulations and experimental data due to an inability to alter the rate at which these rings form by changing the moment of inertia in the z-axis. Since our 2D model was of little help in these investigations our attention was turned towards the slightly more computationally costly 3D models. With this increase in dimension there was an obvious increase in computations to be carried out. This increase in the number of computations meant that I could no longer rely on previously written python code as it is an uncompiled language and tends to run more slowly than its' compiled counterparts. Instead new code was written in $C^{++}$ which gave our programs the desired efficiency and will be of more use to future projects as many of the crafted modules within this code can be easily pulled out and reused. This new code was used to help define all Hydrogen bonds within our simulations by geometric means. The maximum distance between bonded Oxygen atoms was set

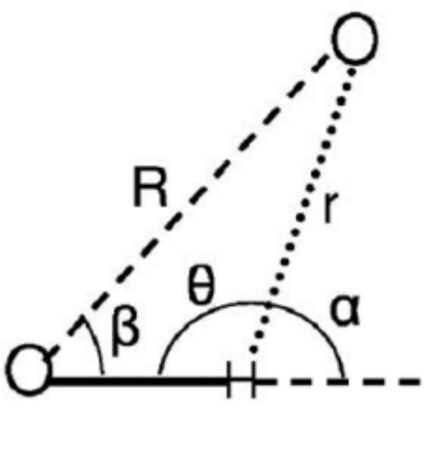

Figure 5.1: Illustration of geometric requirements for a Hydrogen bond.

to 0.35 nanometers, and the maximum angle formed between the O-H–O bond was required to be less than 30°. These bonding criterion were discussed by Kumar and Schmidt, and visually represented figure 5.1 with our distance criteria labeled R and our angle criteria labeled $\beta$. [21]

While other definitions of Hydrogen bonds have been used and are readily available for the sake of computational efficiency we chose the most simple and direct route. [21] In order for this program to properly work and not over or under count the number of rings every possible O-H–O angle must be calculated for any particles with a distance less than the previously mentioned distance tolerance. This portion of the calculations were completed relatively quickly and it was clearly evident, even in small initial cases, that the time-consuming step would be to navigate through these lists and enumerate every ring within the system. An example of this navigation on a simple 4-member ring will show why this is the rate determining step:

```
Particle 0 has neighbors [1,3]

Particle 1 (the first neighbor of 0) has neighbors [2,0]

Particle 2 (the first neighbor of 1) has neighbors [3,1]

Particle 3 (the first neighbor of 2) has neighbors [0,2]

Stop when a neighbor matches the original particle of 0
```

Even in this simple example, where the particles naturally progressed through the small ring by not having to check secondary neighbors would be a costly calculation when carried out on large scales. This complexity grows as the size of neighbor lists, and fruitless neighbor checks expand.

After a complete list of rings has counted and saved into arrays, a check for smaller rings contained within larger rings had to carried out in order to avoid counting any ring twice. For the purposes of our project the smaller ring was always chosen over the larger one. Meaning if all members of a three-member ring were contained within a listing of eight-member ring the eight-member ring was thrown out and the three-member ring would be kept. With a complete and accurate list of all rings within any given system distributions of ring sizes can be given and the predominance of ring size at a given temperature can be shown. Becuase our calculation required saved arrays containing every ring a simple module was made and incorporated into the code to output Pov-Ray files of the analyzed simulations. Our code has the ability to generate several versions of visual output. The first of which being a complete, and messy, representation of every ring in the system represented by a number of triangles. As an example, if wished you created a 2 dimensional square containing particles 1,2,3,4. You would first create a triangle containing particles 1,2,3 and then create a triangle containing particles 1,3,4. This leads to the necessity of creating a number of triangles equal to the ring size minus 2. This ”full” visualization is not advised, and therefore not included, as Pov-Ray will render any ring it is given. This leads to overlapping rings anytime two rings intersect in our 2D viewing plane. Which leads to unhelpful visual output. The second visualization option is a modified version of the first where a portion of the z-axis is trimmed from the frame after the simulation and ring calculations have been carried out leading to a slightly less cluttered view. This code allows for the representation of any single (or combination) of ring sizes to be viewed, which can be useful when trying to gain insight into areas of density for individual ring sizes. The final and most useful output is a version where all rings are represented by spheres of varying sizes in the average location of the ring. This leads to a representation that can be used to quickly identify the location, and method by which nucleation events occur. As referenced in the channel diffusion section these images could easily be generated for every frame of a trajectory and looped together to form a gif if desired. These looped images could then be used to gain insight into how nucleation events proceed via the types of rings being formed.

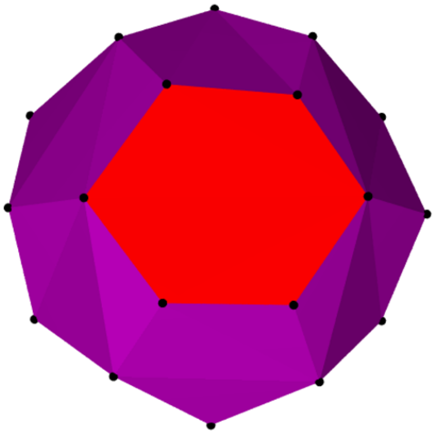

Figure 5.2: Pov-Ray rendered image using the triangle method with a Clathrate-1 Hydrate.

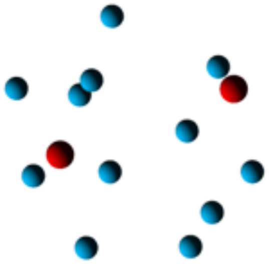

Figure 5.3: Pov-Ray rendered image using the sphere method with a Clathrate-1 Hydrate.

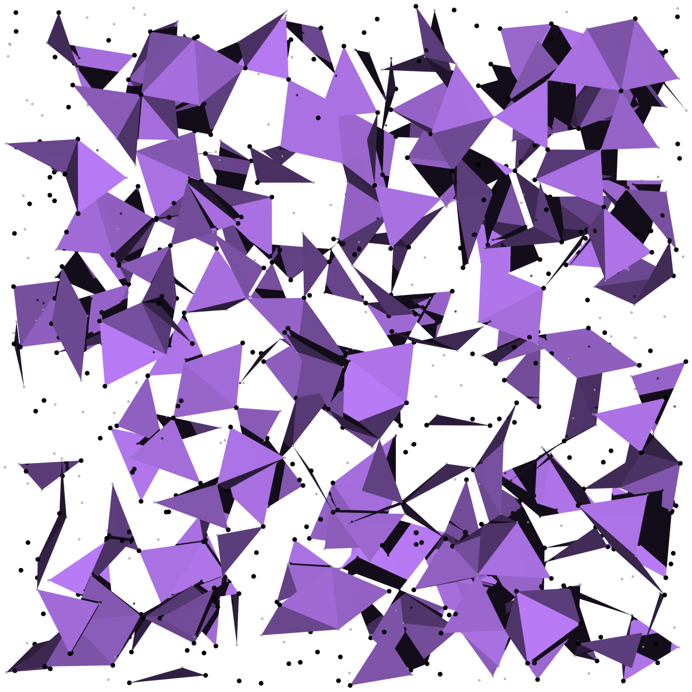

Figure 5.4: Pov-Ray rendered image using the triangle method with only three member rings represented.

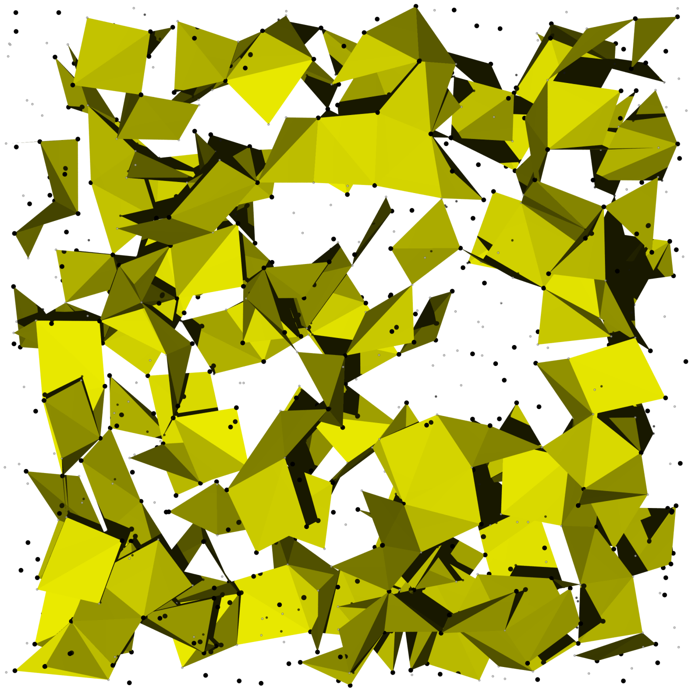

Figure 5.5: Pov-Ray rendered image using the triangle method with only four member rings represented.

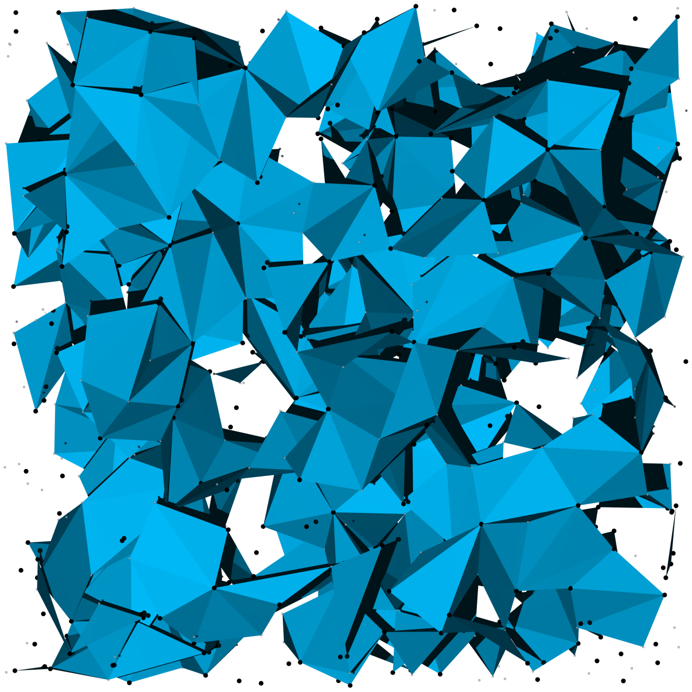

Figure 5.6: Pov-Ray rendered image using the triangle method with only five member rings represented.

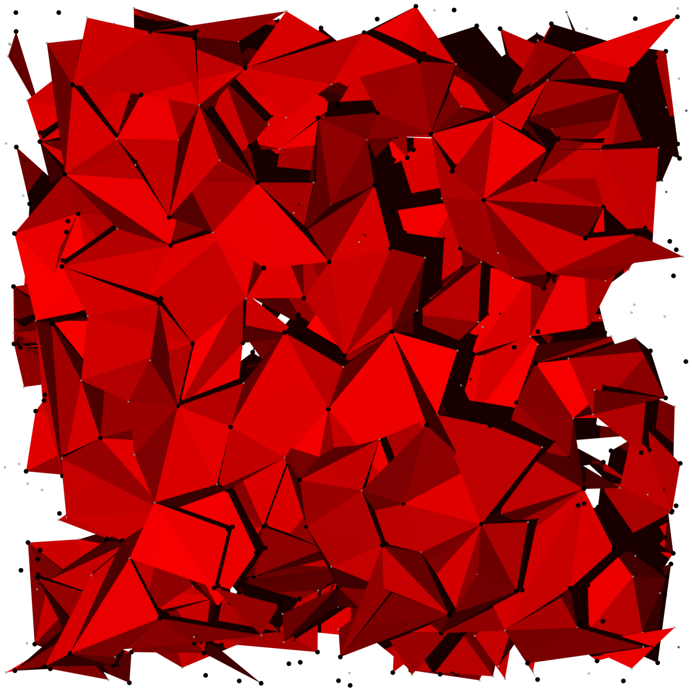

Figure 5.7: Pov-Ray rendered image using the triangle method with only six member rings represented.

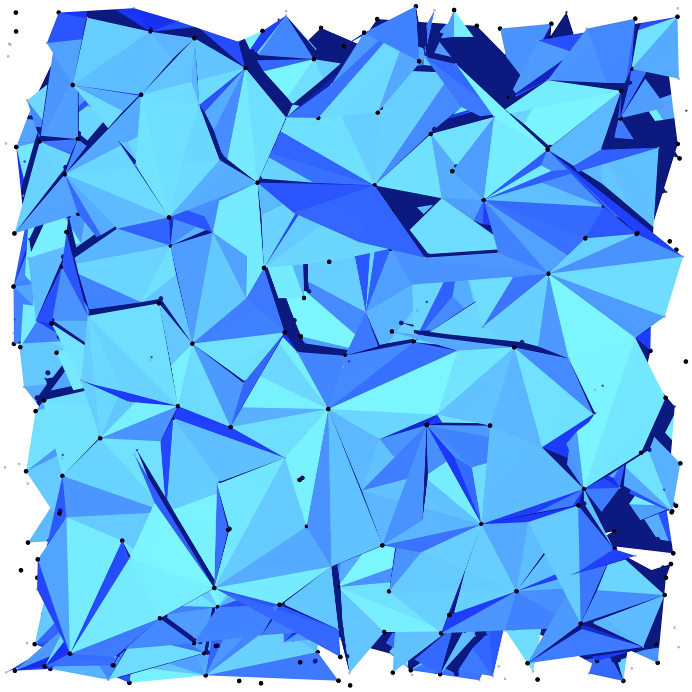

Figure 5.8: Pov-Ray rendered image using the triangle method with only seven member rings represented.

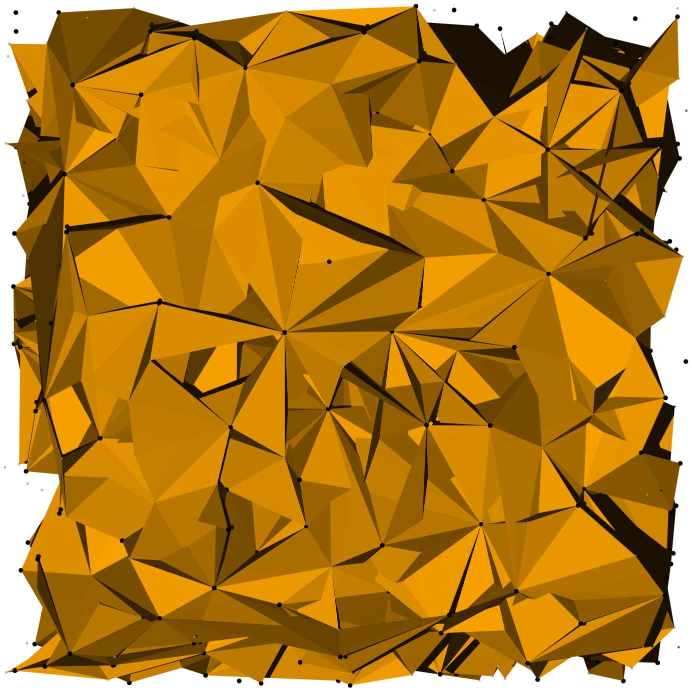

Figure 5.9: Pov-Ray rendered image using the triangle method with only eight member rings represented.

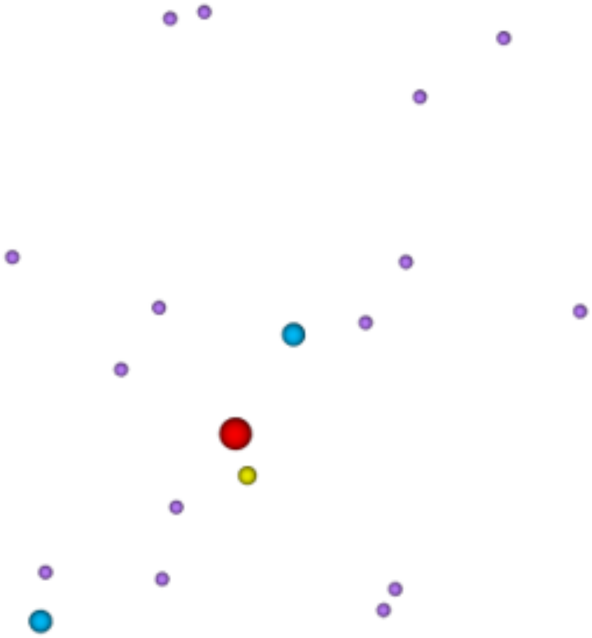

Figure 5.10: Pov-Ray rendered image using the sphere method with 300 particles represented.

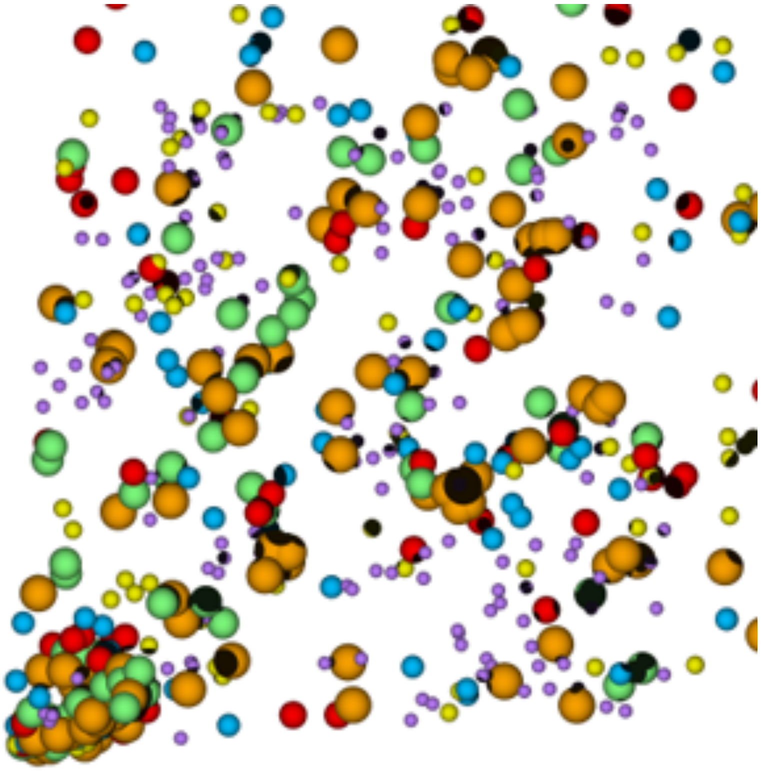

Figure 5.11: Pov-Ray rendered image using the sphere method with 600 particles represented.

## References

[1] K. A. Silverstein, K. A. Dill and A. D. J. Haymet, Hydrophobicity in a simple model of water: Solvation and hydrogen bond energies. *Fluid Phase Equilibria*, 150: 83–90 (1998).

[2] H. J. C. Berendsen, D. van der Spoel and R. van Drunen, Gromacs: A message-passing parallel molecular dynamics implementation. *Computer Physics Communications*, 91: 43–56 (1995).

[3] M. A. Meineke, M. A. Vardeman, T. Lin, C. J. Fennell. and J. Gezelter, Oopse an open source object-oriented parallel simulation engine for molecular dynamics. *Journal of Computational Chemistry*, 26: 252–271 (2010).

[4] A. Ben-Naim, One-dimensional model for water and aqueous solutions. ii. solvation of inset solutes in water. *Journal of Chemical Physics*, 128: 024506 (2008).

[5] A. Ben-Naim, One-dimensional model for water and aqueous solutions. iii. solvation of hard rods in aqueous mixtures. *Journal of Chemical Physics*, 129: 104506 (2008).

[6] A. Ben-Naim, One-dimensional model for water and aqueous solutions. iv. a study of "hydrophobic interactions". *Journal of Chemical Physics*, 129: 104506 (2008).

[7] A. Ben-Naim, Statistical mechanics of "waterlike" particles in two dimensions. i. physical model and application of the percus-yevick equation. *Journal of Chemical Physics*, 54: 3682–3695 (1971).

[8] A. Ben-Naim, Statistical mechanics of “waterlike” particles in two dimensions. ii. one component system. *Molecular Physics*, 24: 705–721 (1972).

[9] K. A. Silverstein, A. D. J. Haymet and K. A. Dill, A simple model of water and hydrophobic effect. *Journal of American Chemical Society*, 120: 3166–3175 (1998).

[10] K. A. Silverstein, A. D. J. Haymet and K. A. Dill, Molecular model of hydrophobic solvation. *Journal of Chemical Physics*, 111: 8000—8009 (1999).

[11] N. T. Southall and K. A. Dill, The mechanism of hydrophovic solvation depends on solute radius. *Journal of Physical Chemistry*, 104: 1326–1331 (2000).

[12] N. T. Southall, K. A. Dill and A. D. J. Haymet, A view of the hydrophobic effect. *Journal of Physical Chemistry*, 106: 521–533 (2002).

[13] B. N. Hribar, N. T. Southall, V. Vlachy and K. A. Dill, How ions affect the structure of water. *Journal of American Chemical Society*, 124: 12302–12311 (2002).

[14] T. Urbic, V. Vlachy and K. A. Dill, Confined water: A mercedes-benz model study. *Journal of Physical Chemistry*, 110: 4963–4970 (2006).

[15] C. H. Williamson, C. J. Fennell and J. R. Hall, Two-dimensional molecular simulations using rose potentials. *Journal of Molecular Liquids*, 00: 1–13 (2016).

[16] M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids*. Oxford University Press, first edition (1989).

[17] I.-C. Yeh and G. Hummer, System-size dependence of diffusion coefficients and viscosities from molecular dynamics simulations with periodic boundry conditions. *The Journal of Physical Chemistry A*, 108: 15873–15879 (2004).

[18] G. Hummer, J. Rasaiah and J. P. Noworyta, Water conduction through the hydrophobic channel of a carbon nanotube. *Letters to Nature*, 414: 188–109 (2001).

[19] A. Haji-Akbari and P. G. Debenedetti, Perspective: Surface freezing in water: A nexus of experiments and simulations. *The Journal of Chemical Physics*, 147 (2017).

[20] I. Bako, A. Bencsura, K. Hermannson, S. Balint, T. Grosz, V. Chihaia and J. Olah, Hydrogen bond network topology in liquid water and methanol: a graph theory approach. *Physical Chemistry Chemical Physics*, 15: 15163–15171 (2013).

[21] R. Kumar and J. R. Schmidt, Hydrogen bonding definitions and dynamics in liquid “waterlike”. *The Journal of Chemical Physics*, 126: 204107:1–204107:12 (2007).

# APPENDIX A

## Python code

### A.1 Diffusion and Relaxation time

The following code is a small section of python code used in the determination of coefficient of diffusion. Although not shown earlier portions of the code build arrays consisting of the $x$ and $y$ position of every Oxygen atom within the trajectory. These arrays are listed as "Oxarray" and "Oyarray". The variable of "molenum" represent the total number of constant particles used a this simulation. The "finalarray" represents an array built from average distance each particle has traveled in given timestep where timesteps range from their smallest size to the largest possible time step.

```
for j in range(0,frames):
    n = 0
    while (n+(molenum*interval) < len(Oxarray)):
        for r in range(0,molenum):
            x = (pow((Oxarray[n]-Oxarray[(n+(molenum*interval))]),2)+
                   (pow((Oyarray[n]-Oyarray[(n+(molenum*interval))]),2)))
            n += 1
            mean.append(x)
    finalarray.append(sum(mean)/len(mean))
    interval += 1
    mean = []
```

This section of python code is used to determine the linear regression fit of the "finalarray" created in the above section of code. As a note only a portion the data is used as much of the data will not form a linear region. The statements of n = int(.1*(len(finalarray))), and (int(.5(len(finalarray)))) result in this scaling.

```
polyarraymsd = []
polyarrayt = []
n = int(.1*(len(finalarray)))
while n <= (int(.5*(len(finalarray)))):
    polyarraymsd.append(samparray[n])
    polyarrayt.append(finalarray[n])
    n += 1
p1 = polyfit (polyarraymsd,polyarrayt,1)
```

This portion of python code results in the total integration of relaxation data. Where "trapz" results in the analytic portion, and "curve_fit" results in the approximated portion.

```
while n <= (len(lines[0].split())-251):
    xarray.append(n*20)
    yarray.append(float(lines[0].split()[n]))
    n = n+1
resultanalyt=trapz(yarray,xarray)
n=1251
xarray=[]
yarray=[]

while n < (len(lines[0].split())-100):
    xarray.append(n*20)
    yarray.append(float(lines[0].split()[n]))
    n = n+1

def func(x,a,b,c):
    return(a*exp(-b*x)+c)
popt,pcov=curve_fit(func,xarray,yarray)
a=popt[0]
b=popt[1]
c=popt[2]

resultnum=0
resultnum=((a*exp(-b*25000))/b)+ (25000*c)

print(resultnum+resultanalyt)
```

# APPENDIX B

## C code

The following $C^{++}$ code wraps coordinates of particles that my have left the originally defined area of our simulation box. All of the simulations mentioned in this paper make use of periodic-boundary-conditions, causing this wrapping function to be necessary when any absolute distance is desired.

```
xVal2 -= boxVector[0]*copysign(1.0,xVal2)*
              floor(fabs(xVal2/boxVector[0])+0.5);
yVal2 -= boxVector[1]*copysign(1.0,yVal2)*
               floor(fabs(yVal2/boxVector[1])+0.5);
zVal2 -= boxVector[2]*copysign(1.0,zVal2)*
              floor(fabs(zVal2/boxVector[2])+0.5);
```

The following $C^{++}$ code is used to find the angle between an O-H–O bond. This angle is the angle formed when one water particle's Hydrogen has bonded to another water particle's Oxygen. Where the "xVal1" variable represents the $x$ distance difference between the Oxygen atom of one particle and one of the Hydrogen atoms of another particle. Similarly the "xVal2" variable represents the $x$ distance between two particle's Oxygen atoms.

```
double normba = sqrt((xVal1*xVal1)+(yVal1*yVal1)+(zVal1*zVal1));
double normbc = sqrt((xVal2*xVal2)+(yVal2*yVal2)+(zVal2*zVal2));
double ba[3] = {(xVal1/normba),(yVal1/normba),(zVal1/normba)};
double bc[3] = {(xVal2/normbc),(yVal2/normbc),(zVal2/normbc)};
double dotpdct = ba[0]*bc[0] + ba[1]*bc[1] + ba[2]*bc[2];
double anglecalced1 = (dotpdct);
anglecalced1 = acos(anglecalced1) * (180.0 / PI);
```

This section of $C^{++}$ code was used to check for larger rings that contained smaller rings. Meaning if a four member ring was listed as containing particles 1,2,3,4 and a three member was listed as containing particles 1,2,3 this function would return a value of "true".

```
bool isSubset(vector<int> (&arr1), vector<int> (&arr2), int m, int n){
  int i = 0;
  int j = 0;
  vector<int> tempvect;
  vector<int> tempvectcopy;
  tempvect = arr1;
  tempvectcopy = arr2;
  sort(tempvect.begin(), tempvect.end());
  sort(tempvectcopy.begin(), tempvectcopy.end());
  if (m < n)
    return 0;
  while (i < n && j < m ){
    if( tempvect[j] < tempvectcopy[i] )
      j++;
    else if( tempvect[j] == tempvectcopy[i] ){
      j++;
      i++;
    }
    else if( tempvect[j] > tempvectcopy[i] )
      return 0;
  }
  return  (i < n)? false : true;
}
```

# VITA

CASEY H. WILLIAMSON

Candidate for the Degree of

Master's of Science

Thesis: Computational Studies of 2D Diffusion and Water Nucleation

Major Field: Chemistry

Biographical:

Education:

Received a Bachelors of Science in Chemistry at East Central University in December 2015.

Completed the requirements for the degree of Master's of Science at Oklahoma State University in July 2010.

Experience:

While obtaining an undergraudate degree I was advised under Dr. Dwight Myers with a major in Chemistry and a minor in Mathematics. During this time I was an active member of the Honors Student Association and member of the McNari's Scholars program. As McNair Scholar I completed research into organic solar cells under the advisement of Dr. Dane Scott. This work was transitioned from a summer spent at Oklahoma State University under the advisement of Dr. Toby Nelson I synthesized poly-3-thiophere for the future use in solar cells. I received my B.S. in Chemistry from the East Central University in December of 2015 and transitioned into the Chemistry department of Oklahoma State University the following January under the advisement of Dr. Christopher Fennell. At Oklahoma State University I carried out multiple molecular dynamics calculations making use of the programming languages of python, and $C^{++}$. During this time at Oklahoma State University I received a publication in the Journal of Molecular Liquids with an article titled "Two-dimensional molecular simulations using rose potentials".