

MULTI-SCALE MODELING OF NANOSTRUCTURED TIGHT ROCKS

by

Gøran Brekke Svaland

THESIS
for the degree of
MASTER OF SCIENCE



Faculty of Mathematics and Natural Sciences
University of Oslo

May 2015

To my brother whom I admire for his courage and dedicated workaholicness
being an excavator operator. And to my parents who always have supported me
with their endless love and encouragement.

Acknowledgements

I have experienced so much joy and good times through my studies at the Faculty of Mathematics and Natural Sciences at the University of Oslo. I have participated in field trips and had training in rocket science at Andøya Rocket Range with Canadian students whom I befriended, and will stay in touch with for many years to come. In my last few years here, I have been living at Campus, in a student home called Blindern Studenterhjem, where I have met people from all fields of study, including my own. That place has shaped me, and will always bring out the best of memories.

My first teacher in natural sciences was Øivind Bangor, he taught us about Darwin's theory, genetics, electricity and showed us some pretty cool chemical reactions that were important for my choice to pursue my fascination of the nature. At Bø videregående, which is the Norwegian word for high school, I met the teacher that was going to be the most important person for my future choice in studying physics at the University. Dag Arvid Aarskog. Dag taught us about Newton's laws of motion, Albert Einstein's abstract thoughts, sound and waves, astrophysics and mathematics. He told us about the incredible theory of everything, an ultimate theory which goal is to explain all interactions of nature, even explain our existence. At the same time in high school, I got some extra lessons from the fantastic lecturer Arne Myhre at Høgskolen in Telemark. Arne Myhre taught me mathematics like trigonometry, integration methods and how to solve difference and differential equations. He also introduced me to the Schrödinger's wave equation, and explained that the electrons orbiting the nucleus does not follow perfect orbits, but is better explained by a probability density distribution that can be found from solving this complicated equation. Little did I know at that time, that I would encounter the same equation years later, and learn how to solve it.

Meteorology were the direction I picked out at first on the FAM-program (Physics, Astronomy and Meteorology) at the University of Oslo. I wanted to know how the climate system works, how heat and energy is transported in the atmosphere, and in the ocean. But I was introduced to a course in computational physics, lead by Morten Hjorth-Jensen, a professor in theoretical physics and computational quantum mechanics, who showed me the beauty and possibilities of computer science, almost every kind of new knowledge now-days is based on, or supported by heavy computational approximations. He introduced me to the computational physics group, known as CompPhys, where I have had office for two years, studying computational quantum mechanics, percolation, geophysics and molecular dynamics. All the way through my bachelor and masters degree, I have had the pleasure of working and exchange knowledge and thoughts with

Jostein Blyverket. I would like to thank you for the countless discussions on the topics that we have encountered. I would also like to thank my supervisor, Anders Malthe-Sørensen, who have guided and inspired me in my study.

I would like to thank the people that develop and publish *open source* software, like *LATEX*, *Python*, *moltemplate*, *Ovito*, *VMD*, *Inkscape* and *Linux*. This is software that have been extremely important in my everyday work, and many others with me. I would like to give a special thanks to the developers of the software *LAMMPS*[43], that have been used extensively to run molecular dynamics simulations in this thesis. These guys have used years to develop three hundred thousand lines of code that works extremely well, and that gains more and more interest among computational physicists who would like to run advanced molecular dynamics simulations.

Collaboration details

On my thesis, I have been working closely with Jostein Blyverket. We have collaborated on setting up the system, and working on theoretical concepts.

Contents

1	Introduction to the thesis	5
1.1	Modern molecular dynamics studies of material properties	11
1.2	Ethics concerning the climate	12
1.3	Goals	14
1.4	Structure of the thesis	15
I	Theory and methods in molecular dynamics	17
2	Introduction to molecular dynamics	19
2.1	Time-integration	21
2.1.1	Velocity Verlet algorithm	22
2.2	Minimum image convention	23
2.3	Ensembles in thermodynamics	25
2.3.1	Microcanonical ensemble	26
2.3.2	Canonical ensemble	27
2.3.3	Grand canonical ensemble	27
2.4	Potentials	28
2.4.1	Van der Waals forces and the Lennard-Jones potential	28

2.4.2	Coulomb interaction	30
2.4.3	Long range Coulomb interactions	30
2.4.4	Bond potential	31
2.4.5	Angle potential	31
2.4.6	The Morse potential	32
2.4.7	The Tersoff potential	33
2.4.8	ReaxFF	33
2.4.9	Using ReaxFF in <i>LAMMPS</i>	36
2.5	Thermostats - controlling the temperature	36
2.5.1	The Berendsen thermostat	38
2.5.2	The Nosé-Hoover thermostat	38
2.6	Adsorption	39
2.7	Filling a nanopore	43
2.8	Diffusion	44
2.8.1	Calculating the diffusion	47
2.9	Tetrahedral order parameter	49
2.10	Radial distribution functions	51
2.10.1	Angular distribution	52
3	Software and <i>LAMMPS</i> specifics	55
3.1	The <i>LAMMPS</i> software	55
3.1.1	How to make <i>LAMMPS</i> executable	56
3.2	The input script	57
3.2.1	The header	57
3.2.2	Computes	58

3.2.3	Run time	59
3.2.4	Minimization procedures	60
3.3	Building molecular structures using moltemplate	62
3.4	Visualization	65
3.4.1	<i>OVITO</i>	65
3.4.2	<i>VMD</i>	66
II	Simulations	67
4	Benchmarking	69
4.1	Benchmarking, reproducing natural behavior	69
4.2	Bulk water	70
4.3	Bulk carbon dioxide	72
4.4	Portlandite	74
5	Quarz surface energy and binding energy of CO_2	79
5.1	Studied systems of quartz	79
5.1.1	Surface energy	81
5.1.2	Sequestration of carbon dioxide	81
5.1.3	Density functional theory	82
5.2	Estimates on binding energies and surface energies of quartz systems	83
5.2.1	Results on the binding energy of CO_2	85
6	Diffusion in portlandite nanopore	89
6.1	Studied systems in portlandite nanopore	89
6.1.1	Mixing water and carbon dioxide	89

6.1.2 Water and carbon dioxide within portlandite nanopore . . .	91
7 Summary, conclusions and prospects	97
7.1 Discussion of the benchmarking results	97
7.2 Discussion of the quartz results	98
7.3 Discussion of the nanopore system	99
7.4 Prospects for future work	101
III Appendices	103
Appendix A Force fields	105
A.1 Force Field parameters	105
A.1.1 ClayFF	106
A.1.2 Water models	106
A.1.3 Models for carbon dioxide	109
A.1.4 FPF	110
A.1.5 Morse	111
A.1.6 Tersoff	112
Appendix B Example code and input scripts	115
B.1 Scripts and data files	115
B.2 Minimization procedure	119
B.3 Minimum image convention for triclinic systems	120
B.4 The Abel computing cluster	122

List of Figures

1.1	CO ₂ storage possibilities	6
1.2	Chattanooga shale in Tennessee	8
1.3	Assessed world shale basins onshore	9
1.4	SEM images of shales	10
1.5	SEM nanoporous shale cross-section	10
2.1	Minimum image convention overview	25
2.2	Minimum image convention explained	26
2.3	Lennard-Jones potential	29
2.4	Comparing Morse with harmonic oscillator potential	32
2.5	Work of adhesion	40
2.6	Mean square displacement of water	46
2.7	Mean square displacement as function of distance to surface	49
2.8	Tetrahedral structure of water	50
2.9	Radial distribution, binning particles	51
3.1	One dimensional potential surface	61
4.1	Bulk water radial distributions	70
4.2	Diffusion coefficient of bulk water	71

2 List of Figures

4.3	Tetrahedral order of bulk water	73
4.4	Density of carbon dioxide for three force field models	74
4.5	Unit cell of portlandite	76
4.6	Portlandite slab	77
5.1	Structure of quartz surfaces	80
5.2	CO_2 sequestration	82
5.3	Structure of quartz systems	84
5.4	Surface energy of quartz	86
6.1	Water and carbon dioxide in portlandite nanopore	92
6.2	Density profiles in nanopore systems	94
6.3	Water diffusion as function of distance to nanopore surface	95
A.1	Water model	107

List of Tables

4.1	Bulk water radial distributions	70
4.2	Diffusion coefficient of bulk water	72
4.3	Benchmarks for CO_2	74
4.4	Benchmarks for portlandite	77
5.1	Surface energy of quartz	85
5.2	Surface energy of quartz at higher temperatures	87
5.3	Binding energy of CO_2 to a quartz surface	87
6.1	Water and carbon dioxide	90
6.2	Nanopore system information	91
6.3	Carbon dioxide diffusion in nanopore	96
A.1	ClayFF parameters for portlandite	107
A.2	SPCE and SPC force field parameters. H_2O	108
A.3	TIP-4P force field parameters. H_2O	108
A.4	TraPPE force field parameters. CO_2	110
A.5	EMP2 force field parameters. CO_2	110
A.6	FPF force field parameters. CO_2	111
A.7	Morse force field parameters. SiO_2	111

4 List of Tables

A.8 Tersoff force field parameters. CO_2	112
--	-----

Chapter 1

Introduction to the thesis

Interaction of carbon dioxide with mineral surfaces is becoming more important to understand because of an increased interest in CO_2 sequestration on mineral surfaces in relation with an enhanced and more environmental conscious oil industry[56]. CO_2 sequestration is widely discussed as one of the most important measures for reducing the emissions of greenhouse gases, and it is believed that this is possible in geological formations deep below the seabed. Formations of special interest are porous, water-filled aquifers and empty oil-reservoirs located under impermeable clay layers. The reason for the existence of the oil-reservoir is this covering layer of shale or slate that has kept the organic materials, as *kerogen*, there for so long. Shale and slate are both sedimentary nanostructured tight rocks composed of silt-sized particles, like quartz fragments, and clay minerals. They often contain large amounts of hydrocarbons within the nanosized structures as they are formed by sedimentary deposits that may have contained and covered large amounts of organic material. Most of the continental shelf is covered by such sedimentary rocks, and now, companies are trying to pump carbon dioxide into near-empty oil wells in, and below such formations in order to extract more oil from the well. This is possible because the carbon dioxide turns into liquid. At a constant temperature of 300 Kelvin degrees, carbon dioxide becomes liquid at 66 *atm* pressure, which corresponds to the pressure at 670 *m* below sea surface. The density of the earths crust is on average approximately three times as large as the density of water, so that kind of pressure will be reached already about 220 *m* below the surface of the earths crust. A temperature of 300 *K* will however be reached at about a kilometer below the surface since the geothermal gradient increases approximately linearly with about 25 degrees per kilometer, and we keep in mind that the temperature at the seabed is near four degrees Celsius, equivalent to 277 *K*. A reasonable pressure to use, is said to be about 100 *atm*. Oil companies already pump seawater into near-empty oil wells to extract more

from the reservoir. The method is known as water injection or water flooding, and the idea is to increase the pressure in the reservoir and push the oil into a well where it can be collected. So what if we could use carbon dioxide instead? Or in combination with the injection of water?

How do we make sure that it stays in the well, and for how long can we assume that it is safely stored?

The ideas of carbon storage and the possibility of storing toxic and radioactive wastes in deep-crustal formations, are not new. A lot of research have been going on in these areas the latest few years, and important progression is made.

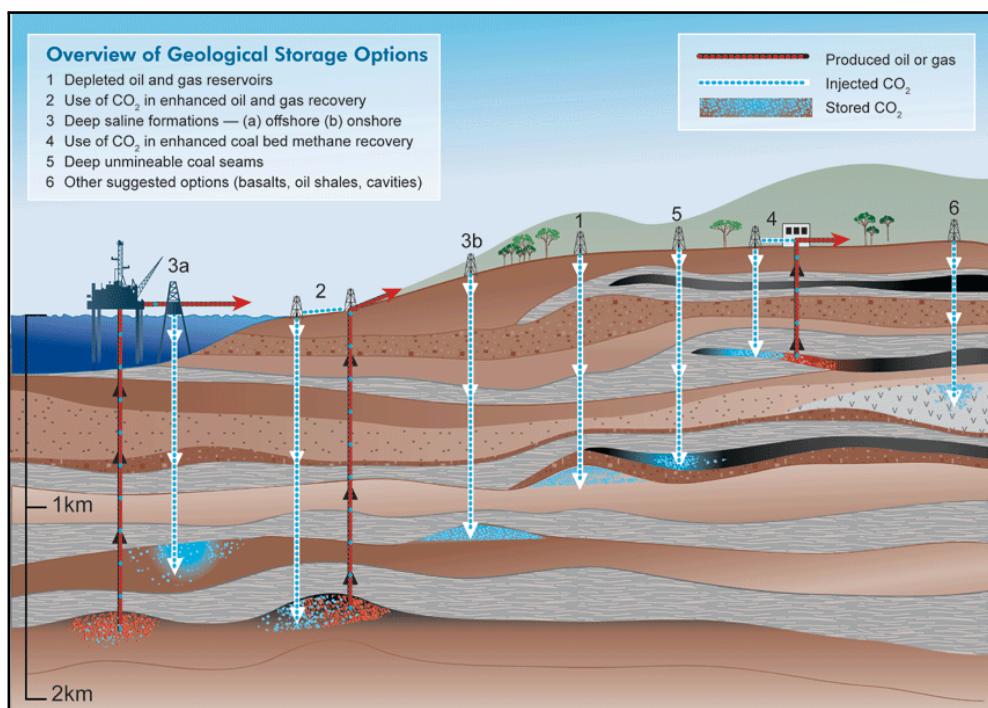


Figure 1.1: The possibility of injecting CO_2 into near-empty oil wells to extract more from the well, or just store the carbon dioxide here. Displace oil and gas in fractured nanoporous shales and coal.

Figure from *Lawrence Berkley National Lab, division of Earth Science*[17]

In fig. 1.1, some thought scenarios are illustrated. Near-empty oil and gas reservoirs can be depleted by injecting liquid carbon dioxide at 100 atm pressure. The CO_2 can be injected into shale rock formations, fracturing (fracking) the rock and displace oil and gas from within the nanoporous rock formations, or in a similar manner be used to displace liquid methane from within coalbed formations. It is also believed that CO_2 can be stored in saline aquifers under impermeable clay layers, as is being tried out in the Utsira sandstone formation on the continental shelf in the North Sea[2]. Looking back at fig. 1.1, the carbon

dioxide can be compressed and sent both from land and offshore platforms, and be pumped down into an empty oil well or a shale rock for storage. Regardless of the location, the sealing layer for the storage chamber seems to be almost exclusively, shales or mudstones. This is because locations where hydrocarbons already have been stored for millions of years sounds like a safe and secure place to put it. Since the carbon dioxide then will come in contact with compounds found in slate and shales, it is therefore important to know how it behaves in such an environment. What properties of the surroundings are essential for the transport of the carbon dioxide within the pores inside the shale structure, will it react and be bound to any of the compounds, or will it dissolve its way through the materials? And how much can we pump into that specific well?

In order to better understand if, where, and in what kind of formations it is possible to store carbon dioxide (and perhaps other greenhouse gases), and for how long, it is necessary to know what is going on at the molecular level. The chemical and physical properties of the combined systems. How does water affect the transportation of carbon dioxide in the pores, how does the presence of hydrocarbons effect sequestration and transportation, and how does the pore size alter the diffusive transportation properties? If carbon dioxide is captured (sequestered) by some mineral, how well is it stored?

To get some answers to these questions, it is obviously important that research is being carried out. There are a few places that such clay layers can be studied in broad daylight. In Chattanooga, Tennessee, some parts of a shale formation has emerged to the surface, and is exposed in several rivers and in the Highland Rim surrounding the central basin of Tennessee. An image of a characteristic layer is shown in fig. 1.2. The carbon-rich layer is squeezed between sedimentary rocks that has kept the organic material locked up for millions of years.

However, most information about the shales and clay structures are gathered from geo-radars that gather images of the geological structures based on electromagnetic radar pulses in the microwave frequency band, and from core samples that are drilled, and raised to the surface. For the kind of depths we are talking about here, a geo-radar, or a *ground-penetrating radar* (GPR), is only used to gather information of the whereabouts of larger structures in the crust, like faults, groundwater and oil well locations. To study structures far into the earths crust, you would like to use low frequencies. This is because high frequencies tend to loose its energy more rapidly than low frequency, and therefore low frequencies will travel further in the material. Using lower frequencies is a trade off for poorer resolution. The method of seismic mapping of the seabed and the earths crust is widely used, and estimates on available resources in the discovered locations is made from such surveys along with content measurements found from core samples. A map over onshore shale structures is shown in fig. 1.3, where estimates



Figure 1.2: The Chattanooga shale in Tennessee. It is said that the carbon content in some areas is so high that it is possible to light the rock on fire
Figure from *Geochaching*[19]

on oil content is made. The assessment from this study show that there could be 345 billion barrels in these shales. In comparison, IEA -the International Energy Agency has worked out that more than 34 billion barrels of oil is used every year world wide. So it will hold for ten years or so based on these results. But the number seems to be rapidly increasing, and what about the offshore locations at the continental shelves? It is also believed that with climate change, the north pole will reveal large amount of such resources.

As mentioned just a little earlier, we gain a lot of information from core samples, where we might study the layers and the microscopic structure of the materials found in the rock. The core sample can be cut into fine pieces and studied under microscope, or a sample may be prepared for x-ray analysis to gather diffraction information about the crystal structures present. Another important method is the use of Scanning Electron Microscopes, SEM-imaging. With a SEM device, it is possible to study the surface structure of the sample down to the nanometer scale[11], the electron beam can even be focused so much that it evaporates a spot on the sample, and the evaporate gas is studied to gain information about the elements present at that location. SEM imaging in 3D has newly been possible, where porosity, micro-structures and kerogen content is mapped. The method clarifies nanoscale properties of gas accumulation in shales, but the method destroys the sample. In figs. 1.4a and 1.4b, two SEM images of the inside and surface texture of a shale is shown. The leftmost image

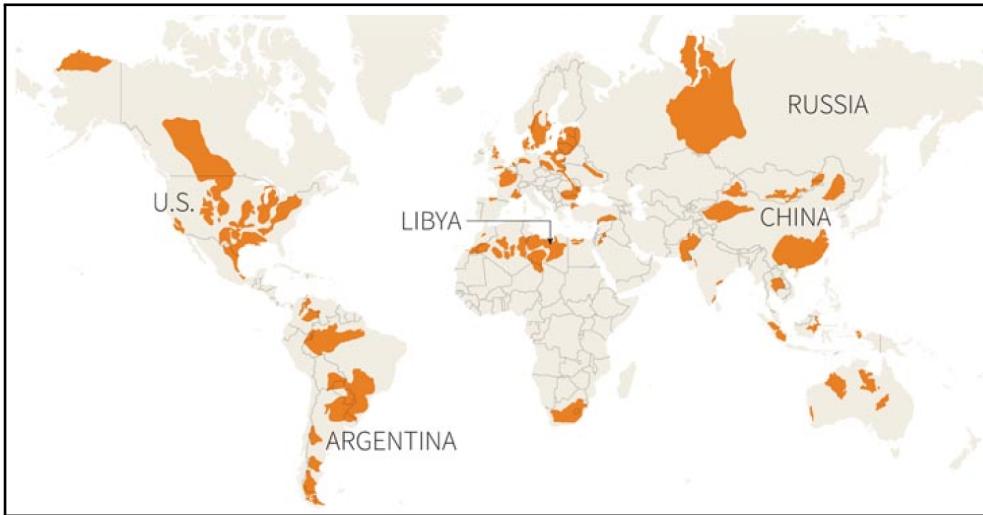


Figure 1.3: World map over assessed shale basins onshore. The number of estimated resources in these basins increased dramatically from 2011 to 2013. From 32 billion to 345 billion barrels. The estimates were performed by IEA. Figure from *Tomson Reuters*[47]

shows a quite large pore, almost a micron wide. The rightmost image shows the surface structure of a middle-aged shale, where the edges of the flakes have been somewhat rounded compared to a younger shale. It is obvious that there is room for gas and liquids inside the pores as can be shown in the cross-section sample in fig. 1.5, where nanopores of organic material is seen as black spots.

From these kind of samples, we gain information about chemical compositions and geometrical structures, but when the sample is raised to the surface, the pressure and temperature is changed, and the material properties of whatever were inside the pores have changed. Much of the natural gas that were inside the pore is long gone. A pressure chamber would be a location where we again could study the system in the environmental conditions that we would like to. So-called hyperbaric chambers have been developed[48], and core samples have to be preserved at the *in situ* pressure (although stresses in the core sample is released since the lithostatic pressure is not conserved when the sample is cut), so that the sample can be studied in an environment close to the one found *in situ*. The method of retrieving the core samples at the *in situ* pressure is called *pressure coring*, and is used to maintain accurate oil saturations, gas volumes and water content.

Although we have good methods, analyzing the overall and large-scale properties of the material, it is difficult to know what is going on at the molecular level inside the pores, and the molecular level structure of these systems is important for the transport properties. Porosity and permeability are examples of macro-

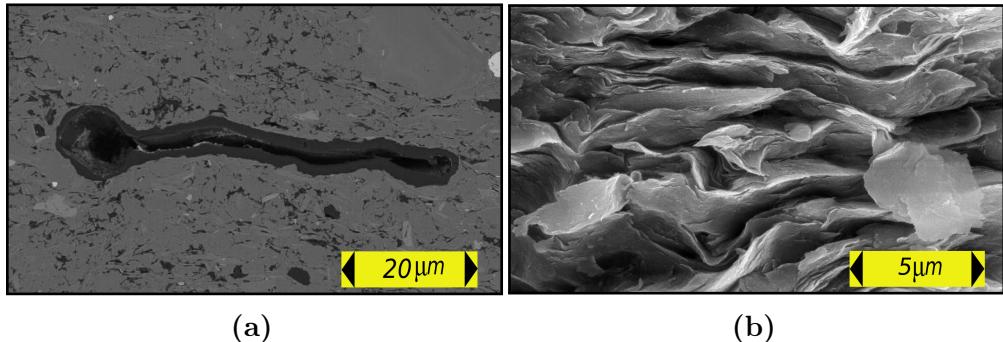


Figure 1.4: SEM images. fig. 1.4a is a backscatter image that shows a cross-section through a shale rock. fig. 1.4b shows the surface structure of a middle-aged shale.

Figures from *JEOL solutions for innovation*[22]

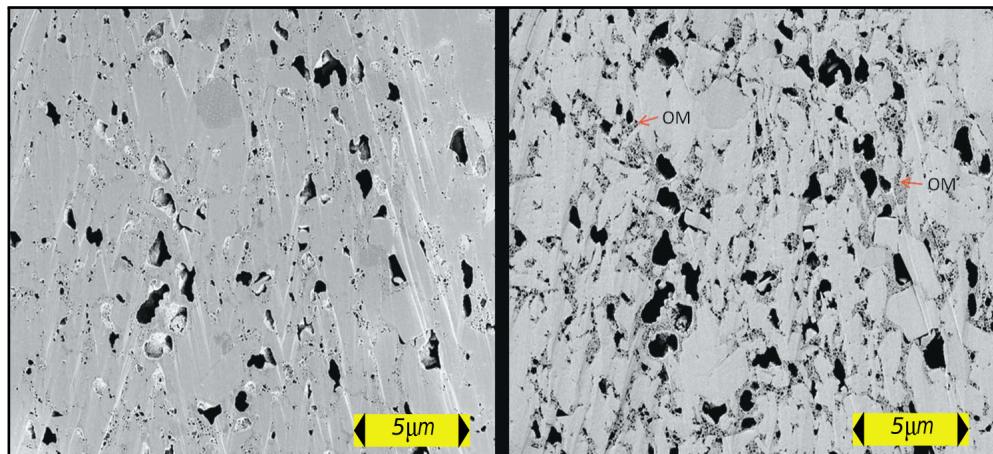


Figure 1.5: Here we see two images of the same shale cross-section sample, where organic material is seen in the nanopores. The small nanopores pointed out is examples of nanopores that are not visible in the leftmost image.

Figure from *AAPG subsurface science*[51]

scopic measurements of material properties that are the result of the microscopic structures. The porosity of a rock is the relative amount of non-solid space in a material, and permeability is a materials ability to let a fluid flow through it. It is obvious that for a shale type of structure, with layers, fluids flow much more easily along parallel planes with the layers, while it can be almost impermeable perpendicular to the layers. The pores and cleavages in the shales are at the nanometer scale, so a large amount of the flow will be very close to the surfaces where the properties of the fluids may be very different from the bulk behavior, and therefore will have an impact on the large scale dynamics.

Simulations of molecular dynamics can help us to understand what happens

at the molecular level and is an important link to the continuum level of macroscopic systems. During a molecular dynamics simulation you have the possibility to control exactly what particles that are present in the system at any time. One may easily monitor thermodynamic observables like temperature, pressure, volume, densities, and calculate thermodynamic properties like viscosity and diffusion. Using a physical laboratory to carry out the experiment can sometimes be expensive. You never have the full control over what particles are present in the system, and what influence these may have on the experiment. Another challenging problem with physical experiments in the laboratory arise when you want to study pore systems. To measure what happens at the molecular level inside nanostructured tight rocks or artificial structures like *zeolites*[53], which are materials formed by cages that together form networks of pores inside the zeolite crystal, is problematic because most of the activity occurs inside the material. This is also the case for shales. Therefore, molecular dynamics becomes an important source for comparing with experimental laboratory results, and to gain greater insight of the behavior of fluids in such systems. The backside of molecular dynamics simulations is that, on this length scale, it is an approximation to the real world physics governed by the laws of quantum. We must be conscious about errors that may occur, and the limitations of the models that are being used. In order to study the large scale molecular interactions and their implications, we therefore need to compare with experimental results to say something about the accuracy and precision of our measurements. For example when studying a shale system, it would be foolish to create too complicated systems at first, since the mechanisms is not understood. Adding complexity should be done with care. Therefore, in MD simulations, we usually study very simple systems, where we have a very good understanding of what is going on, and the errors is limited to the model and methods that we use.

1.1 Modern molecular dynamics studies of material properties

Many studies using MD simulations in their research give their reasons for choosing molecular dynamics exactly because of the rapid increase in cheap computational power[63]. The cost of performing MD simulations is pointed out as an important aspect, as for example in a study of possible ionic liquid solvent for CO_2 , there were simply too many possible liquids to attempt measuring thermodynamic and transport properties for all of them. MD simulation were therefore used to predict liquids for further experimental testing[41]. Another reason for using MD is because the “predictive power of molecular models is superior to classical models”[36] since MD is adaptable, and it provides information about

all thermophysical properties, like geometrical structures, transportation of materials and thermodynamics. Also reasonable time scales for observing these properties are usually within range. Water is one of the most widely modeled materials[63], and is used to study for example evaporation rates from hydrophobic confinements as is believed to play an important role in biophysical phenomena like the closing and opening of ligand-gated ion channels in a number of proteins so that certain ions can pass through[52]. The understanding of water close to hydrophobic materials could also be very important for the design of new self-cleaning materials[52], and for studying the properties of new discoveries in material science.

MD-simulations is much used to study protein folding because it is possible to run atomistically detailed, physics-based simulations that are sufficiently long for observing multiple instances of foldings and unfoldings within a single run towards minimization of the system energy for the protein[42]. The simulations of proteins provides valuable insight in the driving processes of the foldings. In a late study in 2014, an atomistic molecular dynamics simulation of carbon dioxide in water were performed to investigate the diffusive properties at a wide range of temperatures and pressures. The study performed by Muoltos et al.[37], showed that the pressure dependence of the diffusion coefficient for CO_2 in the water environment at constant temperature were negligible for temperatures lower than 473.15 K , and that their results were in good compliance with experiments. Which I would say is a confirmation of the models that are being used.

When it comes to the behavior of fluids in clay nanopores, several investigations of flow and diffusive properties have been performed. Because the “understanding of mass transfer through clay pores”[4] is important for carbon sequestration and the disposal of toxic and radioactive waste in petroleum and gas-reservoir engineering, Boțan et al.[4, 34] conducted MD simulations of hydrodynamics in montmorillonite and pyrophyllite clay nanopores. The survey showed that the Navier-Stokes equation for describing classical fluid dynamics only were in accordance with the MD simulations if the nanopore were wider than 3 nm (approximately 30 water molecules wide), and that proper boundary conditions with a stick or slip condition is taken into account. Basically what they show, is that the continuum dynamics is insufficient for small systems, and that MD simulations can be a method for understanding such behavior.

1.2 Ethics concerning the climate

I hope that it is a bit more clear why I would like to study carbon sequestration and fluid transport in clay minerals, and that I have motivated the use of molecu-

lar dynamics in doing so. But before I go on and tell you about the goals for this thesis, I find it necessary to say something about the ethical aspect of the research that is being done in this field. Because it is clearly an ethical aspect with the whole process of helping the oil industry with research that benefit their economy, and that will prolong the search for hydrocarbons. At one hand side is it a political matter, where science have to be brought to the people and politicians so that the right measures are made. On the other, science can help an existing oil industry reducing their emissions, find alternatives and new methods for reducing the amount of greenhouse gases in the atmosphere, and thereby make a difference on the climate in the long run. I think that this is a better solution than not to help the industry, because then I think that little or no improvements will be made, and the industry will exist as long as they make money, and they will make money until new, better and cheaper energy technology is made available.

1.3 Goals

We are at a stage where technological and engineering methods have surpassed our basic understanding of processes in tight rock systems like in nanoporous shale and clay structures. The tight rocks pose new scientific problems because of the small length-scales in the range of nanometers. I want to study the behavior of water and carbon dioxide within a clay nanopore, to investigate the transport properties as a function of distance to the surface of the clay. I would also like to look into the possibility of carbon dioxide sequestration, by investigating the binding energy of CO_2 to a quartz surface, which is commonly found in shales. A shortlist of goals that will be important in this line of work would then be:

- Create a clay unit-cell structure
- Develop a clay nanopore model
- Investigate properties of water and carbon dioxide inside the pore structure
- Evaluate models for the constituent parts. Benchmarking
- Compute surface energies of quartz surfaces
- Calculate the binding energy of CO_2 to a quartz surface
- Compare surface and binding energies with DFT simulations

1.4 Structure of the thesis

In the first part, basic theory, concepts and methods in molecular dynamics are introduced. Also some more advanced topics is discussed since they are widely used, like the Nosè-Hoover thermostat. An introduction to the software that have been used extensively for the simulations and visualizations performed in this thesis, is presented. This passage is ment to be a handy compression of the most important information that I can give someone who wants to get started with *LAMMPS*, and can easily be omitted for a first time reader. The second part of the thesis presents the simulations. This part can be divided into four. The first present the constituent bulk systems used in the larger simulations and results from the bulk simulations. The results from the larger simulations is divided into two parts, where the first presents the systems and results from simulations of quartz surfaces and the carbon dioxide interaction with the surface. The second part of the larger simulations presents the simulations of carbon dioxide and water inside a portlandite nanopore, where we look at the transport properties inside the nanopore. Finally we will have a brief summary of the thesis and a discussion of the results. In the appendices you'll find more detailed information about the force fields and potential coefficients used in the thesis, example input scripts for *LAMMPS* and other example code snippets.

Part I

Theory and methods in molecular dynamics

Chapter 2

Introduction to molecular dynamics

Molecular dynamics is the simulation of the movements of atoms confined within a system. The atoms are modeled as point particles, meaning that they have no actual volume in the mathematical terms. Their interactions are described by force fields, collections of potentials that describe the physical interactions between the particles. From these interactions we use Newton's laws of motion to predict their movements. This is a simplification. We are applying classical mechanics on a system that we know is governed by the laws of quantum mechanics. But if we were to use quantum physics, we would have to solve Schrödinger's wave equation, which in most computational methods scales non-linearly with system size. Even for the most powerful supercomputers in the world, this would rapidly become a problem too large to solve when the number of particles and particles increase. In order to look at sufficiently large systems, and also longer timescales, we therefore have to make some simplifications. Although we miss out on some complex chemical interactions at phase transitions and in special environments, we have to use molecular dynamics to perform tasks at this scale. As mentioned, the atoms are modeled as point particles, no electrons are included as free particles in the simulations, so within a molecule, each atom is assigned a charge that is calculated from evaluations of the electron density, a common result from quantum mechanics. Also bonds between particles have to be created specifically since we do not include the electrons. Density functional theory is a method that is closer to quantum mechanics, because the electrons are modeled as a density function around the atoms, and that allows to look at some dynamics at larger system sizes¹

¹If you wish to read more about density functional theory, I suggest reading *A Primer in Density Functional Theory*, by C.Fiolhais F.Nogueira M.Marques. And have a look at work done by the Kitchkin research group <http://kitchingroup.cheme.cmu.edu/dft-book/dft.pdf>

In order to model the electromagnetic interactions, potentials have been developed and fitted to reproduce natural behaviour. There are different force fields developed for molecular dynamics simulations. Some are good at large scale simulations because they are computationally cheap, and others are better to use on smaller systems because they are more precise but would not be able to crunch larger problems, since they would use too long time. Many potentials are developed to work for special cases, and are obviously very precise in their estimates at the given circumstances that they are designed for. More general force fields are way more reliable because they are not designed to work just in a narrow pressure and temperature range, and not for just some special atoms and molecules. Models like the TraPPE - Transferable Phase Potential Equilibria, ClayFF - Clay Force Field and the ReaxFF - Reactive Force Field, aim to be widely usable. Therefore we are going to have a closer look at these force fields later on.

In this thesis we will only focus on classical molecular dynamics where the motion of the particles are computed using Newton's equations and where the potentials describing the forces between the particles are either empirically fitted to the behavior of the real world materials, or fitted using the results from quantum mechanical calculations. From quantum computations one may obtain potential energy surfaces for the interactions between atoms. We then find mathematical formulas (potentials) that we fit empirically to the data set and may use in the molecular dynamics simulations. One of these potential energy surfaces is the well known *Lennard-Jones* potential, which aims to describe the *van der Waals* forces between particles.

In physics today, it is well known that there are four forces of nature. The *strong* and the *weak* -interactions are the forces that works between elementary particles like quarks and leptons and also particles that they constitute, like bosons, hadrons and baryons. These two forces belong to the nuclear and high-energy physics domain, and their interactions have very short range, typically $< 10^{-5} \text{ nm}$ [23]. The two other forces are effective over a much larger range of distances, all the way from subatomic distances to practically infinite distance. These are the *electromagnetic* force that interacts between charged particles, and the *gravitational* force that interacts between all particles with mass. The electromagnetic forces is the source of intermolecular dynamics, and therefore determines the material properties. Since molecular dynamics is a simplification of the quantum world, and we only wish to look at intermolecular interactions, all the potentials that describes the forces between atoms and molecules are based on the electromagnetic force. Gravity can also be applied, but then as a simple external force field, for example to simulate flow.

In this chapter we are going to introduce some common potentials used in

molecular dynamics, theoretical concepts and methods used to look at structure and physical properties of the materials. But first we will have a look at how Newton's law of motion is used to update the positions and velocities of the particles.

2.1 Time-integration

Newton's second law of motion states that the sum of forces acting on an object with mass m , is the origin of the objects acceleration: $\sum F = m\mathbf{a}(t)$, and determines the movement of the particles. By time-integration from an initial state $(\mathbf{r}(0), \mathbf{v}(0))$, we will find the system state at a later time $(\mathbf{r}(t), \mathbf{v}(t))$. We know that the acceleration is defined as the derivative of the velocity with respect to time, and that the velocity is the derivative of the position with respect to time. So that $\mathbf{a} = \frac{d\mathbf{v}}{dt} = \frac{d^2\mathbf{r}}{dt^2}$. Solving for the velocity and the position of a particle i is done by time-integration of the acceleration:

$$\mathbf{v}_i(t) = \mathbf{v}_i(0) + \int_0^t \mathbf{a}_i(t) dt \quad (2.1a)$$

$$\mathbf{r}_i(t) = \mathbf{r}_i(0) + \int_0^t \mathbf{v}_i(t) dt \quad (2.1b)$$

On a computer, we need to discretize the time evolution to solve the integrals. The integral becomes too complicated to be solvable by hand, and we cannot iterate over the infinite amount of real numbers. The solution is simply to discretize the time into timesteps Δt . The discretization leads to errors, but as long as the time step is small enough, the error in every time step is also small. It is important that we avoid accumulation of the error, so that the result does not drift further away from the actual solution! Many integration methods have been developed to predict the evolution of such systems, manipulations of the time-discretized integrals have given us many different algorithms for calculating the values at later time steps. The standard method is the Euler's method, where a Taylor expansion of the integral is used to predict the next values, here illustrated by the well-known Euler-Cromer method in eq. (2.2), where the obtained value for the velocity is used to estimate the position in the next equation. The $\mathcal{O}(\Delta t^2)$ is a standard method of representing the error. And goes like Δt^2 for the Euler and Euler-Cromer method.

$$\mathbf{v}_i^{n+1} = \mathbf{v}_i^n + \mathbf{a}_i^n \Delta t + \mathcal{O}(\Delta t^2) \quad (2.2a)$$

$$\mathbf{r}_i^{n+1} = \mathbf{r}_i^n + \mathbf{v}_i^{n+1} \Delta t + \mathcal{O}(\Delta t^2) \quad (2.2b)$$

In eq. (2.2) above, we have used that $\mathbf{v}_i(t) = \mathbf{v}_i^n$ and $\mathbf{v}_i(t + dt) = \mathbf{v}_i^{n+1}$, which is common notation. The Euler-Chromer method is known to be a better solution than the plain Euler method simply because it uses the knowledge of what will happen to the velocity at the next time evaluation to predict the next position. The standard method to obtain more complex solutions, is to use Taylor expansions of higher order, and doing the expansion both forward and backwards in time. We will in the next section introduce the method that is most widely used in molecular dynamics. The reason for this is a combination of precision, stability and efficiency, all of which is important properties for a molecular dynamics simulation!

2.1.1 Velocity Verlet algorithm

The Velocity Verlet algorithm is the first choice time-integration algorithm used in molecular dynamics simulations. This is mainly because it has a good numerical stability and preserves the system energy very well, unlike simpler Euler schemes where the total energy in the system will typically have a drift. It also has the ability, like the “Leapfrog” scheme, of time-reversibility, which is an important feature for energy preservation. But the Leapfrog time-integration has an error that goes like $\mathcal{O}(\Delta t^2)$, and since the Verlet algorithm goes like $\mathcal{O}(\Delta t^4)$, with almost the same effort, Verlet is the weapon of choice for molecular dynamics simulations.

To find the velocity Verlet algorithm we approximate the time-integration by a third order Taylor expansion, and does so for one time step forward in time, and also another expansion for one time step backwards in time as shown in eq. (2.3).

$$\mathbf{r}_i(t + \Delta t) \approx \mathbf{r}_i(t) + \mathbf{r}'_i(t)\Delta t + \frac{1}{2}\mathbf{r}''_i(t)\Delta t^2 + \frac{1}{6}\mathbf{r}'''_i(t)\Delta t^3 + \mathcal{O}(\Delta t^4) \quad (2.3a)$$

$$\mathbf{r}_i(t - \Delta t) \approx \mathbf{r}_i(t) - \mathbf{r}'_i(t)\Delta t + \frac{1}{2}\mathbf{r}''_i(t)\Delta t^2 - \frac{1}{6}\mathbf{r}'''_i(t)\Delta t^3 + \mathcal{O}(\Delta t^4) \quad (2.3b)$$

The following step is to add together the Taylor expansions in eq. (2.3), resulting in eq. (2.4).

$$\mathbf{r}_i(t + \Delta t) \approx 2\mathbf{r}_i(t) - \mathbf{r}_i(t - \Delta t) + \mathbf{r}''_i(t)\Delta t^2 + \mathcal{O}(\Delta t^4) \quad (2.4)$$

Using a simpler notation, where n denotes the time, we have now found the Störmer-Verlet method eq. (2.5). This algorithm is not widely used because it involves adding the small Δt^2 term to the much larger terms, which may lead to large round-off errors[15].

$$\mathbf{r}_i^{n+1} = 2\mathbf{r}_i^n - \mathbf{r}_i^{n-1} + \mathbf{a}_i^n \Delta t^2 \quad (2.5)$$

The way that the Verlet algorithm deals with this, is that it first calculates the velocity at a half time step $\frac{1}{2}\Delta t$, forward in time, as shown in eq. (2.6)

$$\mathbf{v}_i^{n+1/2} = \mathbf{v}_i^n + \mathbf{a}_i^n \frac{\Delta t}{2} \quad (2.6)$$

Then the position is updated in eq. (2.7) using this intermediate velocity

$$\mathbf{r}_i^{n+1} = \mathbf{r}_i^n + \mathbf{v}_i^{n+1/2} \Delta t \quad (2.7)$$

Next step is to calculate the acceleration based on the updated position

$$\mathbf{a}_i^{n+1} = \frac{\mathbf{F}_i(\mathbf{r}_i^{n+1})}{m} \quad (2.8)$$

Once the acceleration is calculated through eq. (2.8), the velocity is calculated by stepping another half time step $\frac{1}{2}\Delta t$, using the updated acceleration. Ultimately, eq. (2.9) is the final step in the *Velocity Verlet algorithm*:

$$\mathbf{v}_i^{n+1} = \mathbf{v}_i^{n+1/2} + \mathbf{a}_i^{n+1} \frac{\Delta t}{2} \quad (2.9)$$

As Dragly points out in his master thesis, this way of performing the calculations ensures smaller round-off errors. You may read more about the Velocity Verlet algorithm in literature like in [62].

2.2 Minimum image convention

Periodic boundary conditions are widely used in molecular dynamics as a way to simulate a continuous system, much larger than the actual system that we have, without lots of extra computational cost. When periodic boundary conditions are used, it means that, when a particle moves out of the system boundary, it reappear at the other side of the system. Leaving the system in positive x-direction for a system that is sized $x \in [0, L]$, yealds that the particle gets a new position $x_{i+1} = x_{i+1} - L$. This way we also makes sure that particles does not drift away from the system, and gets lost. However, the particle may escape the system if the forces on the particle is so large that the particle is sent a distance larger than two times the system size in just a time step. That is, if x_{i+1} suddenly becomes two times larger than L , it will not come back into the system. This happens from time to time, typically if the system is initiated with particles too close to each other. We make fun of it, and say that the system exploded, but in secrecy, we are sad and depressed because we'll have to find the reason for the

sudden bond or angle-loss, or even a segmentation fault² that arised because of the loss of a particle.

When we use periodic boundary conditions, it is important that the particles close to the boundary feels the forces from the particles at the other side of the boundary. If not, particles close to the boundary will frequently tend to rapidly jump from one side of the system to the other. And there will be an increased probability that particles will suddenly come very close to eachother. In order to have periodic boundary conditions, we need to wrap positional information about all particles in the system to make sure that we are looking at the shortest distance between two particles. This method is called the *minimum image convention*, and needs to be taken into account when, e.g. the forces on a particle is calculated, or when we are looking at the diffusion of particles in the system.

Imagine that we have a system with only two particles like in fig. 2.1. The particles are close to the system boundaries on each side of the system. When applying minimum image convention, the distance measured from the red particle to the blue is found to be the shortest distance between the red particle and a blue particle in one of the system replicas. Because the replicas are mirror images of the system, the method is called the minimum image convention.

In a computer program, the syntax for a rectangular system would be straight forward to implement. The distance vector can be decomposed and one may look at one direction at a time. Take a look at the one-directional problem illustrated in fig. 2.2. Herein, the distance vector from the red particle to the blue, have to be changed into the mirror image of the blue particle, which points in the negative direction. We may then easily see that the new distance $dx_{(new)} = dx_{(old)} - L_x$. If the distance $dx_{(old)}$ were less than minus half the system length (i.e. $< L_x/2$), you will have to add the system length, L_x to the particle distance $dx_{(new)}$, so that $dx_{(new)} = dx_{(old)} + L_x$. This is the minimum image convention, and is applied for every direction of the distance vector $\mathbf{r}_i = (dx_i, dy_i, dz_i)$, between every particle i in the system. The method is simple enough as long as we are working with an orthogonal system, where all three system vectors are perpendicular onto each other. In case of working with triclinic systems, where the system vectors are tilted, you will have to look more closely into the minimum image convention. The minimum image convention for triclinic systems is briefly explained in the Appendix, section B.3.

²A segmentation fault is something that arise when the computer try to read some part of the memory that is just junk, but understand that it is junk, and then exits the program

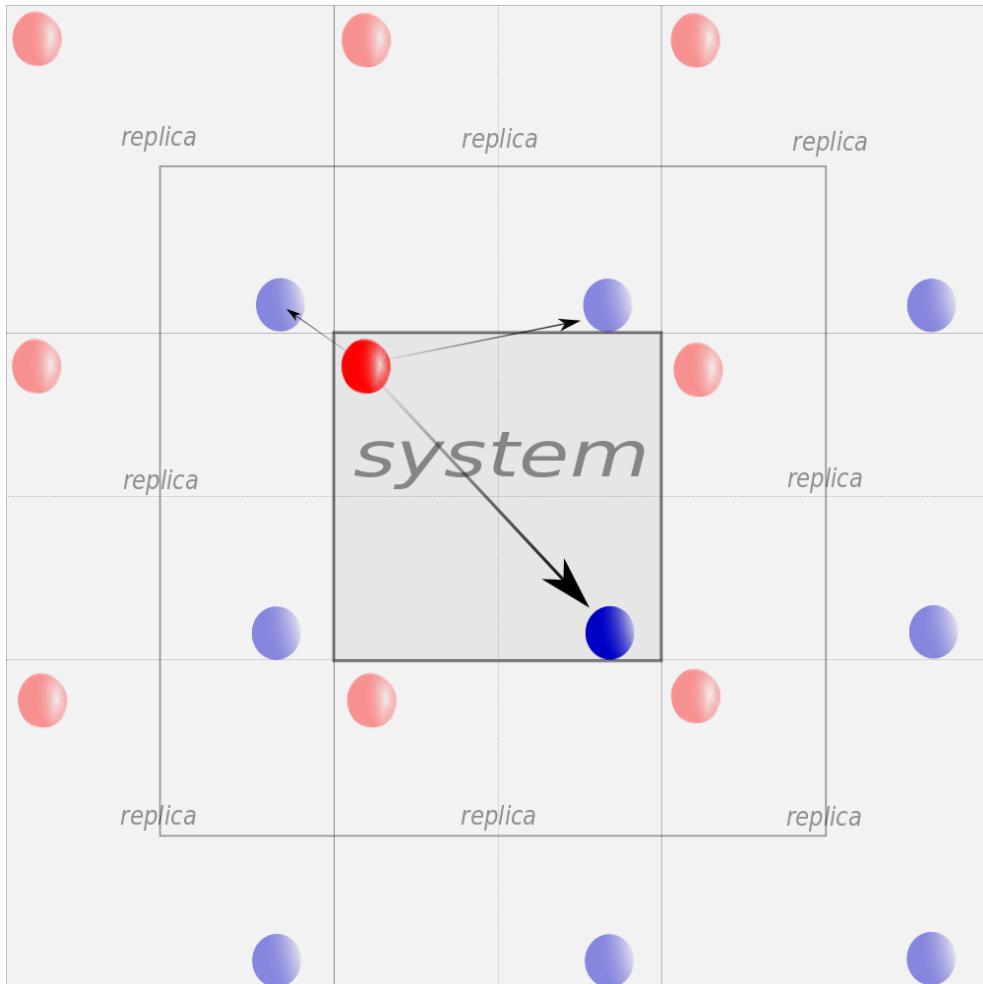


Figure 2.1: A system consisting of a red and a blue particle. They are obviously very far away from each other within the system, but if we use periodic boundary conditions, we can imagine that the system is replicated around the original. Now we see that there are two mirror images of the blue particle that results in a closer distance between them

2.3 Ensembles in thermodynamics

Thermodynamics is the statistical mechanics approach to the understanding of energy and its relation to heat and work. It states that the total change in thermal energy ΔU , in a system is equal to the heat Q , added to the system and the work W , done on the system, $\Delta U = Q + W$. Thermal heat is just the warming or cooling from an external heat bath, and the work that is done on the system is in practice the net change of volume for the system due to external pressure. This is known as the first law of thermodynamics, and it tells us how

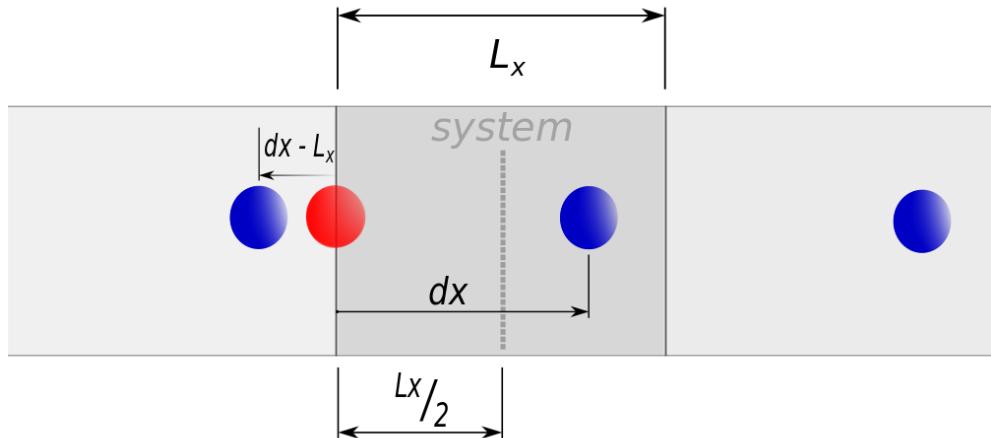


Figure 2.2: A system consisting of a red and a blue particle. The minimum image convention changes the distance between the red particle to the blue (distance vector directed in positive direction), so that $dx_{(new)} = dx_{(old)} - L_x$

the internal energy of a system is changed by influence from the surroundings. So we can change the temperature of the system by adding heat from an external heat bath, or we can change the volume of the system by applying an external pressure.

It is always important to know what kind of ensemble that is being studied, i.e. what variables are held fixed, what variables are allowed to fluctuate round a mean, and what variables is of no interest because they are allowed to fluctuate freely. The most well-known ensembles is the *microcanonical*, the *canonical* and the *grand canonical* ensemble. These systems all contain some particles, and the particles have thermal and potential energy which sums up to the total energy of the system at any time. The system has a temperature and is confined within a volume that defines the system boundaries.

2.3.1 Microcanonical ensemble

The microcanonical ensemble (NVE), is the ensemble of a fixed number of particles N at a constant volume V , in which the total energy of the system is constant. You may say it a little obscenely that theoretically this is a canonical ensemble without a thermostat, so that the temperature is allowed to fluctuate freely with the kinetic energy of the system. In practice, is the microcanonical ensemble not a realistic ensemble to control experimentally because it is very difficult to estimate what the total energy of the system really is. In a molecu-

lar dynamics simulation however, the total energy is easily determined from the sum of potential and kinetic energy of all particles in the system, since we have perfectly control of all particles locations and velocities at any time.

2.3.2 Canonical ensemble

In the canonical ensemble (NVT), the temperature is controlled, and kept at an averaged fixed value. This means that the kinetic energy fluctuate, and therefore we have an exchange of energy with an imagined external heat bath while the total number of particles N , and the total volume of the system is kept fixed. This is the ensemble that is most used in molecular dynamics to compare with experimental results. In a lab experiment however, it is not easy to know how many particles are present in the system, but the temperature and the volume is more easily controled at an average.

2.3.3 Grand canonical ensemble

The only variable kept fixed in the grand canonical ensemble is the volume. Heat and particles are allowed to be exchanged with a reservoir to keep the temperature and the chemical potential μ at a constant average. The grand canonical ensemble is therefore also referred to as the μVT -ensemble. The chemical potential is a lesser known quantity, but just like we have *thermal equilibrium* when two systems have the same temperature and *mechanical equilibrium* when they have the same internal pressure. Two systems are said to be in *diffusive equilibrium* when they have the same chemical potential. A system with a highter chemical potential tends to loose particles to the system with a lower chemical potential. But what is chemical potential?

The chemical potential is associated with the change in the system energy if you add a particle and keep the entropy and volume fixed in the process. So how do you keep the entropy constant?

Entropy is widely known as the disorder in a system, and is basically the number of ways to arrange the particles in the system³. In order to keep the entropy fixed when adding a new particle to the system, we must make sure not to add heat to the system as that would change the number of possible configurations. So $Q = 0$. Refering to Schroeder in [50], the change in entropy can be calculated as $dS = Q/T$.

³ Entropy $S = k \ln(\Omega)$, where Ω is the multiplicity (the number of ways to arrange the particles) of the system and k is Boltzmann's constant

2.4 Potentials

A potential field describes how a force acts on a body affected by it. The gravitational field is an example, and is a radial symmetric potential field, just like the electromagnetic field. The gravitational field works on an object with mass m , pulling it towards the center of mass of another object M . When at the atomic level, the gravitational field is very weak compared to other forces like the electromagnetic forces described by the *Coulomb* and *van der Waal* forces. So gravity is seldom accounted for in molecular dynamics except when someone wants to simulate a flow driven by some external pressure field.⁴ In chemistry and biology the potential field is most often referred to as a force field. In this thesis we will talk about a force field as the collective use of potentials used to describe all interactions in the system. This is because sets of potentials are used to describe different interactions between the particles, like the Lennard-Jones potential is used to describe van der Waals forces and the Coulomb potential describes the electrostatic two-body interaction. But what is a potential field? A potential field is the latent energy that can be transformed into kinetic energy for some body under influence of a force, and that is why it got its name. The gradient of a conservative potential field is the force that acts on the body, and is mathematically expressed as eq. (2.10).

$$\mathbf{F} = -\nabla V = -\left(\frac{\partial}{\partial x} + \frac{\partial}{\partial y} + \frac{\partial}{\partial z}\right)V \quad (2.10)$$

2.4.1 Van der Waals forces and the Lennard-Jones potential

Van der Waal forces is a collectively term on all forces that works on neutrally charged particles. The repulsion due to the Pauli principle and the attractive force due to either a permanent multipole inducing a multipole in another molecule, or the instantaneous arising of multipoles due to intermolecular dynamics of the electrons⁵. The van der Waal potential between two particles can be

⁴To make a density driven flow is difficult because you have to maintain the density distribution difference over time, which implies that at one location we have a mean density ρ_A , and at another ρ_B , and you have to apply special thermostats for these areas, whereas for a gravitational driven flow you only have to include an extra constant force contribution in some direction to every particle. However you have to be careful in the choice of thermostat so that you avoid killing the flow!

⁵van der Waal forces [40], information also gathered from website http://en.wikipedia.org/wiki/Van_der_Waals_force (04.04.2015)

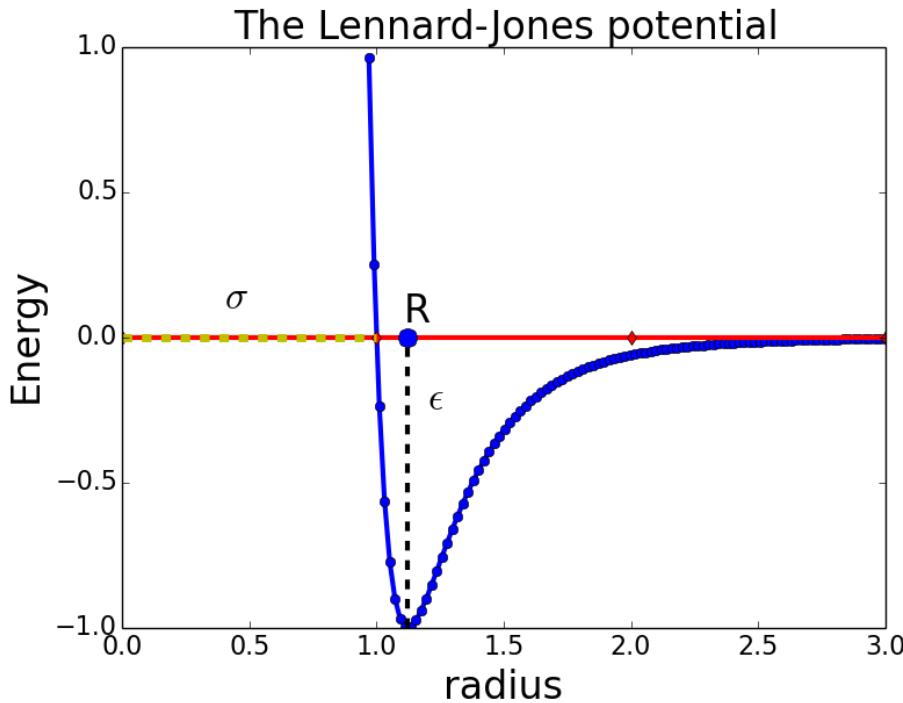


Figure 2.3: Van der Waal forces are modeled as a Lennard-Jones potential, where ϵ is the depth of the potential well, σ is the distance to the first zero point crossing, and R_o (in figure denoted as R) is the distance to the minima of the potential well

found from Hartree-Fock simulations using quantum mechanics where the simulation is performed for two atoms having a radial distance r , from each other. The radial distance is changed systematically, and the potential energy is calculated. For each radial distance, we get a point on a curve similar to the Lennard-Jones potential curve, and from this we can estimate σ , R and ϵ . The Van der Waals interaction is most often modeled as a Lennard-Jones (12-6) function as shown in eq. (2.11).

$$E_{VDW} = D_{o,ij} \left[\left(\frac{R_{o,ij}}{r_{ij}} \right)^{12} - 2 \left(\frac{R_{o,ij}}{r_{ij}} \right)^6 \right] = 4\epsilon_{ij} \left[\left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^6 \right] \quad (2.11)$$

Where $D_{o,ij} = \epsilon_{ij}$ and $R_{o,ij} = \sqrt[6]{2}\sigma_{ij}$, are empirical parameters derived from the fitting of the model to data from observed physical properties [12]. The reason for showing the two equivalent expressions, is to raise a flag of awareness. Although a factor of $\sqrt[6]{2} \approx 1.1225$, is a number quite close to one, it makes a whole lot of difference in a molecular dynamics simulation if a σ value is used in the expression where we should have used R_o ! Important features of the Lennard-Jones potential is the dramatical increase of the repellent force when the

distance between the atoms is getting small. If particles are placed too close in the beginning of a simulation, the forces acting on the particles are going to be huge, and the system will, in lack of a better word, “explode”. But just before this dramatic repulsion, we have a part which is attractive, and has its equilibrium state in $R = R_o$. From hereon and out, the potential energy flattens rapidly out as the $(\sigma/r)^6$ term catches up with $(\sigma/r)^{12}$, and goes to zero as the distance r between the particles goes to infinity. Of course, we cannot evaluate the potential to infinity, so in practice, we have a cut-off distance typically at $R_{cut} = 3\sigma$, which is a good approximation.

2.4.2 Coulomb interaction

The Coulomb interaction is the electric force which works between charged objects. In molecular dynamics we always model forces between sets of particles. The Coulomb interaction potential is a two-body interaction and works pair-wise. It is defined as a constant times the product of the charges of particle i and j , divided by the interatomic distance r_{ij} , as shown in eq. (2.12).

$$E_{Coul} = C \frac{q_i q_j}{r_{ij}} \quad (2.12)$$

Where the constant $C = \frac{e^2}{4\pi\epsilon_0}$, e is the electron charge, and ϵ_0 is the permittivity⁶ in vacuum. The Coulomb force is repulsive for equally charged particles and attractive for particles with opposite charge. Therefore is the electrostatic force between particles only a function of the distance between them, and the charges of the particles that interact. And by the way, charge is given in units of the elementary charge of the electron, where $1e = 1.60217657\text{E-}19$ coulombs.

2.4.3 Long range Coulomb interactions

Long range electrostatic interactions, is a problem in molecular dynamics simulations because they reach so long that they have impact beyond the system size. Therefore, methods have been developed to deal with it. Since the long-range interaction should work on the confined periodic system, a constant term has to be added to the potential field to compensate for the long-range interaction. The Coulomb interaction eq. (2.12) goes like $1/r$, and therefore it dies slowly out as the distance r between the charged particles increase. It is not clear whether the potential can be cut off beyond some finite range [58].

⁶Permittivity is a measure of the resistance in a material when forming an electric field in it

There are three main methods in use to deal with long range electrostatic interactions. These are the *Multi-level Summation Method* (MSM), *Particle-Particle Particle Mesh* (PPPM) and the *Ewald summation* method. We will not go into either of these, but encourage the reader to look at works like [59] and the section about long-range interactions in [58].

Important notice about these methods is that they only add a constant to the potential and does not alter the shape of the potential, hence they do not change the dynamics of the system.

2.4.4 Bond potential

A bond potential is often modeled as a harmonic oscillator, and is, for example, used to describe the *OH*-interaction. The harmonic bond potential is shown in eq. (2.13), and is found to be the one best fitted with observational data on the stretch energy associated with each hydroxyl bond [12]. The bond potential is computationally cheap, and reproduce very effectively radial distributions and dynamics of systems with sufficiently low temperature where we do not expect any bond-breaking. The harmonic oscillator potential is plotted to the left in fig. 2.4. As you see in this figure, a particle affected by this potential can not break free because the forces will rapidly become too large and force it back into the equilibrium length $r = R_o$, or $\theta = \theta_o$ as for the angle potential, which also is commonly modeled as a harmonic oscillator.

$$E_{bond} = K_1(r_{ij} - R_{o,ij})^2 \quad (2.13)$$

2.4.5 Angle potential

As Cygan et.al [12] describes it, “an additional enhancement” to the ClayFF model⁷, to better describe metal sorption of hydrated surfaces, and also to improve the vibrational behavior of the hydroxyl groups, a three-body angle bend potential was used for the hydroxyl groups. The angle bend vibration is also described by an easy harmonic potential as shown in eq. (2.14). The angle potential have got just the same shape as the bond potential, but is a function of angle and not distance.

$$E_{angle} = K_2(\theta_{ijk} - \theta_{o,ijk})^2 \quad (2.14)$$

⁷The ClayFF model can be read more about in section A.1.1

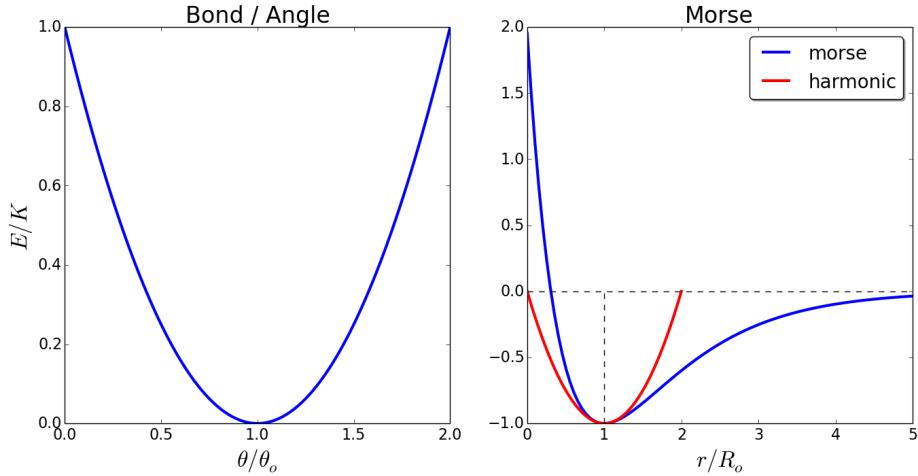


Figure 2.4: Harmonic oscillator potential for the bond and angle (left), and the Morse potential to the right. Notice that the potentials are plotted as a function of the particle distance divided by the equilibrium distance. The potentials energy is plotted as E/K , where the energy scaling factor $K = K_1, K_2, D_o$ for the different potentials and defines the depth of the potential well

2.4.6 The Morse potential

Details about the Morse potential is gathered mostly from the *LAMMPS* documentation and in [27]. The *LAMMPS* software can be read more about in section 3.1. This potential is convenient when looking at higher temperatures where a bond potential is not sufficient because we expect bond-breaking. The Morse potential is computationally much more expensive than the bond potential because of its two exponential terms. It is implemented with a cut-off for the radial interaction distance just as for the Lennard-Jones potential, so particles can be lost, and particles can be caught if within the cut-off distance. The Morse potential is meant to be used instead of a harmonic oscillator potential like the bond and angle-potential. If we are interested in systems where we expect reactions, the Morse potential should be used instead of the harmonic-oscillator potential although this will slow down the code significantly! The mathematical form is shown in eq. (2.15), and is plotted with the harmonic oscillator potential in fig. 2.4. Notice the similarity with the Lennard-Jones potential!

$$E_{Morse} = D_{o,ij} \left\{ \exp \left[-2\alpha_{ij}(r_{ij} - R_{o,ij}) \right] - 2\exp \left[-\alpha_{ij}(r_{ij} - R_{o,ij}) \right] \right\} \quad (2.15)$$

2.4.7 The Tersoff potential

The Tersoff potential is a *bond order* potential which, unlike a pair potential, modifies the bond strength by weakening the bonds according to the angles between a bond and all the other bonds for a specific bond between atom i and j . The potential contribution from V_{ij} is a function of f_R , which is a two-body term, and f_A is the three-body term weakened by b_{ij} . This complicated potential can be studied in more detail by reading J.Tersoff's article [24].

LAMMPS displays the Tersoff potential closely to what is done in Tersoff's article, but with slight modifications, allowing an order m in the ζ -function for the exponential term, and setting $h = \cos(\theta)$ in the bond-angle function $g(\theta)$.

$$E = \frac{1}{2} \sum_i \sum_{ij} V_{ij} \quad (2.16a)$$

$$V_{ij} = f_C(r_{ij}) \left[f_R(r_{ij}) + b_{ij} f_A(r_{ij}) \right] \quad (2.16b)$$

$$f_C(r) = 1 : r < R - D \quad (2.16c)$$

$$f_C(r) = \frac{1}{2} - \frac{1}{2} \sin\left(\frac{\pi}{2} \frac{r - R}{D}\right) : R - D < r < R + D \quad (2.16d)$$

$$f_C(r) = 0 : r > R + D \quad (2.16e)$$

$$f_R(r) = A \exp(-\lambda_1 r) \quad (2.16f)$$

$$f_A(r) = -B \exp(-\lambda_2 r) \quad (2.16g)$$

$$b_{ij} = (1 + \beta^n \zeta_{ij}^n)^{-\frac{1}{2n}} \quad (2.16h)$$

$$\zeta_{ij} = \sum_{k \neq i,j} f_C(r_{ik}) g(\theta_{ijk}) \exp\left[\lambda_3^m (r_{ij} - r_{ik})^m\right] \quad (2.16i)$$

$$g(\theta) = \gamma_{ijk} \left(1 + \frac{c^2}{d^2} - \frac{c^2}{[d^2 + (\cos(\theta) - \cos(\theta_0))^2]} \right) \quad (2.16j)$$

To use the Tersoff potential in *LAMMPS*, a parameter input file have to be added in the *pair_coeff* method according to the *pair_style tersoff*. Example files are found in the potentials folder in the file structure of the *LAMMPS* download.

2.4.8 ReaxFF

All the other potentials are different from the reax force field in the sense that it can form and brake bonds with all particles in the system if this is energetically favorable! In similar manner to other force fields, the ReaxFF divides the system energy into partial energy contributions. A bond interaction, van der Waals, Coulomb, a valence angle term, a torsion term, two terms for under and over-

coordination of the bond-order also called coordination number, and a term for conjugation effects. The sum of the energy contributions is shown in eq. (2.17), where also a penalty term is added to “reproduce system stability of systems with two double bonds sharing an atom”, quoted from the original paper by Goddard III et al. [61].

$$E_{system} = E_{bond} + E_{over} + E_{under} + E_{val} + E_{pen} + E_{tor} + E_{conj} + E_{vdw} + E_{coul} \quad (2.17)$$

I will hereon give a short description of the terms used in the reactive force field. For further insight on the matters, I suggest reading papers like Van Duin and Kulkarni [61, 28].

Bond order and bond-terms

The first term, E_{bond} is the bond order term, and is obtained directly from the interatomic distance r_{ij} between a pair of atoms. It is meant to describe the number of chemical bonds between pairs of atoms, and consists of three terms for the force field developed for hydrocarbons. One called the sigma bond for the carbon-hydrogen and hydrogen-hydrogen interaction, a pi-bond and a double pi-bond for carbon-carbon interaction. All the different terms in eq. (2.18) have their own equilibrium bond lengths and coefficients for the strength of the interaction.

$$BO'_{ij} = \exp \left[p_{bo,1} \left(\frac{r_{ij}}{R_{o,ij}} \right)^{p_{bo,2}} \right] + \exp \left[p_{bo,3} \left(\frac{r_{ij}}{R_{o,ij}^\pi} \right)^{p_{bo,4}} \right] + \exp \left[p_{bo,5} \left(\frac{r_{ij}}{R_{o,ij}^{\pi\pi}} \right)^{p_{bo,6}} \right] \quad (2.18)$$

The bond-order BO'_{ij} is corrected for over-and undercoordination in the valence angles based on the valence bond theory which assumes that bonds are created from atoms sharing electrons. The under-and overcoordination of atoms kicks in if there are irregularities according to the number of valence bonds (coordination number) for a single atom.

The valence angle potential describes the energy associated with the bending, and it is important that it goes to zero as the bond-orders goes to zero since then there will be no bond!

The Torsion is actually a four body potential that accounts for the energy in the twisting of a central bond BO_{jk} that forms angles with the bonds BO_{ij} and BO_{kl} . If one of the bonds dissociates, this is also described as a smooth disappearance of the torsion energy.

The final term for the bond-terms are the conjugation effects, which is a phenomenon where π -electrons (electrons in p-orbitals) can be shared over three or

more than three atoms[49], the sharing of the electron(s) due to the adjacent, parallel, overlapping π -orbitals results in a stabilization of the molecule (i.e. the system energy is lowered).

Nonbonded van der Waals interactions and the Coulomb term

The van der Waals repulsive interaction at short range due to the Pauli principle, and the attractive force due to dipole-moments (Debye and London forces), is modeled as a distance-corrected Morse-potential, so that the repulsion becomes constant as $r_{ij} \rightarrow 0$. This is also the case for the Coulomb interaction. The potentials are said to be “shielded”. The van der Waals forces is then modeled as the shielded Morse-potential in eq. (2.19), where λ_1 , λ_2 and λ_w are constants that is provided for the force field .

$$E_{vdw} = D_{o,ij} \left\{ \exp \left[\alpha_{ij} \left(1 - \frac{f(r_{ij})}{R_{o,ij}} \right) \right] - 2 \exp \left[\frac{\alpha_{ij}}{2} \left(1 - \frac{f(r_{ij})}{R_{o,ij}} \right) \right] \right\} \quad (2.19a)$$

$$f(r_{ij}) = \left[r_{ij}^{\lambda_1} + \left(\frac{1}{\lambda_w} \right)^{\lambda_2} \right]^{1/\lambda_2} \quad (2.19b)$$

The Coulomb interaction is shielded to adjust for orbital overlap between close distanced particles. The potential is presented as in eq. (2.20), where γ_{ij} is also a parameter specific for interactions between particle i and j .

$$E_{coul} = C \frac{q_i q_j}{[r_{ij}^3 + (1/\gamma_{ij})^3]^{1/3}} \quad (2.20)$$

As we see, the reax force field is a complex system of many potentials. Actually, *LAMMPS* uses even some more potentials, 14 terms to be precise, although two of them is always zero⁸.

There is a lot of parameters that has to be optimized for the different interactions, and for interactions between different kind of particles. So it exists sets of parameters obtained from different kinds of optimizations for different kinds of systems, and we have to choose one that sounds promising for the kind of system that we would like to study.

The ReaxFF is a forcefield that is fully reactive, and works quite well for small molecular dynamics systems, but struggles with larger systems and over longer time-scales because it makes use of a long list of computationally expensive exponential terms that has to be called for every particle in the system, every iteration. But still, it performs way better than pure quantum dynamics and

⁸if you want to read more about this, the *LAMMPS* code is based on the supplementary material of an article published in 2008 by Goddard III et al.[8]

quantum-chemical methods if we want to look at systems with more than a few hundred particles and over timescales of picoseconds.

2.4.9 Using ReaxFF in *LAMMPS*

This little digression is ment for people who have some knowledge to *LAMMPS* or have read the chapter about software and *LAMMPS* specifics in section 3.1.

It is very easy to include the reactive bond order force field, called ReaxFF in *LAMMPS*, there are only a few tings to pay attention to. Firstly, the *atom_style* of the input *.data*-file have to be:

`atom_style charge`. Second is, as for the Tersoff potential, a parameter input file has to be added in the:

`pair_coeff` method according to the:

`pair_style reaxc` (remember that the package USER-REAXC have to be installed/included in the build of the executable). And last, a charge equilibration of the system should be performed, and is done by the:

`fix qeq/reax` command, with the `reax/c` keyword. See the *LAMMPS* manual for more information, or even read Goddards article on the subject [45], that *LAMMPS* based their ReaxFF on.

2.5 Thermostats - controlling the temperature

The reason for implementing a thermostat is obviously to control the temperature in the system. Temperature is particle movement, and particle movement result in kinetic energy, or vice verca. Temperature is still being measured with the use of mercury confined within a glass column because mercury is far away from its phase change in everyday normal temperature on earth, and it has the physical property that it expands quite linearly in that temperature range [50]. So when you want to measure the temperature of, say a cup of hot tea, you put the thermometer into the cup and wait until the mercury has the same temperature as the hot water. When the mercury has got the same temperature as its surroundings, it stops expanding, the two objects is said to be in *thermal equilibrium*, and the volume that the mercury occupies shows the temperature. Since it is quite linear in its expansion, it should be fairly easy to define a temperature scale just by using two arbitrary temperatures like the freezing and boiling point of water, and set a mark for the mercury in the glass-tube to 0 and 100 at these measurements and mark up equally spaced intervals in between, then voilà! We have the Celsius centigrade scale! We could also use another material, for ex-

ample an ideal gas at fixed pressure (ideal gas law $PV = nRT$) or a material that change its electrical resistance with temperature! Digital thermometers use thermistors who's resistance varies with temperature, and therefore can be used to read the temperature. The thermistor is made up of a ceramic polymer where we assume that the resistance and temperature is linear $\Delta R = k\Delta T$. This will alter the current in the circuit that the thermistor is part of, and can be read by an ampere-meter in the circuit. From Kirchoff's law and the relation between the resistance and the temperature, we might make an estimate on the temperature in the surroundings.

In molecular dynamics we can read the temperature directly from the kinetic energy in the system, or even in a part of the system! From thermodynamics we have that the kinetic energy, or total thermal energy is $E_k = \frac{f}{2}NkT$, where N is the number of atoms, each with f degrees of freedom. T is of course the temperature and k is the Boltzmann factor. The derivation of the expression for the total kinetic energy from the ideal gas law is shown in [50] and is worth looking at, but I'll leave that to the reader! The total kinetic energy in the system is statistically $E_k = \frac{1}{2}m\langle \mathbf{v}_i^2 \rangle$, so that we can find the temperature according to eq. (2.21). We see from this equation that the statistical average temperature is given by the *root-mean-square* of the velocity, rather than by the average velocity. Also from this equation we see, by taking the square root, that at a given temperature, small molecules travel at higher speeds than heavier molecules, which seems to be in good compliance with our intuition!

$$T = \frac{m\langle \mathbf{v}_i^2 \rangle}{3Nk} \quad (2.21)$$

When implementing a thermostat, it is important to know what it does to the system. The aim of a thermostat is not to keep the temperature fixed, as that would mean to keep the kinetic energy fixed. It is rather to keep the temperature average to be at the desired temperature. To do this, the velocities of the particles has to be rescaled gradually. The Andersen thermostat developed by Hans C. Andersen, a professor of physical chemistry, is a physical realistic thermostat that simulates perfect inelastic collisions between atoms in the system, and atoms in an external heat bath. This is done by assigning new velocities to a random fraction of the atoms in the system. But the Andersen thermostat changes the velocity of a particle directly, and the new velocity has no connection to the old what so ever. Therefore we should avoid using the Andersen thermostat on systems where we want to measure physical properties related to the movement of independent particles, like diffusion.

2.5.1 The Berendsen thermostat

This thermostat rescales the particle velocity of a particle by a constant γ according to the relative temperature between an imaginary heat bath and the system temperature. The idea is simply to give every particle a little extra push if they move too slowly, or slow them down if they move too fast! Also, in the Berendsen thermostat there is a damping parameter τ that typically should be $\tau = 20\Delta t$, where Δt is the time step used in the simulation. If we set the damping parameter $\tau = \delta t$ the temperature of the system will change to exactly the temperature T_{bath} , of the heat bath at every time step that the parameter γ is used to scale the velocity [3].

$$\gamma = \sqrt{1 + \frac{\Delta t}{\tau} \left(\frac{T_{bath}}{T} - 1 \right)} \quad (2.22)$$

The Berendsen thermostat is very good to lower or rise the temperature of the system, and it is computationally cheap. It allows temperature fluctuations and, if we use a reasonable damping factor, it only moves the dynamics of the system towards the temperature of the heat bath. But we change the velocity of all the particles in the system at the same time, which is not physically realistic. A more realistic method would be to give only a few particles this push, because not all particles in the system will collide with reservoir particles at the same time.

2.5.2 The Nosé-Hoover thermostat

Perhaps the most used thermostat in molecular dynamics is the Nosé-Hoover thermostat. It gives very accurate dynamics, and represents the canonical ensemble at a satisfactory level. The simplest form of Nosé-Hoover thermostats is the only one that will be shown here. For further reading on the topic I suggest to look at Filip Sunds master thesis [55], or have a look at [21].

As we shall see, the Nosé-Hoover thermostat does not change the velocity directly. A thermodynamic friction coefficient ξ is introduced in the equations of motion. The force that acts on each particle is as follows in eq. (2.23).

$$\mathbf{F} = -\nabla U(\mathbf{r}) - \xi m \mathbf{v} \quad (2.23)$$

Where in eq. (2.23), the coefficient of friction depends on time $\xi = \xi(t)$. We have to derive our equations of motion using this new way of describing the force that acts on each particle. From Newton's third law we have that $\mathbf{F} = m\mathbf{a}$, and inserting for the acceleration into the velocity verlet algorithm eqs. (2.5), (2.6) and (2.9) yields eq. (2.24).

$$\mathbf{v}_i^{n+1/2} = \mathbf{v}_i^n + \left[-\frac{\nabla U(\mathbf{r}_i^n)}{m} - \xi^n \mathbf{v}_i^n \right] \frac{\Delta t}{2} \quad (2.24a)$$

$$\mathbf{r}_i^{n+1} = \mathbf{r}_i^n + \mathbf{v}_i^{n+1/2} \Delta t \quad (2.24b)$$

$$\mathbf{v}_i^{n+1} = \mathbf{v}_i^{n+1/2} + \left[-\frac{\nabla U(\mathbf{r}_i^{n+1})}{m} - \xi^{n+1} \mathbf{v}_i^{n+1} \right] \frac{\Delta t}{2} \quad (2.24c)$$

We see now that we suddenly have an implicit scheme, because we have to know the velocity that we are trying to calculate at a time step where we don't have it. Implicit solvers exists, and solves problems like eq. (2.24c) using iterative methods, but this results in a non-reversible solution which, as pointed out earlier, is an important property of molecular dynamics simulations, because it guarantees conservation of energy. The derivation of the implicit velocity verlet algorithm can be studied in Frenkel and Smit's "Understanding molecular dynamics simulations" [18]. According to [55], a solution to the problem with the scheme being implicit has been proposed by Martyana et al. [35]. Nevertheless, when solving the implicit scheme, we find that the friction coefficient can be written as eq. (2.25).

$$\xi = \frac{sp_s}{Q} \quad (2.25)$$

And that the time evolution of the friction parameter goes like eq. (2.26), where s , is a degree of freedom introduced to the system, and p_s and Q is the momentum and "mass" associated with s .

$$\xi^{n+1} = \left(\sum_{i=1}^N \frac{(p_i^n)^2}{m_i} - 3Nk_B T^n \right) / Q \quad (2.26)$$

Where in eq. (2.26), $(p_i^n)^2$ is the squared momentum of particle i at time step n . The equations developed by Nosè and improved by Hoover results in a scheme that influence the equations of motion in a manner that alters the forces exerted on every particle as a function of their momentum (which is just a similar way of looking at the velocity or kinetic energy or temperature). The value Q is the connection to the heat bath, and according to eqs. (2.24) and (2.26) will a smaller value Q , result in a stronger coupling ξ .

2.6 Adsorption

Adsorption is unlike *absorption*. While absorption is a volumetric quantity where a gas or liquid permeates a solid surface, or is dissolved by a liquid, adsorption is a surface process where the *adsorbate* attaches to the surface (*adsorbent*) and becomes a part of it, or becomes a layer on top of it. The accumulation of atoms or molecules is called *adhesion* if the constituent particles of the adsorbent and the adsorbate are different, and *cohesion* if they are of similar type.

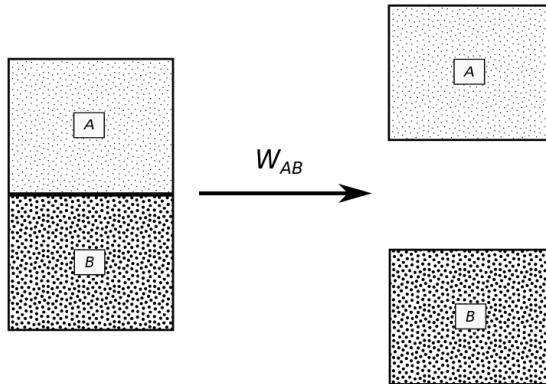


Figure 2.5: Work of adhesion is the work done separating two objects from each other to an infinite distance where the attractive force between the materials is essentially zero

Adsorption is a result of surface energy, where it is energetically favorable to bind to the surface due to van der Waal and electrostatic forces. In vacuum all particles are attracted to each other, so the phenomenon should be studied in an environment of some fluid where the two materials come in contact with each other, but, as I will come back to very soon, the work of adhesion and cohesion is calculated in vacuum. In nature, animals like frogs, geckos and some insects exploit this phenomenon to be able to climb walls and ceilings, and the adhesion effect is normally fortified by increased humidity. There are two types of energy measurements that are interesting when looking at adsorption. One is the energy associated with the work of adhesion, that is, the change in free energy or reversible work to separate the two media[23]. The other one is the binding energy. The binding energy is equal to the difference in potential energy from an unbound state to a bound state, and the excess energy is released in terms of work and heat in the system. When it comes to the work of adhesion, this is also, in theory, a simple calculation. Two materials in a fluid environment like water or air, are connected. This connection is a system state, and correspond to a system energy. Then we ask ourselves what amount of work has to be done to separate the materials? Since the work of adhesion is different in one media and another, the value is calculated in vacuum as a reference.

Let us consider two materials *A* and *B* as in fig. 2.5. The work:

$$W_{AB} = \int_0^\infty F_{AB} ds$$

of adhesion, is the integral of the total forces acting on *A* from *B* (or vice versa, N3L⁹) from zero to infinite distance. Fortunately one may consider the work of

⁹Newton's third law of motion

adhesion in terms of surface energy and surface tension. The free energy change for a medium whose surface area is increased by a unit area of the surface, is equal to the process of separating two half-unit areas from contact (cohesion remember!). Therefore we might postulate that the free energy change γ of creating a twice as large surface, is equal to the work of cohesion (of separating) of one unit surface area. Mathematically described in eq. (2.27) [23].

$$\gamma_A = \frac{1}{2} W_A \quad (2.27)$$

Now the problem is reduced to calculating the surface energy instead of calculating the work of adhesion. Surface energy is usually measured in units of energy per unit area [Jm^{-2}], and can be calculated as the potential energy difference of bulk material and the potential energy of the material with its surface. Since the potential energy of a material with a surface is larger than the bulk material because of a higher disorder, we would like to calculate the surface energy per unit area as in eq. (2.28).

$$E_s = \frac{E_{surf} - E_{bulk}}{A_{surf}} \frac{\frac{N_{surf}}{N_{bulk}}}{\frac{N_{surf}}{N_{bulk}}} \quad (2.28)$$

There are some important notes to make to this equation. First of all we have taken into account and corrected for an uneven number of atoms in the bulk and in the surface system by including the particle numbers of the systems, N_{surf} and N_{bulk} . And the area that we have to divide by is the area that spans the surface. You might question; what is the surface energy of bulk material?

Well, a surface is a part of the material which is in contact with something else, and works as an interface between the material and the outside world. The material in contact with the surface will interact with the particles of the surface, and thereby change the potential energy of the system. But what if the material is in vacuum and not in contact with anything, you may ask?

Since the particles at the surface does not have the same surroundings as the bulk material, there will always be a difference! The surface particles does not feel any forces from particles above them like the particles in the bulk material, and because of this fact, this definition of a surface still holds. To say that the surface is the interface between two materials would be wrong, because an interface has an infinitesimal width, but a surface is the part of the material that confines the bulk.

Let us approach the topic of adsorption. Adsorption is, as described, the binding of a material onto another materials surface. The potential energy of a bound state is lower than a unbound state, and this energy difference is the binding energy. In order to tell how strong the bond is, we have to find the energy released during the formation of the bond. But this is a difficult task since that means

to measure the heat released, and work done from the particles in the bond on the system. An easier approach is to measure the potential energy of the bound state, and compare with the potential energy of an unbound state. The difference must be the energy associated with the binding of the two!

$$E_b = E^{bound} - E^{unbound} \quad (2.29)$$

But then I ask, how do we find the energy of the unbound state?

This is in fact an important question. If the two components are in the same system, the lowest energy state will be the bound state, and therefore we have to look at two separate systems to find the energy of the unbound state. I.e. we have to find the minimum potential energy for the adsorbate and the adsorbent independently, $E^{unbound} = E^{adsorbent} + E^{adsorbate}$. We would also like to find the binding energy per particle bound to the surface, so that if we study a system of N particles bound to a surface we will find eq. (2.30).

$$E_b = \frac{E_N^{bound} - E^{adsorbent} - NE^{adsorbate}}{N} \quad (2.30)$$

In eq. (2.30), we denote that the bound system has N bound particles, and that we calculate $E^{adsorbate}$ looking at a system consisting of one particle (atom, molecule, or whatever is the number of bound particles) of the binding material. If we look at the above equation we should expect a negative binding energy E_b , since the bound state has a lower potential than the unbound. Having more particles bound to the surface will normally inflict the resulting binding energy in a fashion that gives a lower binding energy because the surface is being saturated.

In this thesis the binding energy of carbon dioxide to a silica surface is studied, and will be presented in detail later on. The calculations are straight forward, but as with the surface energy, the minimization procedure is the important part. As for the surface energy, we have to study the systems with a non-periodic boundary condition in the surface direction for the adsorbent, and for all directions for the free carbon dioxide molecule, or make sure that we have systems large enough that the particles are not affected by their own potential field across the boundary of the system. For these systems, and also when we find the surface energy, we use the minimization procedure described in section 3.2.4 to find a potential minimum, we also try to use molecular statics and just apply minimization procedures where we iterate a thousand minimizations where the forces and the energy of the system is evaluated. We stop the procedure if we have reached a state where the difference in the system energy from one iteration to the next is typically lower than 10^{-6} (using unitless energy). The stopping tolerance for the total force is of the same order, but in units of $kcal/mol$. The stopping tolerance is valid if the largest force-component is lower than this order of magnitude.

2.7 Filling a nanopore

After relaxation of a bulk material, we would like to carve a hole in it. This can easily be done by removing all particles within a defined volume. You have to make sure to also remove all particles that is bonded, that is, if you remove a oxygen molecule in a OH-group, you have to remove the hydrogen as well. At the end, charge should still be conserved, and no bonds, angles or dihedrals should be broken, but removed if the particles don't exist anymore.

When we have created a nanopore in a bulk material and want to fill this volume with some fluid, like water, there are a few methods that points out as the weapon of choice. An obvious solution is to use a Monte-Carlo method, where a molecule is given a random location within the system. If this location is within the volume, and is sufficiently far away from any other atom, then we will include this molecule in the system. You have to keep on doing the process until the desired density is achieved. *Voxelation* is a method where the system are divided into smaller volumes, called *voxels*. The system is divided into $n_x \times n_y \times n_z$ adjacent boxes. Each with size $l_x \times l_y \times l_z$. If we define the voxel size, we may find the number of voxels from the relation $n_i = L_i/l_i$, where n_i is the number of voxels in the i -th direction, L_i is the system size, and l_i is the voxel size. The remainder between L_i and l_i should be zero! One would probably like to define a maximum or a minimum size for the voxels, l_i^{\max} and l_i^{\min} . We can in that case calculate the number of voxels by the relations:

$$n_i = \left\lfloor \frac{L_i}{l_i^{\max}} \right\rfloor \quad \text{or,} \quad n_i = \left\lceil \frac{L_i}{l_i^{\min}} \right\rceil$$

where $\lfloor x \rfloor$ is the `floor`-function and $\lceil x \rceil$ is the `ceil`-function. The next step is now to find all the voxels in the system that does not contain any particles, and that makes up the void that we would like to fill. We should find the volume V , that every inserted molecule averages occupies from the molar mass M , of the molecule, and the desired density ρ . The relation is expressed in eq. (2.31), where N_a is Avogadro's number.

$$V = \frac{\frac{1}{N_a} M}{\rho} \tag{2.31}$$

The voxel size can then be used as the cube root of the volume. However, in practice we will not be able to fill the system with the desired density only by doing this, because we have to use an integer number of voxels. We would therefore like to only fill a fraction of voxels found from the ceiling-function, because these voxels are smaller or equal to the voxel size that is found indirectly from eq. (2.31). If we were to fill all voxels with a molecule, we would get the

density:

$$\tilde{\rho} = \frac{\frac{1}{N_a} M}{\tilde{l}_i^3}$$

where \tilde{l}_i is the actual voxel size we are using, found after the ceiling-function is used and an overestimate on the number of voxels is found. The number of voxels that we would like to fill is then given by eq. (2.32), where N is the total number of voxels.

$$\tilde{N} = N \frac{\rho}{\tilde{\rho}} = N \frac{\tilde{l}_i^3}{l_i^3} \quad (2.32)$$

We can now use eq. (2.32) to fill the voxels, by drawing a random number uniformly distributed between zero and one, and then fill the voxel if the random number is smaller than \tilde{N}/N . The molecule have to fit within the voxel, and needs to be placed within it so that it does not get too close to surrounding molecules. In most cases it is smart to start out with a density that is a littlebit too high, and then relax the system so that we do not produce any bubbles. Bubbles will arise if you start out with a correct density, and the density of the injected fluid is larger close to the material surface.

2.8 Diffusion

Self diffusion is the measurement of a particles mean squared displacement from its original position, and is measured in [m^2/s]. It says something about how easliy a particle move in its given surroundings. If you have a large diffusion coefficient, the particle moves statistically over greater distances per unit time than for a smaller diffusion coefficient. You might say that diffusion is a product of the thermodynamic properties of the material and material surroundings because the rotational, vibrational and translatory movements of the particles in the system is mixing them. It is these series of complementary events that is being refered to as Brownian motion. And it is therefore Browian motion, driven by thermodynamics, that is the reason for diffusion, and that results in mixing of particles.

In molecular dynamics simulations, we most often make use of periodic boundary conditions, and a particle confined within such a system reappears at the other side of the simulation box if it exits the system in some direction. Imagine that your friend is in a room, moving freely about, and that you are allowed to open the door at certain intervalls to observe your friends position. If you open the door at a high frequency, your friend will not have been able to move far from the last observed position. At a bit lower frequency, the person might have

been able to move across the entire room. At longer time intervals, you can not say if the person has moved across the room several times, you can only see the relative distance from the last position, and you are ignorant of the actual total displacement. This is how a periodic simulation box is as well, and to monitor the total displacement of a particle, we must add or subtract a system size if the particle moves out of the system boundaries. This method, known as the minimum image convention (section 2.2), is wise to take into consideration for periodic systems. If you do not, a maximal displacement will be reached, and the mean square displacement (MSD) will flatten out and become constant, that represent an unphysical behavior of the system. The flattening of the curve will be dependent on system size, temperature and the material properties.

Particles vibrate, therefore at a high frequency of observing the position, you will catch this behaviour, seeing that the total displacement over short time intervals are low because the netto displacement is not far from its original position. But at some larger time intervals, and a lower frequency of spotting the particle, we may see the ballisic range of the particle, where it moves in some general direction away from its original postion.

The plots in fig. 2.6 shows the MSD of bulk water, where in fig. 2.6a, different estimates on the diffusion constant is made using different ranges of the time for the linear regression. We see that the choise of time-range is imoprant for the estimate on the slope of the MSD, f_1-f_4 . It is perhaps a bit difficult, but in the first pico second, we see the effect of the high frequency vibrations and collisions between the molecules. After this, the MSD becomes fairly constant for this system¹⁰, as is better seen in the rightmost figure, where the MSD in x,y and z direction is also shown. In a bulk system, we expect the diffusion in every dimention to be very similar.

The mean squared displacement at time step t , is calculated according to eq. (2.33).

$$\langle \Delta \mathbf{r}^2 \rangle_t = \frac{1}{N} \sum_i^N (\mathbf{r}_{i,t} - \mathbf{r}_{i,0})^2 \quad (2.33)$$

The MSD of a particle with diffusion constant D , is expected to follow eq. (2.34), where d , is the dimentionality (for all points and purposes in this thesis, $d = 3$) of the system where the particle is moving. This equation is outlined in Einsteins

¹⁰Although the time here starts ar 500 ps, the 500 ps timestep were used as the origin for the MSD

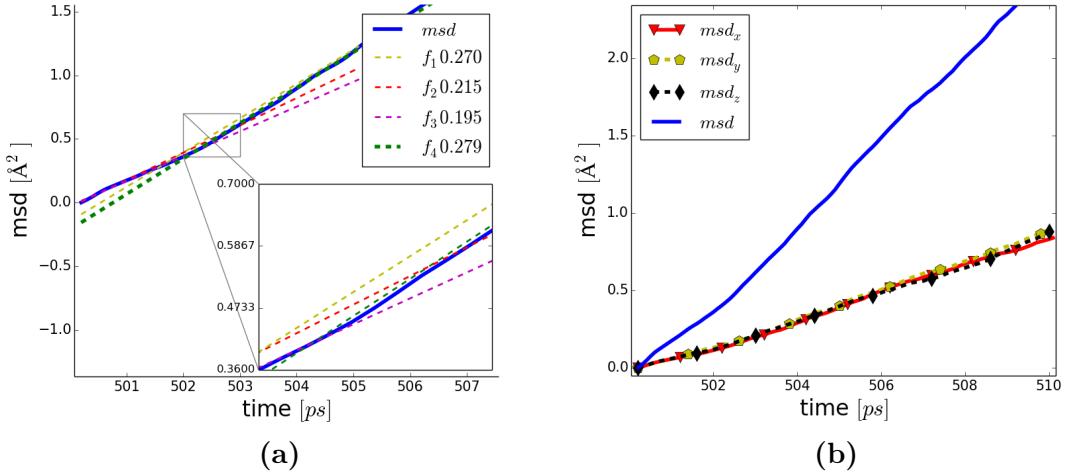


Figure 2.6: The diffusion of particles in a system is found through the mean squared displacement (MSD) of the particles as a function of time. In fig. 2.6b you see the MSD of oxygen atoms in bulk water as function of time for the x,y and z direction only, and also for the total displacement ($msd = \langle \Delta r_i^2 \rangle = \frac{1}{N} \sum_i^N (\Delta x_i^2 + \Delta y_i^2 + \Delta z_i^2)$). In fig. 2.6a, f_1-f_4 (given in $[10^{-8} m^2/s]$) is the slope of the linear fit to the MSD curve, and illustrates the importance of choosing the right time span for the estimate

work on Brownian motion [16].

$$\langle \Delta r^2 \rangle_t = d2Dt \quad (2.34)$$

We rearrange eq. (2.34) into eq. (2.35). This is the equation used to estimate the diffusion coefficient. It tells us that the diffusion D , should be constant in time for the mean squared displacement if we divide by a function of time, $f(t) = d2t$.

$$D = \frac{\langle \Delta r^2 \rangle_t}{d2t} \quad (2.35)$$

If we now look back at fig. 2.6, we noticed that the MSD were not linear at the very beginning of the curve. If we had sampled the positions at even higher frequency, we would have been able to see the timescale where particles hits eachother and moves rapidly in one direction before hitting another particle. This MSD will have a lower diffusion coefficient because the particles moves a lot back and forth, contributing to a low net displacement. But the transition from this regime to the time scale where the particles move away from eachother is easily spotted, and as you see from the estimates on the diffusion coefficient, which is just the slope of the MSD plot, the diffusion coefficient will be very different for the two.

2.8.1 Calculating the diffusion

The self-diffusion of particles in a system can be difficult to calculate when you add complexity to the calculation. But the inner loop for the basic calculation is quite simple. All particles needs to be tracked and their displacement calculated. We take the minimum image convention into consideration for a periodic system. This way we will avoid that the linear curve for the MSD flattens out when the displacement reach the system size. A python script of the inner calculation is shown in listing 2.1, where the initial position of the atoms is found in the `atoms0` matrix, and the positions of the particles at time T is found in `atoms`.

```

counter = 0
for i in range(Natoms):
    if (matrix['type'][i] == Type):
        ID = atoms['id'][i]
        ii = None
        for j in range(Natoms):
            k = atoms0['id'][j]
            if (k == ID):
                ii = j
                break

        dx = atoms['x'][i] - atoms0['x'][ii]
        dy = atoms['y'][i] - atoms0['y'][ii]
        dz = atoms['z'][i] - atoms0['z'][ii]

        dx,dy,dz = minimumimage(dx,dy,dz,Lx,Ly,Lz)
        msd += dx**2 + dy**2 + dz**2
        counter += 1

MSD.append(msd/(6*counter))

```

Listing 2.1: Calculating the mean square displacement. `minimumimage()` is a function for evaluating the minimum image convention

This way of calculating the diffusion for all particles in the system, we will get a plot of the MSD as a function of time. Notice that the MSD calculated in this code snippet is not the real MSD, but divided by six since the diffusion coefficient is to be calculated according to eq. (2.35), and since the dimensionality $d = 3$, we divide by six. We would ofcourse like to make several estimates on the diffusion coefficient to get good statistics and also get an estimate on the error. We sample the diffusion every 100 fsec and stop the sampling after 10 ps , then calculates the diffusion coefficient by approximating the curve by a straight line. The sampling time that you can allow yourself is dependent on the system size. How long does it take until a particle most likely have crossed the system? This

would also necessarily be temperature dependent. A larger system size will allow longer sampling times, but remember that we will not see from the MSD when the particles cross the boundary if we use periodic boundary conditions without the minimum image convention.

The system where we want to calculate the diffusion coefficient is equilibrated in the NVE -ensemble and the NPT -ensemble so that we know the density and radial distributions are all good before we run the simulation in the NVT -ensemble. The NVT -ensemble is the best for looking at the displacement since the system size does not change, and the temperature is kept at a average. The estimate on the diffusion coefficient is then done in two ways. The first method is to make independent estimates on the diffusion coefficient by approximating the mean square displacements that we get, by straight lines, and then take the average. Another way is to take the mean of the MSD by adding the curves together and divide by the number of curves, then estimate the resulting MSD by a straight line. These two methods does not necessarily give the same answer. If there is a drift in the estimates, this is most easily seen in the first method, but the lowest standard deviation will be obtained from the latter one. You might say that the first method is a microscopic method, and that the last one is a macroscopic method looking at the larger picture. In the latter case, it can be a bit difficult to make a good estimate on the standard deviation, but a method is described in Squires [54] under complicated functions, page 46-48. One finds the standard deviation in the MSD data set, and adds this deviation to the mean value and takes a new estimate on the slope. The absolute difference in the slopes is the standard deviation of the slope.

Calculate the diffusion as a function of distance to a surface

When analyzing the transport properties of a fluid material close to a solid, we would like to see how the diffusion coefficient is altered as a function of distance to the surface. The diffusion coefficient is then calculated at different distances from the surface. Imagine a system like in fig. 2.7, with boxes 0, 1, 2..., that divide the system into bins with hight dL , where bin j , is located within $[j dL, (j+1)dL]$. Where $x_{i,0}$, is the distance from the surface for particle i at time $t = 0$. As long as the particle is inside the same box as it was inside at $t = 0$, the MSD is calculated and added to the total MSD for that box. If a particle moves out of the box that it originated from, it is left out of the calculation because it would then (most likely) contribute with a wrong MSD to that box. So the particle to the lower right in fig. 2.7 will not be followed anymore since it left its box.

When we look at the diffusion of particles confined within a flat pore where we have two surfaces, a floor and a roof, we define the pore size L , as the distance

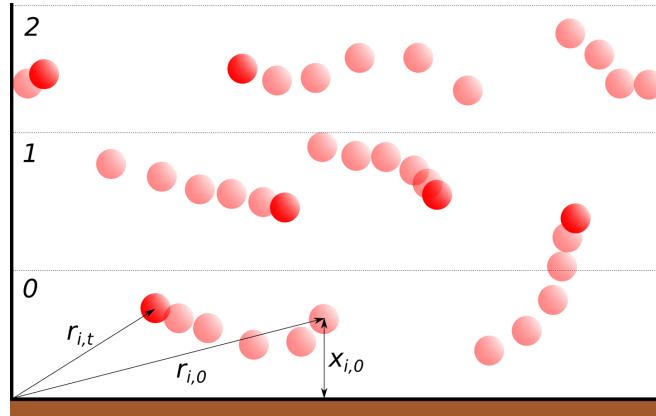


Figure 2.7: The diffusion of particles within regions $j = 0, 1, 2, \dots$, located at increasing distances from a surface located in the lower region

from one surface to the other¹¹. We define a box size dL , and divide the pore into $N = \left\lceil \frac{L}{2dL} \right\rceil$ boxes. After the `ceil`-function, the actual box size becomes smaller, and is calculated as $dL = \frac{N}{2L}$.

The distance $x_{i,0}$ from the surface for particle i , is calculated, and the particle belongs to a bin $j = \left\lfloor \frac{x_{i,0}}{dL} \right\rfloor$. We also have to count how many particles that belongs to every bin at every timestep so that we can perform the MSD calculation according to eq. (2.33).

2.9 Tetrahedral order parameter

The tetrahedral order parameter is a way of characterizing for example water, on how tetrahedral the water molecules is organized. Water in its crystalline form, ice, has a tetrahedral structure when looking at a molecule and its four closest neighbour molecules, as shown in fig. 2.8. The oxygen atoms are forming the tetrahedral structure, while the hydrogen atoms are facing the negatively charged oxygen, creating hydrogen bonds.

The tetrahedral order parameter is defined as eq. (2.36), where the angle θ_{ikj} is the angle between atom i and j for the structure parameter Q_k for atom k . As the equation for the tetrahedral order parameter shows, we have to find all angles within the tetrahedron that is spanned from the center atom k to atom i

¹¹It is not trivial how to define the location of a surface in molecular dynamics. We have defined the surface as the average x -position of the surface molecules

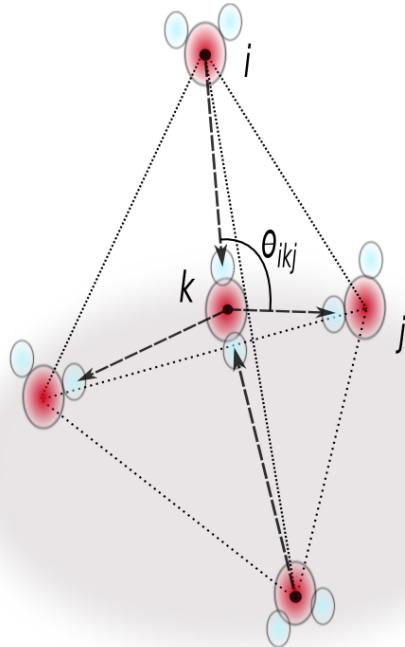


Figure 2.8: Tetrahedral structure of water. Hydrogen bonds are created between molecules

and j .

$$Q_k = 1 - \frac{3}{8} \sum_i^3 \sum_{j=i+1}^4 \left(\cos(\theta_{ikj}) + \frac{1}{3} \right)^2 \quad (2.36)$$

The first task is to find the four closest neighbours for every oxygen atom in the system. For every oxygen atom k , we therefore have to leap through all other oxygen atoms, that is, except itself, and find which ones have the shortest distance. Next we have to find the angle θ_{ikj} spanned by atom i , k and j . This is done by leaping over three of the closest neighbours i , and for the $j = i + 1$ other closest neighbours, we find the angle, and keep on doing the calculations according to eq. (2.36). For every tetrahedron (one for every oxygen atom in the system) we get a value for Q_k , which we place into bins ranging from -1.5 to 1.0 (the range is entirely optional, but this range were found to be nice for water). The tetrahedral order parameter of the system is taken as the mean value, but it is also interesting to look at where we get the highest occurrence of Q_k values. A large Q_k value represents a more tetrahedral structure.

2.10 Radial distribution functions

A radial distribution shows the occurrence of something as a function of distance. In molecular dynamics we calculate the radial distribution of particles as function of other particles, like the radial distribution of molecules or molecule clusters. We choose to denote the radial distribution of some material B , at a distance r from material A as $g_{AB}(r)$. So for every A , we count the number of B 's within some finite volume V_i located within the distance $dr = r_i - r_{i-1}$, like in fig. 2.9, where the distance r_i is i times dr from A . We use N such points so that the maximal distance we will search for B -particles, is at r_N from A . For every volume V_i , we create a bin i , that holds the number of particles B found in that volume. The distance between A and B is R_{AB} (remember to use minimum image convention for a periodic system).

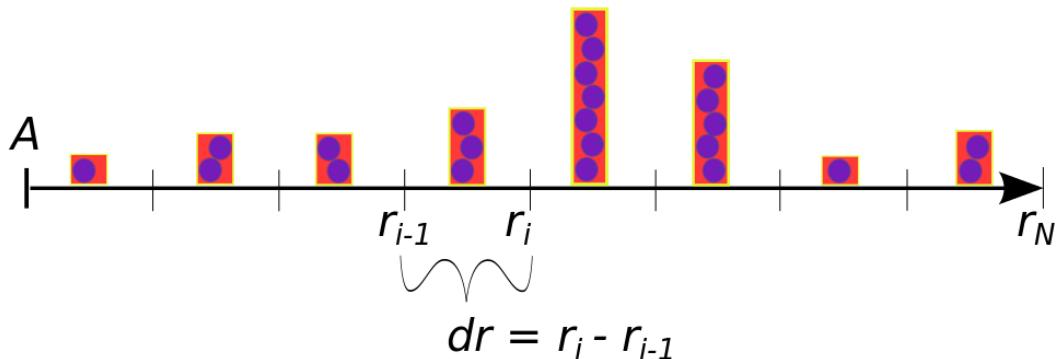


Figure 2.9: Slicing a finite volume around a point A into N equally spaced bins, and find the number of particles B , that lie within every bin i

image convention for a periodic system). Now, as in fig. 2.9, the search volume V around A is divided into N equally spaced bins which keeps the number of B 's found within $r_i - r_{i-1}$ (illustrated with bars filled with balls, as a way to think about how the bins are filled up one by one). To place a ball, or a particle B , in its correct bin, we have to find the relative distance R_{AB}/dr . Then round off this number, and we have the integer number of which bin the particle belongs to! This is done with the `floor`-function:

$$i = \left\lfloor \frac{R_{AB}}{dr} \right\rfloor$$

Which rounds a decimal number down to an integer number. If particles are located at distances greater than the search area you have defined, then these are not counted for. Now we have found the number of particles B , as a function of distance from the particles A . But it is common to display the radial distribution function as a probability density function, where we display the *probability*

of finding the particle B , as a function of distance to particle A . The volume $V_i > V_{i-1}$, obviously, so we also have to divide the number of particles in bin i , by the volume of this shell in addition to dividing by the number density of particles in the entire system! A short code snippet is shown in listing 2.2, where the normalization procedure is done using python syntax.

```
sqNumdens = Nmolec1*(Nmolec2/(Lx*Ly*Lz))
for i in np.arange(1,Ncontainers,1):
    V1 = (4.0/3.0)*pi*r[i-1]**3
    V2 = (4.0/3.0)*pi*r[i]**3
    dV = (V2 - V1)
    g[i] = (atomcontainer[i]/(dV*sqNumdens))
```

Listing 2.2: Normalization of the radial distribution. The atomcontainer holds the number of particles in bin i . We divide the number of particles in bin i , with the volume of that bin, and the atom number density

Without presenting the mathematical formula, we have now seen a way to calculate the radial distribution function numerically. The radial distribution function $g(r)$, shown mathematically in eq. (2.37), is a normalization of the function of the number of particles $n(r)$. The number density of particles is $\rho_v = N/V$, where N is the total number of particle pairs. Finally we have to divide by the discrete volume $dV = \frac{4}{3}\pi(r + dr)^3 - \frac{4}{3}\pi r^3 \approx 4\pi r^2 dr$.

$$g(r) = \lim_{dr \rightarrow 0} \frac{n(r)}{\rho_v} \frac{1}{4\pi r^2 dr} \quad (2.37)$$

The mathematical terminology were studied in Levine et.al. [30], and describes the radial distribution as a continuous function since we should let $dr \rightarrow 0$, which is in practice impossible.

2.10.1 Angular distribution

An angular distribution function can be used to show the distribution of intermolecular angels in a sample. For example to show how well a force field reproduce the $H - O - H$ -angle within the water molecule. The angles of every $H - O - H$ in the system should be calculated and binned just like it is done for a radial distribution, except that now, we are not interested in the radius between two atoms, but the angle spanned by three atoms. We therefore have to create a distribution $\theta \in [0, \pi]$ radians, and use N bins on this interval. This results in equally spaced bins of size $d\theta$. If we continue with the water example, we have to locate the two nearest neighboring hydrogen atoms to an oxygen atom. We then

need to calculate the angle θ_i of the i -th molecule before finding out what bin the measurement belongs to. Finding the bin is done in a similar manner as for the radial distribution where the index of the bin is found with the `floor`-function of the relative angle $\theta_i/d\theta$.

The radial distribution will give information about how static the angle of the molecule is. We know that the angle should not vary much for bulk water, but that the angle is altered in contact with molecules and at surface boundaries.

Chapter 3

Software and *LAMMPS* specifics

3.1 The *LAMMPS* software

A Large-scale Atomic/Molecular Massively Parallel Simulator, or *LAMMPS*[39], is a open-source software that was, and is still being, developed by Steve Plimpton, Aidan Thompson, and Paul Crozier. The code is distributed under the GNU Public License (GPL). The *LAMMPS* web site at <http://lammps.sandia.gov> has a lot of information about downloads, how to get started and how to use the code. At the webpage you can find tarballs to download, and a very informative manual for the software. When you start to get into the functionality and commands in *LAMMPS*, the easiest way to look up information is through a common browser search engine. But ahead of this, you should get to know some of the main features through reading the manual and some example input scripts that follows the latest *LAMMPS* tarball that you download from the Sandia web page. The main objective of this chapter is to give you information that is learnt through working with the software for some time. I will present to you some key notes on how to get started on an Ubuntu desktop, but other platforms are supported as well, so no worries! Further on, we are going to have a look at how the input script, used to tell *LAMMPS* what it is supposed to do, is build up. And we are going to explain how minimization procedures are carried out using the software. We will introduce *moltemplate*, a nice tool to build up the system that you are going to simulate, and we will talk briefly about visualization tools.

3.1.1 How to make *LAMMPS* executable

To make an executable you first need to download the latest *LAMMPS* tarball, and unpack it wherever you want. This is done by moving the tarball to that location, and, for manual unpacking, run the commands

`gunzip filename.tar.gz` and then

`tar xvf filename.tar`, since the tarball is packed using *gzip* and *tar*. Then `cd` into the *src*-directory and run:

`make package-status`, this will give you a list of available packages, and which one of them is active in the *src*-directory¹. *LAMMPS* is not prebuilt with MPI (Message Passing Interface), so make sure that Open MPI² is installed on your computer. It could be that it is prebuilt on your computer and you should not overwrite it with a new version. Try for example to find information about the compiler `mpirun` by typing `mpirun -version`, which will, if it exists, list version information. Also you may run `whereis mpirun`. If you need to install it, you may download the latest tarball from the MPI webpage and extract the files, then run the config-file inside the download, using the command

`./configure --prefix=/usr/local`, this will install MPI in `/usr/local`. Next you will have to run the command `make all install`.

Another library that you should consider if you want to use the kspace style *pppm* - particle-particle particle-mesh solver, is the *FFTW*-library³. *pppm* maps atom charge using FFTs to solve the Poisson equation and find the long range coulomb contribution for every particle in the system. The *pppm* method is almost always a faster solution than using *ewald* summation, and the *FFTW*-library provides the fastest Fourier solver method.

When you have read all this, it will seem like there is a lot to do, getting started with *LAMMPS*, but that is actually not the case. There is a lot of “nice to know”, and that is why I mention these two things before letting you know how to make a *LAMMPS* -executable.

Making a *LAMMPS* -executable is done in the `/src` -directory in the *LAMMPS* -folder where you may run the `make` -command. In general you run the correct makefile by typing `make <machine>` for whatever `<machine>` -operating system you are running on, and is supported by *LAMMPS*. If you are on an Ubuntu desktop, just run `make ubuntu`. If the compilation did not create an executable (i.e. there does not exist an `lmp_ubuntu` file after the `make` command is done) you

¹You can get a list over commands that is possible to run just by typing `make` or `make package`. The packages that you should include, is the *KSPACE* (enables long-range interaction through the `kspace` -command), *MANYBODY* (default), *MOLECULE* (bond and angle pair styles, impropers and dihedrals) and *MPIIO* (needs to be build if you want to write and read restart files in parallel)

²Open MPI web site: <http://www.open-mpi.org/>

³<http://www.fftw.org/>

may perhaps need to create your own makefile, the steps for this are described in the *LAMMPS*-manual. You can check available makefiles in the `MAKE`-directory located within the `src`-directory.

To make a *LAMMPS* executable on the Abel computing cluster⁴, is just as simple! See section B.4 for a step-by-step approach on how to compile *LAMMPS* on the super computer.

3.2 The input script

Your executable needs an input script! And they are typically executed with a command like explicitly shown in listing 3.1. This command line⁵ code use *mpirun*⁶ to run the executable *lmp_ubuntu* code in parallel on $n = 4$ CPU's. The executable needs an input script written with *LAMMPS* commands to tell the executable what to do. An example input script is shown in the appendix listing B.3.

```
you@machine:$ mpirun -n 4 ./lmp_ubuntu < in.lammps
```

Listing 3.1: Example on how to execute a *LAMMPS*-executable with a given input script

The input script can, as I see it, be split into three parts. The *header*, which gives information about the system that you want to simulate. The *computes* that you want to do during the simulation, and finally what *run time* options that should be used during the simulations. Another important part is to create *restart* files and *data* files that can be used to continue the simulation from where you stopped.

3.2.1 The header

The beginning of the input-script, I call the header, since a header in computing often refers to the top of a file where vital information is provided. In the *LAMMPS* input script header, we provide information like boundary conditions,

⁴Abel computing cluster <http://www.uio.no/english/services/it/research/hpc/abel/>

⁵If you like to know more about how to use the command line, I suggest reading the web book *The command line crash course*: <http://cli.learncodethehardway.org/>

⁶*mpirun*, *mpiexec* and *orterun* are synonyms and will produce the same behavior

units convention, atom style (structure of input *.data*-file) and k-space style (long range electrostatic correction). How often neighbor lists are created is usually given at this stage, and you may provide one *.data*-file in the input script through the `read_data`-command. The *.data* file contain information about the particles that are present⁷. Particle types, charges and positions, what molecules they are part of, what particles that are bonded, that is, what atoms that should have bond potentials, and what particles that should have angle potentials. Also dihedrals and impropers can be included in this file. This is just some of the information about the particles that you may include in the input file. But you may also give all this information in the input script. Some lines usually put into the header is shown in listing 3.2.

```

boundary      p p p
units         real
atom_style    full
pair_style    lj/cut/coul/long 10.0
kspace_style  pppm 0.0001
neigh_modify  every 1 delay 0 check yes one 3000
pair_modify   tail no
read_data     "SiO2_unit_cell.data"

```

Listing 3.2: Header example

3.2.2 Computes

Here you specify what kind of calculations that you want *LAMMPS* to carry out during the simulation. It is possible to use global variables and create your own variables to perform computes during the simulation. An example could be to perform the computation `compute myTemp all temp/com` which computes the temperature of all particles in the system and corrects for the center of mass for the particles in the group. The computed variable is stored in the compute-ID `myTemp`, and can be accessed within the script with a `c_`-prefix, e.g. `c_myTemp`. For example can a `fix ave/time` be used to average global variables over time and write the information to a dump file. In listing 3.3 I demonstrate a way to calculate the density of water on the fly by counting the number of molecules in the group that contains all water molecules. The volume of the system is fetched, and the density is calculated as $\rho = N_{H_2O} M_{H_2O} / N_a V$, where N_a is Avogadro's number. The resulting units for this calculation of the density will be in g/cm^3 , assuming that the `units real` command is invoked.

⁷How to create system data files will be discussed in the *moltemplate* section

```

variable No equal count(spce)      # number of water molecules
variable MH2O equal 18.01528     # Molar mass of H2O [g/mol]
variable Volume equal vol        # [Å3]
variable waterDensity equal ${No}*${MH2O}/(0.6022*v_Volume)

```

Listing 3.3: Compute density example

In listing 3.3, the `count` function counts the number of elements in a group named *spce*, the group is named using a command like `group spce type 1 2`, which defines the group name for all particle types 1 and 2. Further can some special values, called *thermo keywords*, be accessed directly, like `vol`, `pe`, and `xlo`. There are two ways of accessing variables, one is with the dollar sign and curly braces, which gives access to the initial value of the variable. The other is by the `v_-`-prefix, which returns the value of the variable on the given time step that it is accessed.

3.2.3 Run time

During run time, there are many things that you would like to specify. The time step in the simulation has to be set, also you would like to specify what, and how often information is printed to the screen with the `thermo_style` and `thermo`-command. Notice also here in listing 3.4 how one may access a variable and print it to screen using the `v_-`-prefix. In the `dump`-command, the name of the `.dump`-file is accessed calling `${Dump}` when we have earlier set a variable name to something like `variable Dump string nameofdump.*.dump`

Other important run time setting is to assign particle velocity from a random distribution, and to *fix* the ensemble. Here illustrated in the code snippet by fixing the NVT-ensemble. The number of time steps for the simulation is set by a simple `run`-command. Last, but not least important ting to put at the end of the input script is a `write_restart`-command that creates a binary restart file that can be read by *LAMMPS* on the computer where the binary file was made. In theory it should be possible to transfer such restart files to other similar machines with the exact operating system, but this is not recommended. In stead a `write_data`-command is recommended. This command writes the resulting state to a `.data`-file.

```

timestep      2.0          # fsec 10E-15s
thermo_style  custom step temp press ke pe v_waterDensity
thermo        100          # print output
dump          DUMP all custom 100 ${Dump} id mol type x y z

```

```

velocity      all create 300.0 70590 dist gaussian
minimize      1.0e-6 1.0e-6 10000 10000
fix           SHAKE all shake 1.0E-4 100 0 b 1 a 1
fix           NVE all nve temp ${Tstart} ${Tend} ${tc}
run           5000      # number of time steps

write_restart water.restart
write_data    water.data

```

Listing 3.4: Run time example

There are many possible failures, but the only way to succeed is to try. Most error messages are easily understandable, and all executions should be tried out for few time steps at first to make sure that the program will terminate in an expected fashion, returning all the wanted files. Good luck with that!

3.2.4 Minimization procedures

The total potential energy of the system is evaluated by adding up the potential terms for all particles in the system. The potentials are evaluated according to N atom coordinates, as shown in eq. (3.1), depending on the model. For example a pair term is evaluated by comparing the distance between a pair of particles, while the angle term obviously needs three particles to find an angle.

$$E(r_1, r_2, \dots, r_N) = \sum_{i,j} E_{pair}(r_i, r_j) + \sum_{i,j} E_{bond}(r_i, r_j) + \sum_{i,j,k} E_{angle}(r_i, r_j, r_k) + \text{other} \quad (3.1)$$

Measuring the potential energy of the system directly is in itself not that interesting, other than for observing phase transitions or that the system suddenly goes into a more stable or a more unstable state. Rapid changes in the potential energy while the kinetic energy stays at a stable stage suggests a phase transitions, since heat is added or released from the system, but the temperature remains the same. A ground state energy of the system is found when we cool the system down to absolute zero, and there is almost no dynamics left. An improved geometrical structure can be validated by looking at the potential energy at this state and compare with other geometrical structures of the same material, but in all these cases we look at the potential energy relative to other similar systems, or relative to the system itself that we are studying.

When we want to find the energy associated with some phenomenon, for example the binding energy of a molecule to a surface or the surface energy of a material, we are interested in the ground state energy of the systems and to compare these. A system is said to be in its ground state when it has its lowest

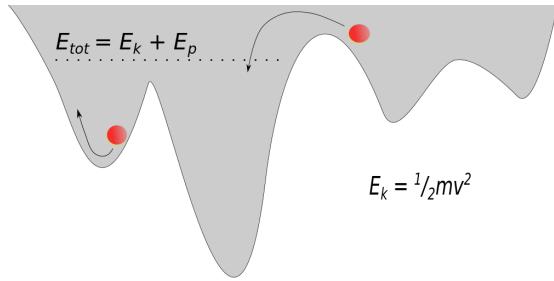


Figure 3.1: Illustration of a one dimensional potential surface. The particles in the potential field is restricted by their kinetic energy on how high in the potential field they can move

possible energy, and the geometrical structure of the system is at this moment most often the one we would find in nature. How come that this is the case? Why is it so that nature strives towards a lowest possible potential energy? I do not have a better answer than that this is because nature seems to seek energy-minimization, but we can easily imagine a potential field like the gravitational, where all particles are drawn towards the center of the field in which the potential energy is at its lowest. If we take a look at the relation between the potential energy and the force on a particle affected by the potential field, there are no force working at a local minima that can contribute to kinetic energy. Nature has an ocean of time to relax the system and bring it to a local minima. Temperature fluctuations are important to bring a system out of a local minima and over to another and perhaps deeper potential well, like illustrated in fig. 3.1. When we are using molecular dynamics, we are limited to the nanosecond timescale. Therefore we do not have the time to relax the systems as perfectly just by waiting, but we know that minimizing the potential energy means to bring particles to the bottom of the potential well. This can be formulated mathematically, for example by a steepest decent algorithm, moving the particle step by step proportional to the negative gradient (in the direction of the force). This method is the simplest one, and will bring the system safely towards the local minima. For a potential field $V = V(\mathbf{r})$, it follows that if:

$$\mathbf{b} = \mathbf{a} - \gamma \nabla V(\mathbf{a}) \quad (3.2)$$

for a positive γ that is small enough, then we have that $V(\mathbf{b}) < V(\mathbf{a})$. Other more complex methods are developed like the conjugate gradient method that normally converges faster⁸, the conjugate gradient method is the weapon of first choice in *LAMMPS* (default), while other methods like the Hessian-free Newton algorithm and the steepest decent can be used instead with the `min_style`-command.

⁸The conjugate gradient method can be studied in detail at <http://www.cs.cmu.edu/~quake-papers/painless-conjugate-gradient.pdf> (available 27.03.2015)

A problem that we meet when we want to minimize the energy of a system, is that we never know if we have reached the global minima, or if we are at a local one. To escape local minimas, the system needs to be warmed up and minimization procedures performed, and then the system needs to be cooled down again. Preferably in steps, so that we do not get caught in a new local minimia. The only way that a particle in a potential field can get from a local minima to another is if it has enough kinetic energy to escape (this is by the way not entirely true according to quantum mechanics, but in a molecular dynamics model such quantum mechanical phenomena is not possible to witness). If we rise the temperature in the system, and then minimize the energy, then we will hopefully find the a global minima. We then have to cool the system so that it falls into the geometrical structure that this potential well suggests is the structure that minimize the force per particle in the system. To minimize the force per particle is essentially the same as minimizing the potential energy. There are of course many ways to do this, but a method that I developed using *LAMMPS*, seems to be working quite fine. The system starts at a given temperature T_i , using the Berendsen thermostat, and is equilibrated in the NVE -ensemble before it goes into a loop where the temperature is gradually reduced by one degree for every run in the NVT -ensemble. The system energy is then minimized, and then the process restarts, causing the temperature to rise very fast to the initial temperature. A code snippet of the minimization procedure is shown in section B.2, where the procedure is explained in detail. To warm up a system too much will lead to a large disorder that will take much longer time to equilibrate and bring towards a minima, this should therefore be avoided. Only small temperatures are needed.

3.3 Building molecular structures using moltemplate

Moltemplate is a program developed for building molecules and complex molecular structures for *LAMMPS*, and is used to hold information about the elementary building blocks and how to combine these into complex systems. The idea is to build up any system from small, but essential building blocks at first, then combine and reproduce them to build larger and more complex systems, and produce readable input files for *LAMMPS*. Therefore, the syntax and structure of the elementary building-block-files in moltemplate are recognizable for someone familiar with the *LAMMPS* software. What you have to do, is to build these essential blocks, defining for example a water molecule, and then you may use this file to reproduce the water molecule and place them into some grid formation that you create. The file is a *LAMMPS* template file, using the extension *.lt*, and

the file holds all text relevant information about the atoms that forms a molecule or a unit cell for some crystal or clay structure. The most important information that needs to go into an *.lt*-file is listed below, and an example of a *.lt*-file for SPCE -water is shown in listing B.1.

1. *units* - what units is used in the system, according to *LAMMPS*
2. *atom_style* - defines the structure of the *data* -file
3. *pair_style* - what potentials are used
4. *bond_style* - what bond potentials are used
5. *angle_style* - what angle potentials are used
6. *Atoms* - initial positions of atoms
7. *Masses* - atom id and their masses
8. *Group* - what atom types is within a defined group
9. force field bonds and angles between the atoms
10. force field pair and angle coefficients

These are at least the most important things that have to be defined for the atoms and the structures that they constitute. Another similar program to create molecular stuctures, called *pyMOL* were tested at first, and were interesting because the *pyMOL* software can be used both to create the initial system, and also for visualizations in *3D*. It is written mainly in python, and is an open-source code like the moltemplate software, but the organizing and structure of the different bond styles and so on were found to be too far away from the *LAMMPS*syntax, and therefore proved to be difficult and time consuming to learn. Since example moltemplate *.lt* - files were possible to find on the internet, it was also fairly easy to understand the syntax of the files. Example files for *moltemplate* can be found at my github account⁹

The installation of moltemplate is also very simple. If you already have downloaded *LAMMPS*, moltemplate will be located in the *tools* -folder. I would recommend to read the manual for moltemplate to make sure that you update the *PATH* environment variable and set the *MOLTEMPLATE_PATH* environment variable to point at the common directory in the moltemplate folder [25]. When this is done, you should be able to run a command like 3.5 from any folder

⁹goranbs@github: <https://github.com/goranbs/Master/tree/master/moleculetemplates>

in a terminal to create four files¹⁰, including the *.data*-file that holds the molecular system information.

```
you@machine:$ moltemplate.sh -atomstyle "full" nameoflfile.lt
```

Listing 3.5: How to run the moltemplate bash script from a command line

```
you@machine:$ moltemplate.sh -vmd nameoflfile.lt
```

Listing 3.6: Run moltemplate and visualize immediately in vmd

When the files are generated, you would really like to display the system (the information in the *.data*-file) in a visualization program like *VMD* or *OVITO*. *OVITO* cannot read the data file directly, but *VMD* can with an extension tool called *topotools*. The simplest way to manage this, is to run moltemplate with a “-*vmd*”-option, that immediately visualize the state after a successful execution of moltemplate. The command that does this fantastic operation is shown in listing 3.6. To make a readable for *OVITO* is more demanding, and I only see two options. Either you have to run a simulation with *LAMMPS* and generate a output file with the *dump* -command, like shown in listing 3.7,

```
dump ID group custom ${Nprint} ${dumpState} keywords
```

Listing 3.7: LAMMPS dump command

where *ID* is the group ID, which is most commonly set to: *all*. The *custom*-option, is the dump style where you specify what kind of information that you wish to write to the dump file. The following options states the number frequency of what time steps information is printed to file and the name of the dump file respectively¹¹. The *keywords*-option is a user-defined combination of thermal keywords, as for example, *id*, *mol*, *type*, *x*, *vx*, *fz*, *pe*. These keywords stand for atomID, moleculeID, atom type and x-position respectively, *vx* is velocity in x-direction and *fy* is force component in the specified y-direction. *pe* is the potential energy of the system. The values are printed in the desired order

¹⁰Four files are generated, *.data*, *.in*, *.in.init*, and a *.in.settings* -file. The *LAMMPS* input script contain force-field styles and coefficients, run settings and so on, while the data file contains information about the number of atoms, number of bonds, angles, dihedrals and the masses of the different atoms. It also contain information about the particle locations, what atom is bonded with what and so on, like in the example *.data* -file found in listing B.2

¹¹Remember that using the dollar symbol and curly brackets, *{}}, refers to a predefined variable in *LAMMPS*, in which you access the initial value it was set to*

in the dump file according to the custom option in the `dump`-command.

The other way to display the initial system in *OVITO* is obviously to write your own program that reads a `.data`-file and that produces a `.dump`-file, which is readable to *OVITO*. If you whish only to use *OVITO*, this is the long term best solution. In the next section, I will give you a brief introduction to *OVITO* and *VMD*, just explaining a little bit about them, so that you have a better starting point for coosing one of them. The reason for mentioning these kinds of tools is also to point out the absolute necessity of visualizing the system. These kind of visualization programs is necessary to be able to do debugging at a quicker pace, look for unphysical behavior and to get a better understanding of the dynamics of the system.

3.4 Visualization

It is important to visualize the system to see how the particles behave. This is important because we can get an idea of how the particles interact and move under the given conditions. The visualization may represent our mental image of the atomic world, and is an important method of learning and teaching molecular dynamics because it is easier to explain what you see if both you and the receiver have the same idea about the system.

Molecular dynamics, molecular statics and Monte-Carlo based simulations are standard methods of modeling material structures at an atomic scale, and the dynamics of the particles is a key factor to the macro-scale behavior of the gas, fluid and solid materials that are simulated. Since the modeled systems are consisting of tens of thousands, hundred thousand and even up to millions of particles, it is therefore most important that the visualization tool can load a large amount of data, and be able to render surfaces nicely where point particles have their location.

3.4.1 *OVITO*

*OVITO*¹² is an Open Source visualization software under the GNU General Public License, written in C++ 11 and runs on all major operating systems, including Microsoft Windows, MacOS and Linux. It is very easy to use because of its pure graphical user interface, and a great way of getting started with visualization. *OVITO* have also got its own *Python* interpreter, which can be import into a

¹² *OVITO* webpage <http://www.ovito.org/index.php>

python script. This way you may load scenes, do fancy rendering and produce pretty visualizations and make computations based on the input dataset.

3.4.2 *VMD*

Visual Molecular Dynamics¹³ (*VMD*) is another *Open Source* visualization software that can be downloaded free of charge. *VMD* runs on Mac OS X, Unix, Windows platforms, and is created to model and visualize micro-biological systems such as proteins and enzymes. *VMD* can handle many different file formats, and also be used to transform from one type of file format to another. *VMD* is more complicated to use than *OVITO* and it can be difficult to use its graphical user interface. The real power of *VMD* comes with its command line interpreter. It takes some time to get into, but is by many scientists seen as the best visualization tool when you just get to know it.

¹³VMD webpage <http://www.ks.uiuc.edu/Research/vmd/>

Part II

Simulations

Chapter 4

Benchmarking

4.1 Benchmarking, reproducing natural behavior

The aim of benchmarking is to make sure that the results are accounted for, and to get awareness of possible flaws or inaccuracies. We have to make sure that we are able to reproduce already obtained results for important constituent parts of the method. For the sake of molecular dynamics simulations, it is to make sure that the MD simulations reproduce natural behavior of the materials that is used in the numerical experiments. It is important to check that each component used in a numerical simulation works as expected in a given environment and under given conditions. For example we will check that the materials have satisfactorily radial distributions of atoms, in other words, that the intermolecular distances and angles are correct. We will check for other important properties like density and the diffusion coefficient.

The results are compared with the natural properties found from lab experiments and the results found from other computational simulations on the material ¹

In this part of the presented simulations, we are going to have a look at bulk water, bulk carbon dioxide and on portlandite. Benchmarking on quartz were not conducted at a very organized way as the structures of the material were given from an external science group located at *Forskningsparken, Oslo, Norway* that had already done the benchmarking and only wished to gain the results from my molecular dynamics simulations of the structures. Of course all systems were monitored during the simulations to see if there were any anomalies, and I was guided in the study of the quartz systems by post doctor Alexandru Boțan.

¹Thermodynamic properties for many different materials can be found at NIST <http://webbook.nist.gov/chemistry/>

Table 4.1: SPCE radial distribution of bulk water. x , values at stippled line. +, values at first maximum. † values obtained in my simulations. * benchmark values[33]

	O-H x	O-O x	H-H x	O-H +	O-O +
Svaland†	0.99	2.74	1.62	1.76	2.74
Mark*	1.00	2.75	1.62	1.75	2.75

4.2 Bulk water

Simulations of bulk water were carried out using the SPC/E water model (see section A.1.2 for more information regarding the force field). The shake algorithm were applied so that a time step $dt = 2.0 \text{ fsec}$ could be used [33]. The shake algorithm makes the angle in the molecule rigid and estimates the high frequency vibration energy, allowing for a much longer time step. Ewald summation with 1.0E-4 accuracy, is used to find the long-range Coulomb interaction correction term. A cut-off radius at 12 Å is used for the Lennard-Jones pair-style with long-range Coulomb interactions.

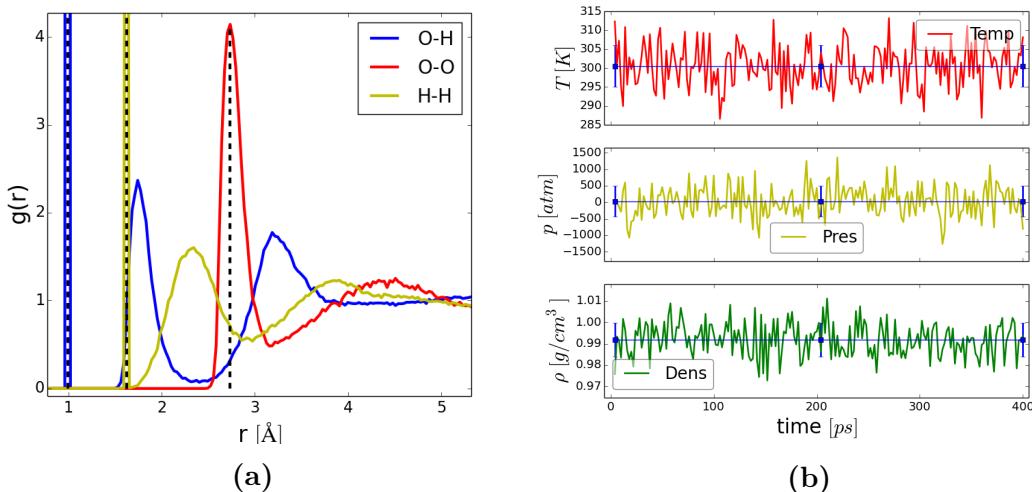


Figure 4.1: Radial distribution of oxygen and hydrogen in bulk water shown in fig. 4.1a. Temperature, pressure and density of the system is shown in fig. 4.1b

The radial distribution of hydrogen and oxygen in the bulk water is carried out in the NVT ensemble at constant temperature at 300K and 1 atm pressure according to fig. 4.1b. The density of the system is measured to 0.993 g/cm^3 which is very close to the correct density of bulk water under these conditions.

The pressure varies a lot around the mean, but is expected due to waters incompressibility and the fact that we are looking at a relatively small system with only 1000 molecules.

The tabulated results of the radial distribution is shown in table 4.1, where we have that x , correspond to values found at the stippled line in fig. 4.1a, whereas $+$, is the fist maximum value after this. The values of \dagger is found with a deviation of 0.03 \AA . The deviation of $*$, is unfortunately not given. The first maximums for the O-H and the H-H distribution is left out of the reference measurements because they are so obvious in the rigid water models. They are shown here, as two very narrow peaks for the blue and the yellow line, suggesting the intermolecular distances in the water molecule.

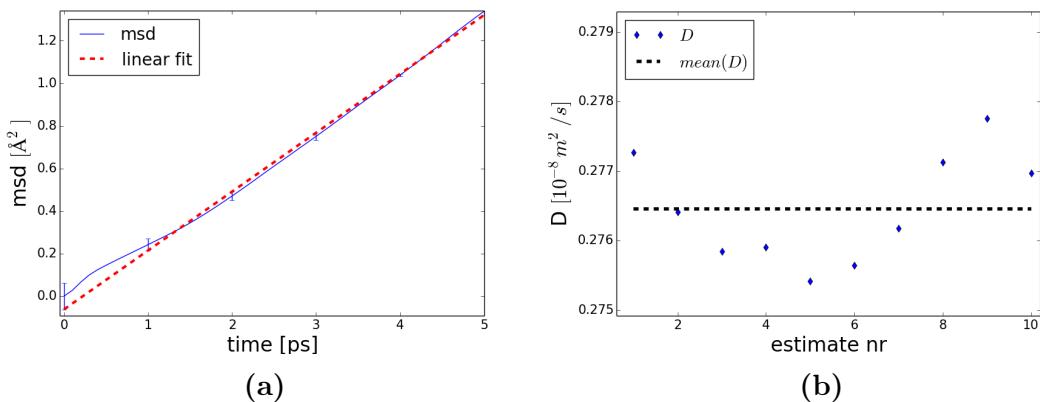


Figure 4.2: Estimates on the diffusion coefficient for bulk water. (a) shows an average of the sum of mean square displacements. This average is used to make an estimate on the diffusion coefficient D_1 (linear polynomial fit to line), while (b) shows estimates on the diffusion coefficient using several origins, D_2 .

The diffusion coefficient is considered as a good benchmark measure for the force field models because the coefficient is very sensitive to geometric stability of the molecule, charge distribution and interaction with surroundings, basically everything. Water models that arrives as poor diffusion coefficients, but are still being used, is mainly because the model is good at reproducing natural behavior at different temperatures and pressures, or that the transport property is not essential for the study as other properties.

Water is known to have a diffusion coefficient of $2.3 \text{ E-9 m}^2/\text{s}$ [33]. For different water force field models, different diffusion coefficients are obtained for bulk water. The SPCE model comes closest in a study by Mark et al.[33], where $D = 2.8 \text{ E-9 m}^2/\text{s}$. But even closer comes the SPCE model in another study conducted by Bourgh et al.[5], with a simulation of bulk water with only 216 molecules, they arrive at $D = 2.4 \text{ E-9 m}^2/\text{s}$, but with a large standard deviation

Table 4.2: Diffusion of bulk water using the SPCE water model. SPCEa and SPCEb is benchmark values[33]. SPCEa being the result using one part of the MSD slope, and SPCEb another

	D_1	D_2	D_{SPCEa}	D_{SPCEb}
$D [10^{-8} \text{ m/s}^2]$	0.2765 ± 0.0027	0.2765 ± 0.0008	0.27 ± 0.012	0.28 ± 0.006
N	24000	24000	820	901
$\rho [\text{g/cm}^3]$	0.993	0.993	1.008	0.998
$T [K]$	300 ± 2	300 ± 2	301 ± 4.4	298 ± 1.4

of $0.4 \text{ E-9 m}^2/\text{s}$. SPCE water model is known to reproduce transportation properties acceptably well, so therefore we choose to use the SPCE model for water throughout this thesis. The results on the diffusion coefficient on the bulk water using the SPCE model is shown in table 4.2 along with the benchmarks.

When estimating the diffusion coefficient, it is important to look at the results and try to make the estimation where the curve is close to a straight line. When it is not, we see that the standard deviation and the resulting estimate on the diffusion coefficient differs more for the two ways of performing the estimate. The results from the two methods is displayed in fig. 4.2. To the left (fig. 4.2a), which shows the average mean square displacement for ten estimates, the resulting mean square displacement seem to follow a straight line. But in the right plot, we get a better understanding of the deviation from one estimate to the other. The estimates in fig. 4.2b are obtained by plotting the MSD for 10 ps and calculate the diffusion coefficient, then another estimate is done for the next 10 ps and so on.

The tetrahedral order parameter of water were estimated to the average value of 0.56, while the relative occurrence of the tetrahedral order parameter 0.79 were the highest value obtained. The binning for the tetrahedral order parameter were 0.015, and the standard deviation in the mean value were 0.26. This large standard deviation is not reduced with the number of calculations, because we allways get a distribution like shown in fig. 4.3. The estimate is comparable with results obtained by Sund[55], where the the values 0.5 and 0.75 were found for bulk water at 300 K .

4.3 Bulk carbon dioxide

Three force field models have been tested for bulk carbon dioxide. The *TrPPE*, *EMP2* and the *FPP* which can be read more about in section A.1.3. The main difference is that the *TrPPE* and *EMP2* force fields have rigid angles, while the

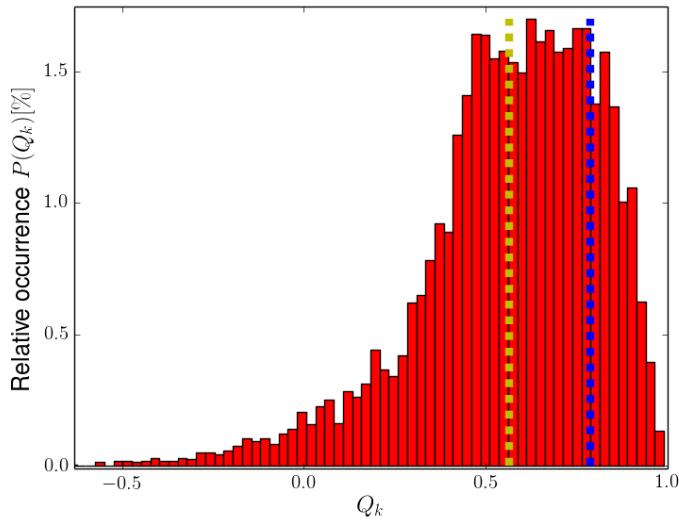


Figure 4.3: Distribution of the tetrahedral order parameter for bulk water at 300 K and 1 atm pressure. Yellow stippled line shows the mean value, while the cyan colored shows where the highest occurrence were found

FPF model is flexible. The results of simulations of bulk carbon dioxide using the three different force fields are presented here. Intermolecular distances, density and diffusion is the properties that are studied.

All models started out at the correct density of carbon dioxide at 300 K and 1 atm pressure, namely $\rho_{exp} = 1.7966 \text{ E-}3 \text{ g/cm}^3$ (according to NIST²). The systems were then allowed to equilibrate in the NPT ensemble for half a nanosecond using Nosè-Hoover. The result is presented in fig. 4.4. We see that the rigid models overestimates the density while the flexible underestimates it. The rigid models are closer, but seem to drift away from the experimental value, this fact can easily be seen by making an estimate on the slope for the density curves. We also notice that the flexible model has a lower standard deviation, which is numerically presented in table 4.3. The table provides results from radial distributions, the density and the self diffusion of bulk carbon dioxide. Notice how the diffusion coefficient for the rigid models are similar, but that it is different for the flexible model. The diffusion coefficients are high, which suggests that the molecules move over large distances per unit time. In heavy oil systems, the diffusion coefficient of carbon dioxide is measured to be around $3.5 \text{ E-}8 \text{ m}^2/\text{s}$ at about 300 K and 23 atm [60], and in another publication [44], CO_2 in air at normal temperature and pressure (295 K , 1 atm) were found to be $1390 \text{ E-}8 \text{ m}^2/\text{s}$.

²Experimental data from National Institute of Standards and Technology: <http://cccbdb.nist.gov/exp2.asp?casno=124389>

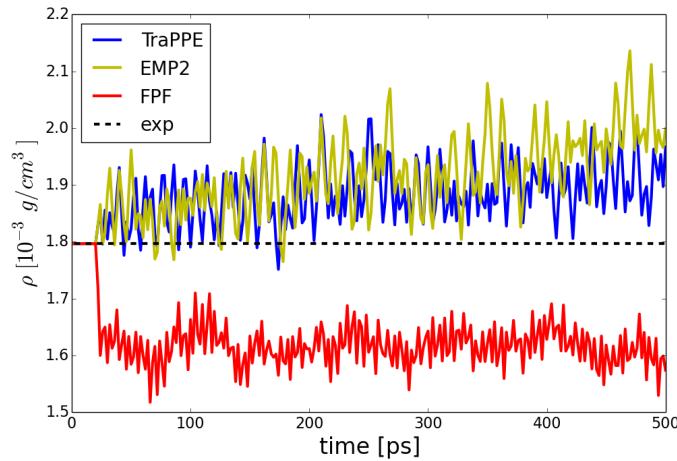


Figure 4.4: Time evolution of the density of bulk carbon dioxide for the three force field models

Table 4.3: Benchmarks for carbon dioxide. Values found at 300 K and 1 atm pressure. Radial distributions for TraPPE, EMP2 and FPF found with precision $dr = 0.03$. Comparing with experimental data in CO_2 . *The angular distribution were not calculated

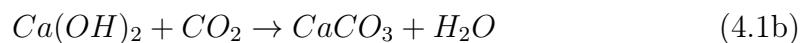
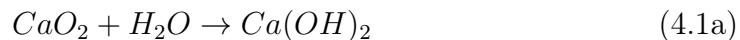
	TraPPE	EMP2	FPF	CO_2
$g(r)$ C-C [\AA]	3.43	3.55	3.61	-
$g(r)$ O-O [\AA]	2.31	2.41	2.31	2.324
$g(r)$ C-O [\AA]	1.14	1.17	1.14	1.162
$\alpha(r)$ C-O [rad]	180	180	180*	180
ρ [$10^{-3}\text{g}/\text{cm}^3$]	1.90 ± 0.05	1.95 ± 0.06	1.62 ± 0.03	1.7966
D [m^2/s]	139 ± 7.5	138 ± 5.9	125 ± 6.4	-

These values tell us that the surroundings are extremely important.

4.4 Portlandite

Calcium hydroxide $\text{Ca}(\text{OH})_2$, is an oxide mineral which got its name, portlandite, because it is a common product from an hydration process used to produce Portland cement. In the cement production, limestone and other clay minerals are heated to about 1700 K and grinded to a powder known as clinker. The clinker consists mainly of CaO_2 , SiO_2 and iron and aluminum compounds. Usually some form of CaSO_4 , calcium sulfate, is added to the clinker. When water is added to the clinker we get the reaction described in eq. (4.1a), where portlandite is formed.

When the excess water evaporates, the carbonation process of the portlandite starts, and we get the reaction described in eq. (4.1b), where carbon dioxide from the atmosphere and water is included in the forming of the calcite. This is called a *non-hydraulic* process as it hardens by carbonation and requires that the cement is exposed to air. But Portland cement is an hydraulic cement that needs no air exposure, it can harden under water, and much of the portlandite then remains as a binding material³.



Under ideal conditions, portlandite crystallize in hexagonal layered structures. During the hydration process in the cement, it turns out that it forms more massive and irregular shapes, but that its cleavage⁴ is preserved [57]. Portlandite can be seen in some volcanic and metamorphic rocks. Since it is known from cement chemistry that portlandite can bind carbon dioxide and form the carbonate calcite, $CaCO_3$, is it also interesting to see how carbon dioxide and water behaves in nanopores of this mineral. The material is wetting. That is, water is attracted to the surface. This can be seen at the molecular level when water molecules have their oxygen atom facing towards the surface. If hydrogen had been facing the surface, it would have produced a surface film interface like on a water droplet. Calcium hydroxide is known for its alkaline properties, giving twice as much contribution to the pH per mol than caustic soda. It is used in dental applications and hair products.

A unit cell of portlandite is displayed in fig. 4.5, where we see the structure in the xy -plane, with the z -axis pointing out of the paper . When many such unit cells are placed into each other, the beautiful hexagonal structure spanned by the calcium atoms emerge, and can be studied in the image shown in fig. 4.6d. A cell structure in geology is described by six parameters; the length of three axes, a, b, c and the angles α, β, γ , that defines the orientation of the axes. Place the axis a , along the x-axis. γ is the angle between a and b . While α and β are the angles between a and c , and b and c respectively.

For the portlandite unit cell, we have that $a = b = 3.589 \text{ \AA}$, $c = 4.911 \text{ \AA}$, and the angles $\alpha = \beta = 90 \text{ deg}$, $\gamma = 60 \text{ deg}$.

³cement. (2015). In Encyclopaedia Britannica. Retrieved from <http://global.britannica.com/EBchecked/topic/101833/cement> (08.04.2015)

⁴Cleavage, is directions in crystals that are less tightly bound so that it tends to break more easily along this plane. This is somewhat different from fracturing because it will leave a smooth surface

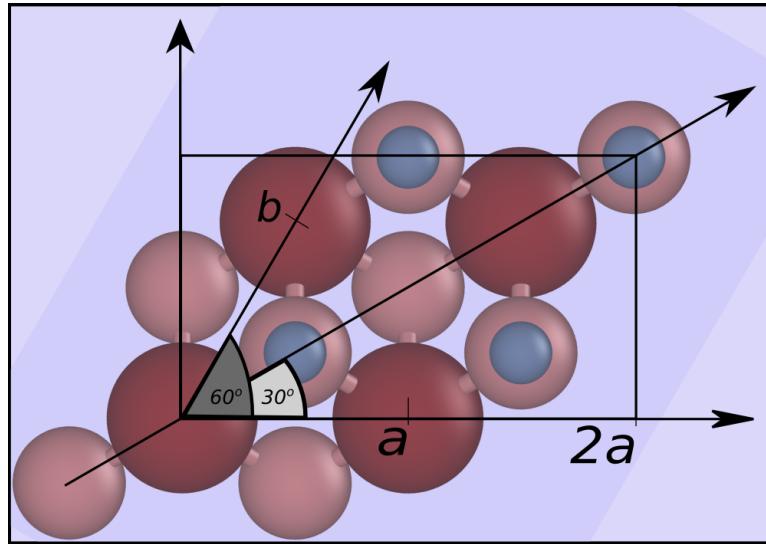


Figure 4.5: Portlandite unit cell. Calcium being the large brown-red-ish particles, oxygen next and the hydrogen is the smallest particles and seen on top of some oxygen atoms. Those oxygen atoms to the left where no hydrogen is seen above, has got hydrogen attached below. These are also located below the Ca surface, while the other OH -groups are located above

In the simulations of portlandite we have used the *ClayFF* force field developed by Cygan et al. The force field parameter are given in section A.1.1. The system were equilibrated using a minimization procedure followed by 10 ps in the NVE ensemble and 20 ps in the NPT ensemble before the radial distributions were calculated in the NVT ensemble. The density were estimated during the NPT run. The results are shown in table 4.4, and the system can visually be studied in fig. 4.6. The binning used in the radial distributions were 0.1 \AA . We notice no large discrepancies in the results. The density is a bit low, but within 98% of the experimental value. In the radial distributions we have not looked at the system in planes, but looked at a larger system with four planes and a total of 6720 particles. The distance between the planes is c , which will give raise to an extra peak for the $Ca-Ca$ distance at approximately 4.9 \AA also located within the confidence interval.

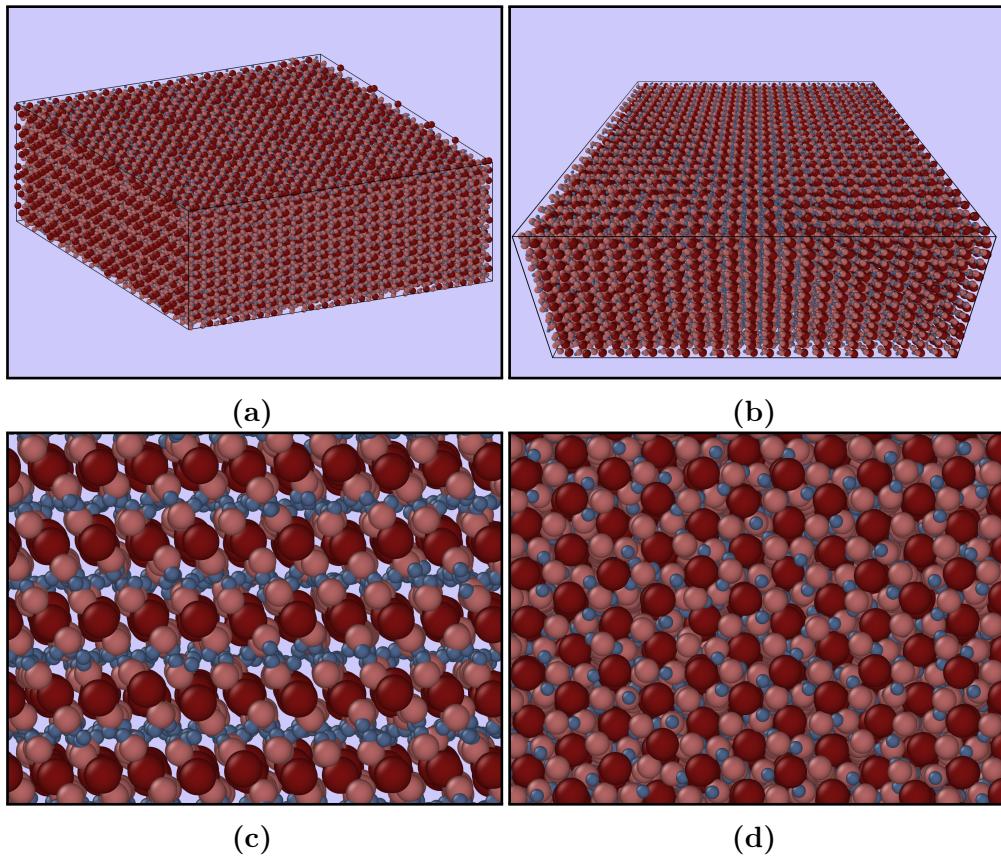


Figure 4.6: Portlandite slab consisting of 60480 particles, seen from different angles figs. 4.6a and 4.6b, and close up from the sides figs. 4.6c and 4.6d

Table 4.4: Tabulated values for the portlandite. All values gathered at 300 K and 1 atm using Nosè-Hover. *fixed

$g(r)$	simulation	Cygan et al.	observed
O-O	3.4 ± 0.2	3.694 ± 0.040	3.589
Ca-O	2.5 ± 0.2	2.515 ± 0.119	2.369
Ca-Ca	3.6 ± 0.2	3.693 ± 0.044	3.589
O-H	1.000*	1.023 ± 0.021	0.942
ρ [g/cm^3]	2.219 ± 0.003	2.239 ± 0.032	2.246

Chapter 5

Quartz surface energy and binding energy of CO_2

5.1 Studied systems of quartz

In this thesis, quartz surfaces have been studied in addition to the shale systems of portlandite and water that will be discussed in the next chapter. In this chapter, the simulations of quartz and the adsorption of carbon dioxide to quartz surfaces is introduced.

Quartz is the crystalline form of silica with an overall chemical formula SiO_2 (silicon dioxide, usually just called silicon oxide), and is the second most abundant mineral in the earth's crust next after feldspar minerals, which by the way, is also mainly made up by silicon and oxygen. Pure bulk quartz is tetrahedral with a continuous structure made up by SiO_4 , forming the tetrahedral geometry. Each oxygen atom is shared by two silicon atoms so that the overall formula becomes SiO_2 . The crystal structure of quartz comes in many different shapes and colors. Impurities arise from presence of elements that the quartz was subject to under the forming of the crystal. For example, iron and silica can crystallize in complex lattice structures, and is known to form the gemstone Amethyst. In theory will quartz at normal pressure and temperature conditions form an overall trigonal structure known as α -quartz, but transforms into a hexagonal form at higher temperatures (850K) to β -quartz or β -cristobalite. Other structures are also known at higher pressure and temperatures. Here we will study pure quartz in its trigonal form, namely the α -quartz. This structure is interesting because of its wide applications in construction, piezoelectric devices and glass industry[32], and it will be interesting to investigate its ability to adsorb carbon dioxide onto

its surface!

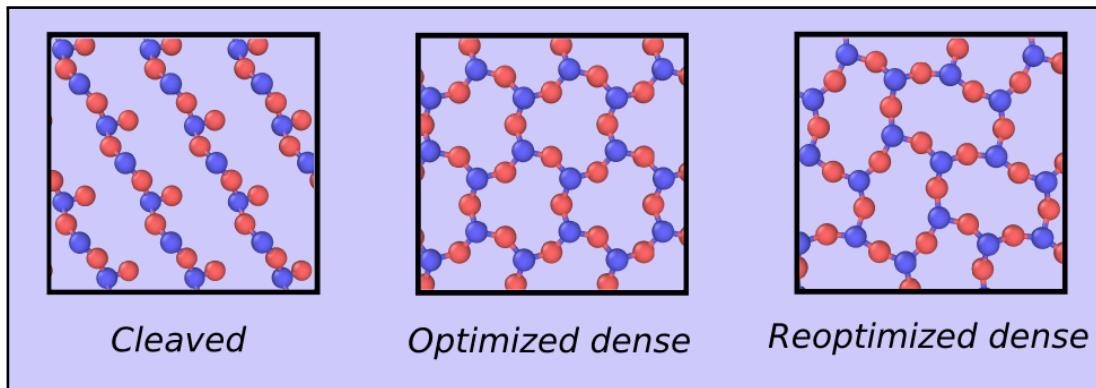


Figure 5.1: The surface structure of the cleaved, dense and reoptimized dense surface of α -quartz. Only the five first Angstrom of the surface is shown, but they are all 27 layer models, and bonds are applied to see the structure more easily. Blue particles is silicon (Si) and red is oxygen (O). The dense surfaces has a hexagonal geometry

The α -quartz is studied with two different surfaces according to the research done in [32] using *Density Functional Theory* (DFT) and *Born- Oppenheimer molecular dynamics* (BOMD). The quartz surfaces are shown in fig. 5.1, but we have not studied in detail the *optimized dense* surface because we obtained the *reoptimized* surface when minimizing its energy. The two *dense* surfaces typically forms hexagonal structure of silicon atoms, where the reoptimized surface has a shifted hexagonal structure with a lower surface energy than the radial symmetric hexagonal dense surface. The cleaved surface were found to have the highest surface energy while the reoptimized had just a little lower surface energy than the optimized dense surface. I was asked by Malyi, the first writer of the article, to perform molecular dynamics calculations that were to be compared with the DFT -results of the surface energy, and the binding energy of carbon dioxide to the α -quartz. The project is partly funded by the Research Council of Norway (project: 221469). The goal was to compare molecular dynamics simulations of the systems with the DFT and BOMD simulations. All the molecular dynamics simulations of the quartz and carbon dioxide in this part of the thesis, were carried out using *ReaxFF* as described in section 2.4.8. A time step of 0.1fsec were used for all simulations according to [8].

The reax force field that was used in the simulations were based upon the values provided by Kulkarni et.al. [28]. Kulkarni et.al. investigate oxygen interaction with silica surfaces using DFT-simulations and Coupled Cluster theory to provide parameters for the ReaxFF potential¹.

¹ReaxFF parameters are not restated in this thesis, but can be fund in the supporting ma-

The systems were given an initial temperature $100K$ from a Gaussian distribution, and then cooled down to $< 0.03K$ in the NVE -ensemble using a Berendsen thermostat. A minimization procedure, utilizing the conjugate gradient method, were used before holding the temperature at approximate absolute zero for $5ps$ where the potential energy of the system was sampled to estimate the ground state energy.

5.1.1 Surface energy

The estimates on the surface energy of the systems were carried out for the cleaved and the reoptimized dense surface. Using the conjugate gradient method with a convergence criteria/tolerance of 10^{-12} for the energy (unitless), and $10^{-12}kcal\ mol^{-1}A^{-1}$ for the forces. The maximum number of iterations and force/energy evaluations were 1000. The system were initialized at $100K$, and cooled down to below $0.03K$ over $30ps$ before another minimization were conducted. The simulations were performed with both a non-periodic boundary condition in the z-direction, and with a vacuum above the surfaces that had an extent three times larger than the cut-off distance of the atomic interactions. The obtained system ground state at absolute zero were compared with two bulk quartz systems, one of which was an SiO_2 unit cell, and the other, a bulk system that had just as many particles as the surface systems (cleaved and dense surface). The estimate on the surface energy were carried out using an average over $5ps$ at the given temperature. The estimate on the surface energy were also conducted at other temperatures. After cooling a system to absolute zero, it was heated to 300,500,1000 and 1500K.

5.1.2 Sequestration of carbon dioxide

Carbon dioxide can bind to the surface in three different ways. With one dangling end, forming a single bond with a surface Si atom, see fig. 5.2a. A tighter connection where also an oxygen attaches to the carbon, see fig. 5.2b. And finally the case where the latter two bondings takes place in addition to a third bond with the second oxygen atom in the carbon dioxide molecule, that forms a bond with another silicon atom. The latter one were not discovered.

Only the cleaved surface is considered for the adsorption of carbon dioxide. But four different systems is tested. The CO_3_1 and $SiO_4_C_O$ has 111 particles and only one carbon dioxide molecule, but the attachment for the $SiO_4_C_O$

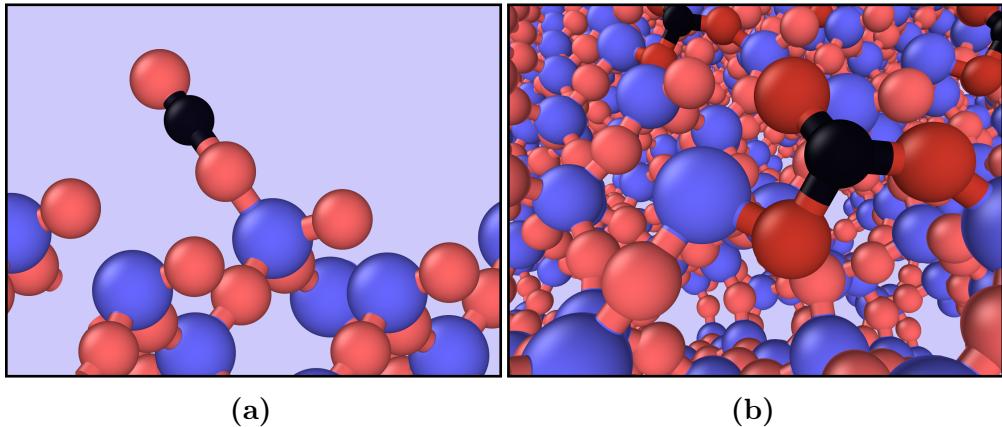


Figure 5.2: Carbon dioxide molecule attached to the cleaved quartz surface. Left, single bond. Right fig. 5.2b, a more tightly bound CO_2 molecule, where the carbon connected oxygen atoms are colored a little deeper red

system is a one-bond like shown in fig. 5.2a, while for the CO_3_1 we have a bonding like in fig. 5.2b. The $CO_3_1_f$ and $CO_3_2_f$ systems have 132 particles in total and eight carbon dioxide molecules, four at each side of the system. Minimization procedure is carried out as explained earlier, and the systems are initiated at $100K$ from a Gaussian distribution and cooled down to below $0.05K$ over $30ps$, where the system is kept at $5ps$ to perform the calculations.

5.1.3 Density functional theory

I would like to include a small digression, where the concept of density functional theory is briefly introduced. Density functional theory is often considered to be *ab initio* calculations, meaning that it comes from first principles. The motion of atoms is governed by the laws of quantum, so their movements are determined from quantum mechanics. But solving the Schrödinger equation is difficult for systems of many particles because the Hamiltonian of the system becomes prohibitively large. We have to do the Born-Oppenheimer approximation, fixing the nuclei, to solve the equation. This is most commonly done using variational calculus and approximating the one-particle wave functions using a basis set of simpler functions to increase the computational efficiency. This method is named the Hartree-Fock method, and can be read about in Thijssen's computational physics [58]. The Hartree-Fock method is difficult to use for large systems, and for this reason it is mainly used to look at single atoms and molecules. Density functional theory is rooted in quantum mechanics as well. It is based on the results of Hohenberg, Kohn and Sham [26, 20]. In a system of N electrons, rather than focusing on the electron orbitals (the one-particle wave functions,

which has $3N$ spatial plus N spin variables), DFT focus on the electron density $n(\mathbf{r})$, and the Hamiltonian is assumed to depend on the local electron density only. Density functional theory is very successful in its way of describing static systems of solids or single molecules. Minimization procedures are used to find the ground state of the system, and therefore accurate in estimations of energy evaluations that can be used for example to calculate the surface energy of a solid.

5.2 Estimates on binding energies and surface energies of quartz systems

In cooperation with Malyi, Oleksandr and Boțan, Alexandru, quartz structures were studied using molecular dynamics and molecular statics to obtain ground state energies of the systems provided from Malyi [32]. The systems were studied to make estimates on the surface energy of two kind of quartz surfaces, and the binding energy of carbon dioxide to these surfaces.

An interesting observation is that the cleaved surface turned into the beautiful structure shown in fig. 5.3a. While the dense27 (optimized dense) turned into a shifted hexagonal structure show in fig. 5.3c. When we look into the quartz from the side (figs. 5.3b and 5.3d), showing the bulk material, we see that the bulk structure is preserved in both systems.

Tabulated results are shown for the reoptimized dense (upper four results) and the cleaved system *c*, in table 5.1. Two bulk systems were used to find E^{bulk} , both with periodic boundary conditions, but one simulation using only a single unit-cell, and the other 12 times larger, which correspond to the same number of particles as in the cleaved and the dense27 systems. Standard deviations were calculated but were very small. The reason for this is not a physical one, all dynamics were killed because the velocities of the particles were rescaled more frequently than the sampling frequency. This is unfortunate, and should be avoided. Tabulated results at higher temperatures were calculated in table 5.2, these estimates shows that a reasonable error is $6.0 \text{ E-}4 \text{ J/m}^2$. In comparison with earlier studies of quartz, the surface energy has been claimed to reign from $0.4 - 1.0 \text{ J m}^{-2}$ [6], but the DFT-analysis gets a surface energy of 2.23 J m^{-2} for the cleaved surface, and $0.39 - 0.35 \text{ J m}^{-2}$ for the optimized surface. In comparison with other types of surfaces, we have for example $\text{NaCl} = 0.5 \text{ J m}^{-2}$, and diamond $C = 5.5 \text{ J m}^{-2}$. A glass surface is given as something in the range of $2.0 - 4.0 \text{ J m}^{-2}$. In general, metals tend to have a surface energy close to 1.0 J m^{-2} , while covalent

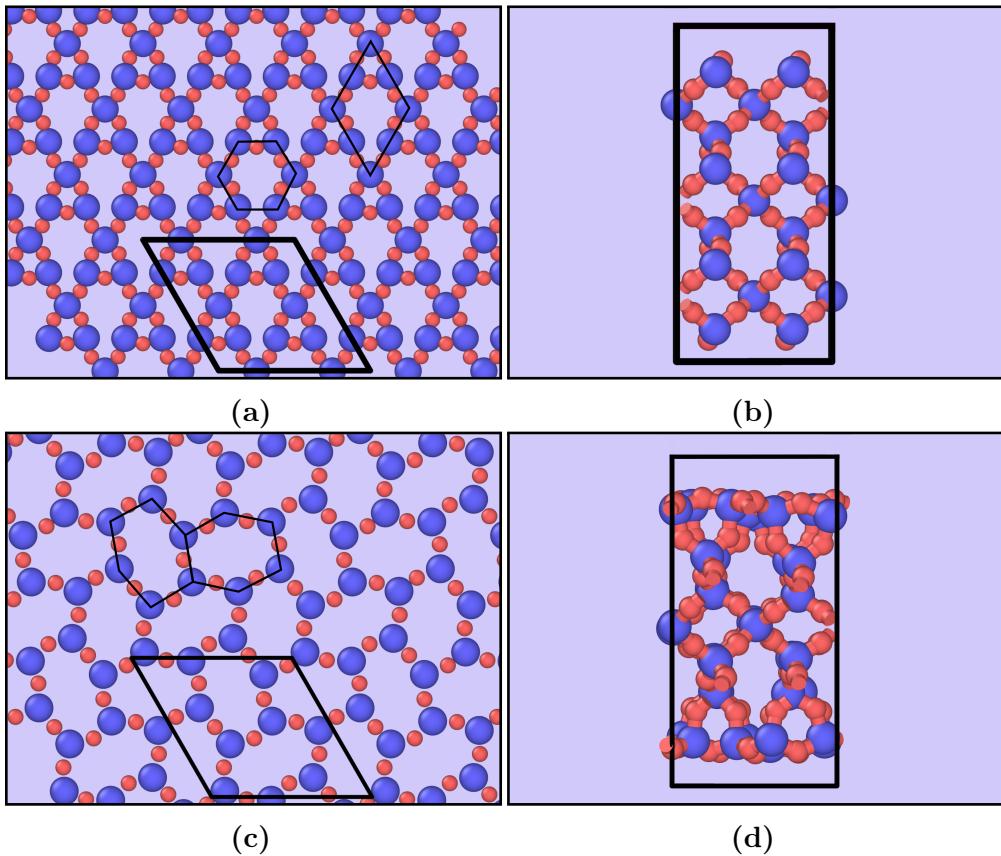


Figure 5.3: Cleaved and reoptimized system, both after energy-minimization procedures. The parallelogram shows the boundaries of the triclinic unit-cell system. The upper left image fig. 5.3a, shows the cleaved surface after minimization. In fig. 5.3b, the system is shown along the x-axis. Equivalent for the reoptimized dense27-layer system in the lower two figures. Blue particles is silicon (Si) and red particles oxygen (O)

solids tend to have higher, and ionic solids lower surface energy².

The results for the simulations on the surface energy is given in table 5.1, and a plot of a simulation on the surface energy of the slab using both periodic and non-periodic boundary conditions in the direction normal to the surface is shown in fig. 5.4. We see that the results are pretty much the same using the bulk one-cell, and the 12 times larger bulk system, but that we get a higher surface energy when using non-periodic boundary conditions in the direction of the surface normal. The difference between the periodic and non-periodic simulation at the end of the simulation is of the order 4.0 E-3. A higher surface energy for the non-periodic

²Tabulated values found at *insula*: <http://www.insula.com.au/physics/1279/L8.html> (20.05.2015)

Table 5.1: Surface energy of the slab (dense27) system using periodic p , and non-periodic np boundary condition in the direction of the surface normal. And the surface energy of the cleaved surface c . The letter o is a calculation where only a unit-cell of the bulk system were used for the calculation, and b is a calculation where there were just as many bulk particles as in the reference surface

system	N_{atoms}	$E_s [J/m^2]$
o,p	108	0.4339
b,p	108	0.4338
o,np	108	0.4383
b,np	108	0.4381
o,c	108	1.6802
b,c	108	1.6800

system is due to a larger potential energy at the ground state, which is associated with larger forces in the system because of a porer distribution in the surface structure. We see from the figure that the approach to the ground state energy is very different when using the one-cell and the twelve times larger system, but that the simulations using periodic and non-periodic boundary conditions is approximately the same. I will assume from other similar simulations, that this is due to a larger disorder for the one-cell system, and that the reason for this is a cut-off length that is larger than the system size. The larger system is therefore more accurate in reproducing natural behavior.

As mentioned earlier, in table 5.2, tabulated results for the surface energy for the dense27 system is shown at different temperatures. We see that the surface energy does not change much, but slowly gets larger, which is in good agreement with the DFT results where the surface energy stays approximately the same up to 1500 K . When the system reach about 1900 K , the material melts, and the potential energy of the surface and the bulk material becomes approximately the same, so that the surface energy rapidly drops towards zero.

5.2.1 Results on the binding energy of CO_2

The binding energy of carbon dioxide were calculated according to four cleaved surface systems, $CO3^1$, $CO3^{1f}$, $CO3^{2f}$ and $SiO4^{CO}$. The systems have got 1, 8, 8 and 1 CO_2 molecules bound to their surfaces, respectively. $CO3^1$ and $CO3^{1f}$ have each got CO_2 loosely bound to the surface, while $CO3^{2f}$ and $SiO4^{CO}$ have CO_2 bound to a Si atom and an oxygen bridge as illustrated in the method part, section 5.1. A pure cleaved system without any carbon dioxide is used as the

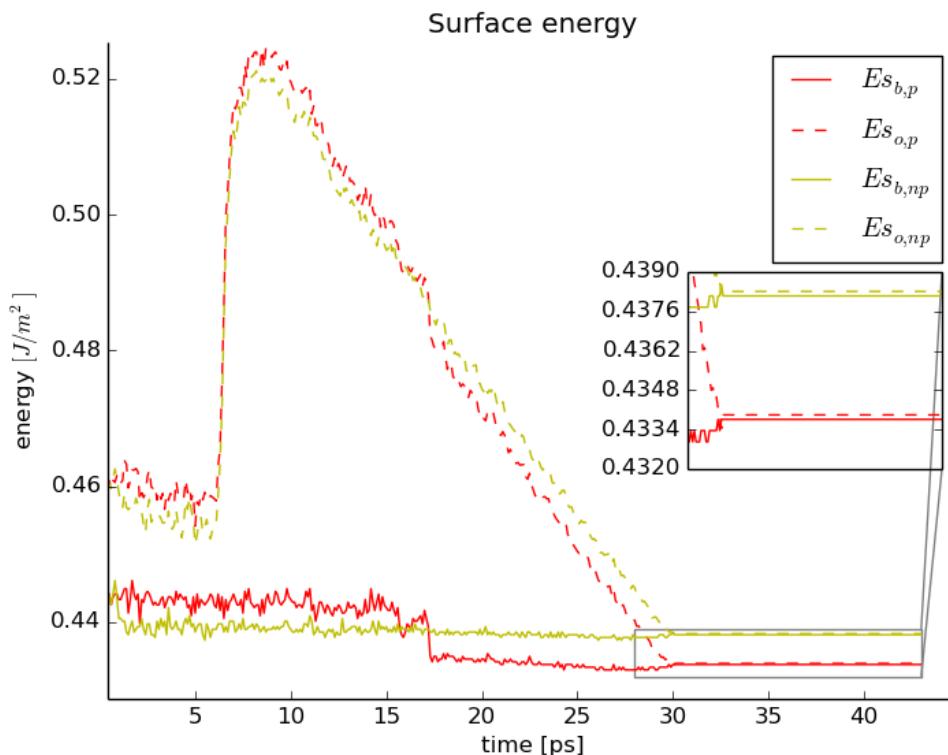


Figure 5.4: When estimating the surface energy of the dense27 slab, a periodic(p) and a non-periodic(np) system were tested, along with two different sized bulk reference systems, the one-cell and the bulk(o,b)

reference surface (adsorbent), according to eq. (2.30). The ground state energy of a single CO_2 molecule were calculated to -16.65 eV .

The tabulated results from the ReaxFF simulations is shown in table 5.3. The results obtained with the Kulkarni ReaxFF is compared with results obtained by Boțan using another set of parameters for the ReaxFF (PDMSDecomp.ff from Chenoweth et al.)³.

Comparing the molecular dynamics simulations with the DFT -results, we see quite some discrepancies. First of all, I have to say that the average total force per particle in the DFT -simulation were reported to be in the order of $0.01\text{ eV}/\text{\AA}$, while for the † -simulation the forces were of the order $0.02\text{ eV}/\text{\AA}$. The results obtained with the Kulkarni ReaxFF were averagely $0.03\text{ eV}/\text{\AA}$. It proved to be hard to get the forces any lower in the molecular dynamics simulations, which suggests that a local minima were obtained as far as the force field allowed.

All binding energies are negative, which is correct. And we see a tendency that

³The ReaxFF parameters can be found through [9]

Table 5.2: Surface energy of the reoptimized system (dense27). The estimates were performed at given temperature after, (as an example) in the first row 30 ps cooling from 300 K to 0 K and then the estimate were made over 5 ps . The results in the second row were obtained after this state, warmed up from 0 to 300 K over 30 ps , and then holding the temperature at 300 K for 5 ps . And so on for the following rows

T [K]	t [ps]	o,p [J/m^2]	o,np [J/m^2]
0	$30 + 5$	0.4339	0.4383
300	$30 + 5$	0.475 ± 0.006	0.466 ± 0.003
500	$20 + 5$	0.472 ± 0.002	0.472 ± 0.002
1000	$50 + 5$	0.484 ± 0.003	0.482 ± 0.002
1500	$50 + 5$	0.510 ± 0.003	0.499 ± 0.006

Table 5.3: Binding energy of CO_2 to the given surfaces. The force, F is the force per particle in the energy minimized system. The binding energies of carbon dioxide to the surface is given as per CO_2 molecule. \dagger -results from PDMSDecomp.ff ReaxFF simulations. $*$ -results from DFT simulations.

system	N_{CO_2}	N_{atoms}	$F[eV/\text{\AA}]$	$E_b [eV]$	$E_b [eV] \dagger$	$E_b [eV] *$
CO_3^1	1	111	0.04488	-1.525	-3.97	-1.35
CO_3^{1f}	8	132	0.02621	-0.722	-2.09	-1.28
CO_3^{2f}	8	132	0.03008	-2.972	-2.57	-1.85
SiO_4^{CO}	1	111	0.04954	-1.805	-3.43	-0.66

the binding energy of a carbon dioxide molecule becomes weaker when the surface is saturated with many molecules. As many molecules attaches to a surface, less bonds are available, and the system is closer to a stable state.

According to a study on adsorption of CO_2 performed in 2000, the binding energies of carbon dioxide with metal surfaces proved to be strongly dependent on the geometry of the adsorbed CO_2 onto Pt(111) and Fe(111) surfaces[10], with binding energies varying from -0.05 eV to -1.35 eV for the platinum surface, and -0.66 eV to -3.73 eV for the iron surface using a semi-empirical molecular orbital method called ASEMO[10]. From our results, it seems to be a too large discrepancy to say anything about such a correlation, except for the results using Chenoweth's ReaxFF(\dagger), which also gives the largest binding energies for the carbon dioxide.

Chapter 6

Diffusion in portlandite nanopore

6.1 Studied systems in portlandite nanopore

In order to get further insight on the behavior of water and carbon dioxide in nanoporous shale systems, a mineral commonly known as *portlandite* in concrete industry, is studied. Portlandite were chosen because of its relatively simple structure compared to other shale components like *kaolinite* and *pyrophyllite*. Also good information about crystallographics and benchmarks were given in [12]. The portlandite is a layered structure controlled in part by hydrogen bonds, and the structure is stable under the conditions that we have been looking at.

The systems are all studied using *ClayFF* developed by Cygan et al.[12, 14, 46], while the SPCE water model is applied for all water, and the FPF model is used for the carbon dioxide. All force field parameters is found in section A.1. The systems are equilibrated to 300 K and 1 atm pressure for all systems using Nosè-Hoover for 50 ps . The Nosè-Hoover thermostat is also applied to control the temperature to 300 K in the NVT ensemble afterwards. In this section we are going to look at the results from simulations of water and carbon dioxide confined within a portlandite nanopore. So first of all, we would like to know how water and carbon dioxide behave in a bulk system consisting only of these two materials.

6.1.1 Mixing water and carbon dioxide

It is interesting to know how the transport properties of water change with the amount of carbon dioxide in the water. And since the usage of CO_2 and water

Table 6.1: Systems of carbon dioxide and water. Weight percent of CO_2 defines the bulk system. Diffusion coefficient ($[10^{-8} m^2/s]$) for water and CO_2 in the system. The number of carbon dioxide molecules and total number of particles are given along with the density of the system and partial density of water (densities given in $[g/cm^3]$).

CO_2	D_{H_2O}	D_{CO_2}	N_{CO_2}	N_{tot}	ρ_s	ρ_{H_2O}	$t_{sim}[ps]$
4.5%	0.28 ± 0.015	0.34 ± 0.02	30	4482	1.021	0.972	200.4
2.3%	0.308 ± 0.005	0.33 ± 0.02	15	4437	1.003	0.978	200.2
1.4%	0.305 ± 0.007	0.23 ± 0.02	9	4419	1.005	0.990	200.2

to enhance oil recovery from oil wells, is considered, accurate knowledge of thermodynamic and transport properties of the mixtures is necessary[37]. Is there a clear relation between the amount of CO_2 mixed into the water and the change in the diffusion coefficient? To get an idea about this question, three simulations of a bulk system with water and carbon dioxide were conducted with 1.4%, 2.3% and 4.5% carbon dioxide (weight percent, m_{CO_2}/m_{system}) where the density and diffusion properties of the system are studied.

The systems were equilibrated in the NVE ensemble and following in the NPT ensemble for so long that carbon dioxide were well mixed into the water before the NVT run. There were no observations of bubbles forming in the water. The results of the simulation is given in table 6.1. The density of the system remains approximately the same as for water, with just a slight increase in the density, but this can be because we are simulating quite small systems. In small systems it is difficult to equilibrate water at the correct pressure because of its incompressibility. We see an increase in the diffusion of carbon dioxide within the water with increased amount of CO_2 . Remember that we obtained very high diffusion coefficients for bulk carbon dioxide in its gas form ($D \sim 125 E-8 m^2/s$ for the FPF model), but in these systems, carbon dioxide shows no interest in becoming a gas. In a later simulation with carbon dioxide and water within a portlandite nanopore, we also see that some CO_2 molecules diffuse into the water, which suggests that these results are plausible. When looking at the self diffusion of water within the bulk system, the results are a bit disturbing. The diffusion coefficient for water in the 4.5% system is almost the same as for bulk water, while the two others suggests a higher diffusion coefficient for water with CO_2 mixed into it. In a similar study by Moults et al. five CO_2 molecules were used in a system consisting of 2000 H_2O molecules, resulting in a weight percent of 0.6% carbon dioxide in the water. The diffusion coefficient of the carbon dioxide at 1 atm and 298.15 K were 0.27 for SPCE-EMP2 model and 0.22 for SPCE-TraPPE. At 1 atm and 323.15 K , the values were 0.38 and 0.39 for the two combinations respectively. All values in $[10^{-8} m^2/s]$. The self diffusion coefficient of water in the mixed system were estimated to 0.26 and 0.43 for 298.15 K and 323.15 K .

Table 6.2: System information. The number of water and carbon dioxide molecules are given as well as the total number of particles in the system. Total simulated time, and total CPU time usage.

system	N_{H_2O}	N_{CO_2}	N_{tot}	$t_{sim} [ps]$	$t_{cpu} [h]$
H_2O	5824	0	29792	302.9	23.3
CO_2	0	60	36180	700.5	31.5
H_2O and CO_2	2760	88	20864	547.9	27.2

respectively.

We see that the results are well comparable with the 1.4% amount of carbon dioxide for our simulation using SPCE-FPF.

6.1.2 Water and carbon dioxide within portlandite nanopore

Three systems of water and carbon dioxide within a portlandite nanopore is studied. Images of the systems can be seen in fig. 6.1. The first system have only got water within the pore. The second have only carbon dioxide within the pore, and in the last system, we study water with a carbon dioxide bubble located at the left surface of the portlandite nanopore. The reason for placing the bubble at this location is obviously to have a symmetric system, so that we are able to look at the transition between the water and the carbon dioxide, and at the same time look at the difference at the portlandite surfaces for carbon dioxide and water.

In table 6.2, some system information is given. It is also worth mentioning that the thickness of the portlandite is approximately 18 Å, and that the cut-off for the Lennard-Jones and Coulomb interaction is 12 Å so that the short-range interactions do not cross the pore wall of total thickness 36 Å. The poresize is approximately 4.6 nm in all three systems, and the total surface area of the portlandite is 71.13 nm (two surfaces).

The density profiles of a pore system is gathered by dividing the system into equally spaced bins with size 1.0 Å, in the desired direction, and count the number of every particle type belonging to a bin. Then the densities are averaged over typically 50 time steps.

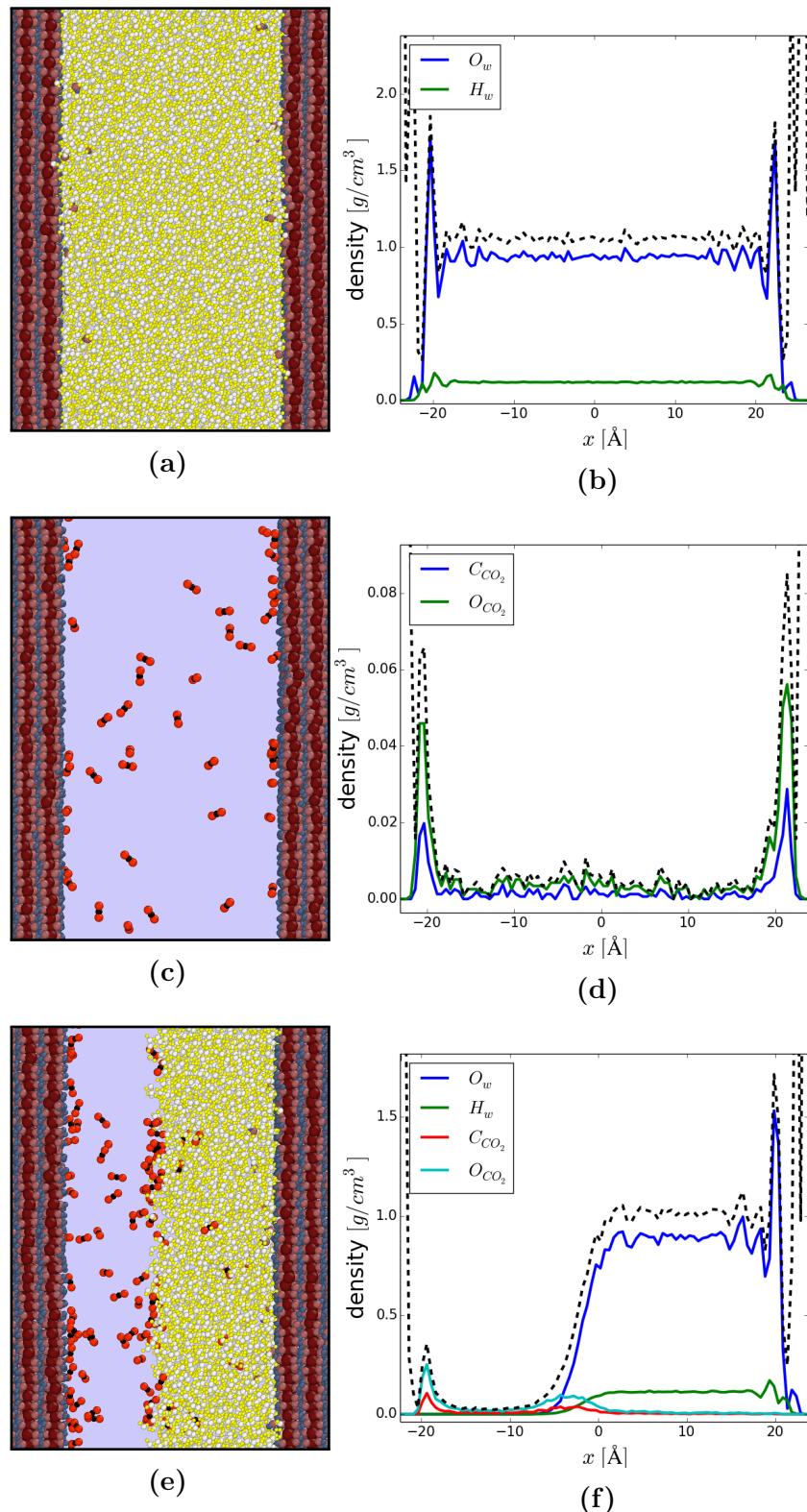


Figure 6.1: Simulated systems of water and carbon dioxide in portlandite nanopore and their density distribution. Total density is illustrated by the dotted line. figs. 6.1a and 6.1b shows water in the nanopore. figs. 6.1c and 6.1d show carbon dioxide, and figs. 6.1e and 6.1f shows the carbon dioxide bubble on the left surface with water occupying the rest of the volume in the nanopore

From the density plots across the nanopores in fig. 6.1, we see that both water and carbon dioxide is “wetting” the portlandite surface. Meaning that they are attracted to the surface. We also see in the simulation that water seems to pull out some of the OH-bonds from the portlandite, and that this seems to be enhanced when carbon dioxide is present in the water. This could be a problem for longer simulations because there is an angle and bond term between the calcium and oxygen in the portlandite, and no breaking of bonds or angles are allowed in this model, but this can easily be avoided by freezing the portlandite matrix. Another notice is that, at the carbon dioxide - water interface, we have a small accumulation of carbon dioxide, and that we have a smooth transition to the bulk water within the pore. And that some carbon dioxide molecules manage to diffuse all the way through 2 nm of water to the other portlandite surface during the 547.2 ps long run.

The density of the water within the pore rapidly becomes the same as for bulk water. The density for carbon dioxide within the pore also rapidly becomes constant, just about five Ångström from the surface, although both the density for the water and the carbon dioxide is higher within the pore than in the bulk system. The density for the carbon dioxide within the nanopore arrives centrally at an approximate value of $4.0 \text{ E-}3 \text{ g/cm}^3$, which corresponds to an experimental density of CO_2 at 300 K and 2 atm pressure¹. We find a fifteen times higher density closer to the surface, while the density of water increase by approximately 1.5 - 2 times by the surface.

¹Can be found from datas at NIST: <http://webbook.nist.gov/chemistry/fluid/>

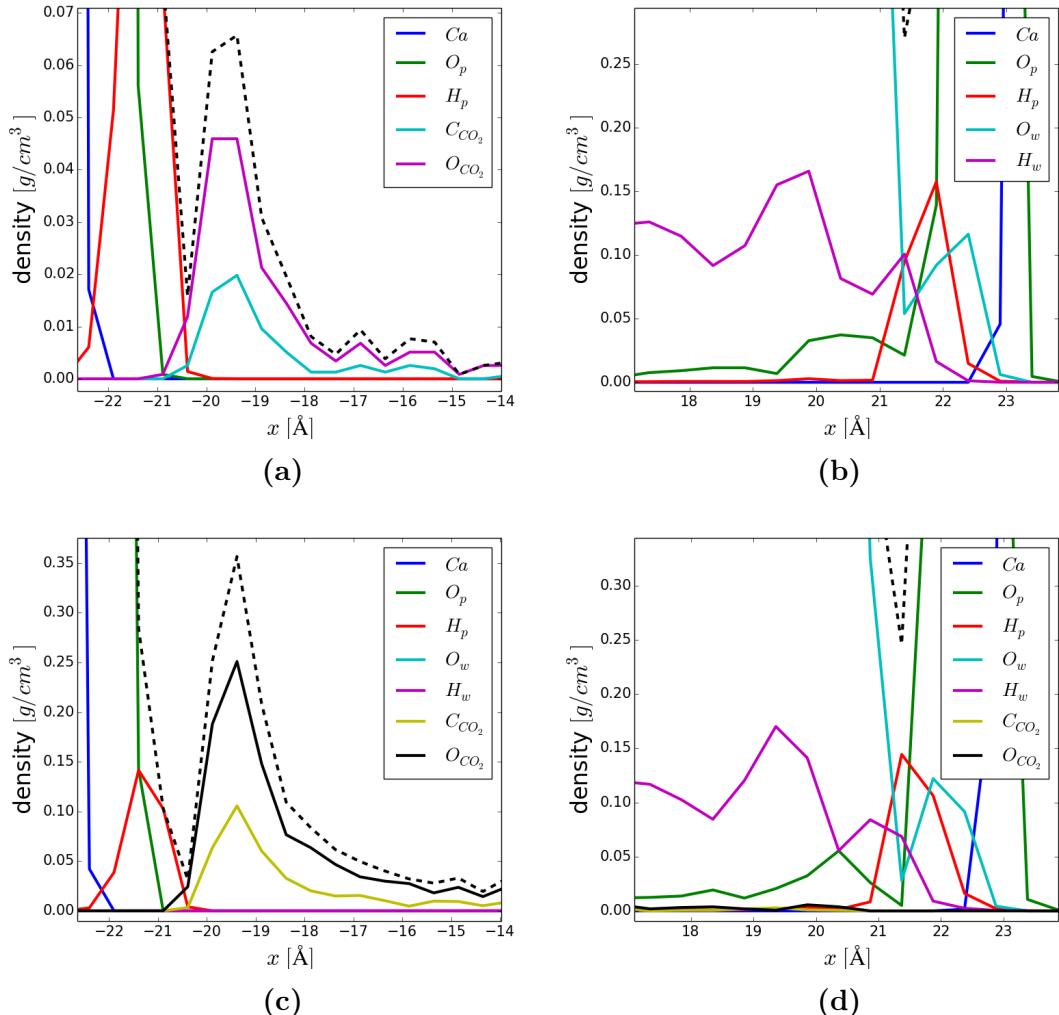


Figure 6.2: Density profile close to the portlandite surface. The x-axis shows the distance from the center of the system. figs. 6.2a and 6.2b is for the pure H_2O and CO_2 systems respectively, while figs. 6.2c and 6.2d is for the bubble system with carbon dioxide - portlandite interface in the left figure, and water - portlandite interface in the right. Total density is illustrated by the dotted line

When we look at the density profiles close to the surface of the portlandite (see fig. 6.2), we notice that there actually is an overlapping of the oxygen (O_w) and the portlandite hydrogen (O_p). We also see from the density plots that we have got a little peak of water oxygen (O_w) closer to the surface than the first peak for the water hydrogen (H_w). The large peak for the oxygen is closer to the surface than the second peak for hydrogen, suggesting that it is the negatively charged part of the water molecule that is attracted to the surface. We do not

see such a relation for the the carbon dioxide. The surface of the portlandite can be predicted to be somewhere in the range between the portlandite oxygen and hydrogen (H_p), which gives it an uncertainty of 1.5\AA .

When looking at the transport properties for the water, the system were divided across the pore with equally spaced bins, each with size 1.0\AA . The diffusion of water molecules belonging to that bin is then calculated from the mean square displacement over 10 ps . Ten time-origins were used to average the mean square displacements. The resulting diffusion of water as a function of distance to the portlandite surface is shown in fig. 6.3. We see that for both systems (#1 and #2), the diffusion coefficient approaches the coefficient obtained for bulk SPCE water ($0.27 \times 10^{-8} \text{ m}^2/\text{s}$), when in the center of the nanopore. Notice how the diffusion coefficient for water in the CO_2 bubble system is very low close to the surface of the portlandite. We also see in fig. 6.3a, that sometimes the estimates are very good, and other times they are bad, resulting in large standard deviation. We look into the mean square displacements, and sometimes we observe dramatic drops in the MSD's, suggesting that we have lost many particles out of a bin representing a distance from the surface. This can be adjusted so that we get a better estimate, either by increasing the bin size, or by increasing the sampling rate for the system. We increased the sampling rate to measure the MSD over 5 ps , and thereby obtained the plot shown in fig. 6.3b, giving overall, better estimates on the diffusion coefficient as a function of distance from the surface.

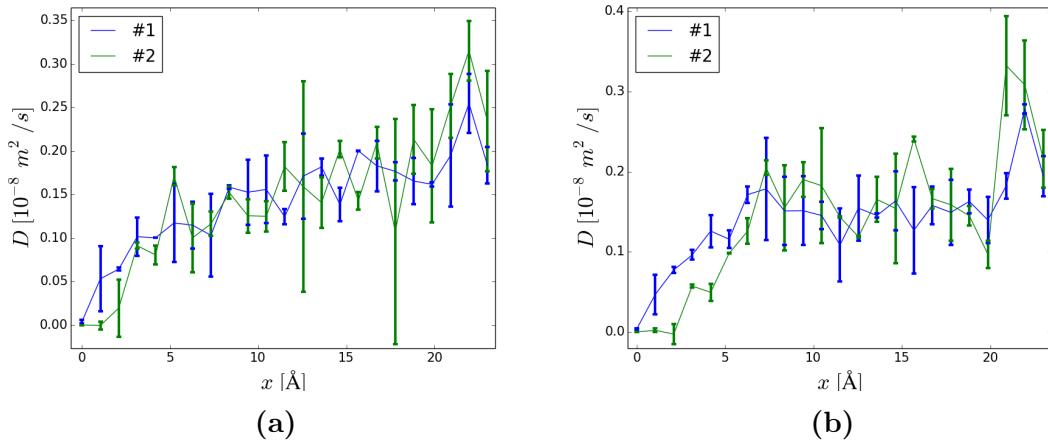


Figure 6.3: Diffusion of water in portlandite nanopore as function of distance to the surface. The error bars indicate the standard deviation. #1 water in nanopore, #2 CO_2 bubble in water inside nanopore. fig. 6.3a shows the results from fewer estimates over longer timescales than fig. 6.3b

In the new estimate in fig. 6.3b, we observe that the diffusion appears to have

Table 6.3: diffusion of carbon dioxide in nanopore. Values given in $10^{-8}m^2/s$

<i>location\system</i>	only CO_2	CO_2 bubble
at surface	9.49 ± 0.09	3.2 ± 0.8
in center	24 ± 3	2.0 ± 0.2

reached the bulk value after approximately 7 \AA , because it becomes fairly constant from hereon and out until the middle, in contrast to the result in fig. 6.3a, where the diffusion coefficient is increasing all the way towards the center of the pore. In the middle, we observe that the diffusion coefficient increases for the three centermost estimates, this could be due to

When looking at the diffusion of carbon dioxide in the systems, we had to change the bin size dramatically, to be able to look at time scales long enough to get estimates on the diffusion coefficient. The results are given in table 6.3. The bin size were 11.625 \AA , giving two bins inside the nanopore. One close to the surface and the other next to this one and out to the center of the pore system. We also had to calculate the MSD over $50ps$, giving four and six estimates on every diffusion coefficient for the leftmost and the rightmost figure respectively. From the table, we see that the diffusion coefficient in the CO_2 -bubble system is almost the same as the one obtained for the bulk CO_2 and water system studied in section 6.1.1, and that is larger at the surface than in the center of the pore. In the nanopore with just carbon dioxide, we obtain in general a higher diffusion coefficient. Although the coefficient does not get close to the one obtained for bulk carbon dioxide ($125 \times 10^{-8} m^2/s$ for the FPF model). Here we get a more intuitive result, with a larger diffusion coefficient in the center of the pore, that closer to the surface.

Chapter 7

Summary, conclusions and prospects

In this thesis, we aimed to develop a clay unit-cell structure, and chose to do so with the material portlandite. With the chosen material, we built a clay nanopore model in which we studied the transport properties of water and carbon dioxide. We also wanted to study the surface energy of quartz, and the binding energy of carbon dioxide to such surfaces, since quartz is a common material in shales and sandstones. We wanted to evaluate the constituent parts by comparing our results with experimental and other computational results.

The results in this thesis, is divided into three parts. A benchmarking part, in which the constituent parts of the nanopore model is tested. A second part, where the *ReaxFF* simulations of quartz and carbon dioxide is performed and compared with DFT-simulations, and the last part where results from the nanopore model is presented.

7.1 Discussion of the benchmarking results

We set out to compare simulations of bulk water, bulk carbon dioxide and bulk portlandite. Radial distributions for the bulk water showed a nearly spot on resemblance with Mark et al.[33], and the diffusion coefficient were estimated to a higher precision than in the comparison studies. A lower density should lead to a higher viscosity and make it more easy for a water molecule to travel in the bulk water, so our estimates should have overestimated the diffusion coefficient, but in comparison with Mark et al., the results are close to other studies on the

SPCE water model.

We observed that the pressure in the system fluctuated a lot for the bulk water. In fact, it had a standard deviation of 400 atm , which is a lot. But we justified it then by saying that water is incompressible, and small changes in the volume would lead to large changes in the pressure. Mark et al. have the same standard deviations in the pressure, but this can be something worth looking into. What standard deviation in the pressure can we expect from waters incompressibility on this length scale?

We also looked briefly into the tetrahedral order parameter for bulk water, but since I find the measure very vague, and somewhat difficult to grasp what it really tells me, I have not paid too much attention to it. We did see that the order parameter gave a result comparable to that of Sund[55].

For the bulk carbon dioxide, we tested three force fields. The rigid models overestimated the density with 6 and 9%, and the flexible model underestimated the density by 10%. Comparing with the water model which is only 1% off, this result is not so good. The result will affect the estimate on the diffusion coefficient, making the result on the diffusion less accurate. I would like a force field that reproduced the density of the carbon dioxide much better than the force fields presented here, as Cygan et al. pointed out in their attempt to make a flexible force field for the carbon dioxide, the energy parameter is not sufficiently fitted. This will lead to a wrong kinetic energy in the bulk carbon dioxide as a result of the temperature in the system, and thereby give a wrong density. They do not mention any discrepancy in the density of CO_2 in their studies. The solution to this problem could therefore be to look into the fitting of the energy parameter in the FPF force field.

We have used portlandite with the clay force field described by Cygan et al.[12, 14], and found that we reproduce the same distances between atoms in the system, and the density of the portlandite is 1.2% off the experimental value, which we reckon to be sufficiently good to say that the modeled portlandite will behave like the one in Cygan's article. We have seen that the portlandite is stable in time for temperature, pressure and energy fluctuations.

7.2 Discussion of the quartz results

We have studied two quartz surfaces, the *cleaved* and the *reoptimized dense/dense27* surface, and compared the results with DFT-simulations. We observed that the cleaved surface re-organized into a hexagonal surface structure, and that the *optimized dense* surface re-organized into a shifted hexagonal structure that had a lower potential energy than the plain hexagonal structure of the cleaved

surface. The cleaved surface were used to measure the binding energy of carbon dioxide to quartz. The molecular dynamics simulations showed that the binding energy of the carbon dioxide should lie in the range -0.7 eV to -2.9 eV per carbon dioxide molecule, which is within the same range as for the results from the DFT-analysis, but we did not see any clear relation between the number of carbon dioxide molecules attached to the surface and the binding energy as for the Chenoweth et al.[8] ReaxFF, where also the binding energies obtained, were larger, and in the range -2.0 eV to -4.0 eV . It is obvious that the choice of force field is important for the result. The precision should be better, and we should expect to see a relation between the number of carbon dioxide molecules attached to the surface, and the average binding energy of a molecule. I would therefore say, that none of the results suffice, but that they can give an indication of the binding energy for carbon dioxide to a quartz surface.

We also noticed that the carbon dioxide were released from the surface when the system were heated up to about 300 K at 1 atm .

7.3 Discussion of the nanopore system

In the final part of the performed simulations, we studied a portlandite nanopore system which we filled with water and carbon dioxide. As a first experiment, we tested a mix of carbon dioxide and water for 1.4%, 2.3% and 4.5% carbon dioxide. The systems all became well mixed, and the simulations showed that the diffusion coefficient for both H_2O and CO_2 were close to the results obtained in a similar study, but with a 0.6% carbon dioxide content. This suggest that our simulations of carbon dioxide and water for the FPF-SPCE model is reliable, and close to reality. A slightly higher density were obtained for the mixed system than for bulk water, these results are not backed up with external information, but since no bubbles are created, it seems probable that the density will remain approximately the same.

We moved further, and injected water into a 4.6 nm portlandite pore. The density profile showed that the portlandite were wetting, because water accumulated at the surface, and that water oxygen atoms were found to move in between OH -groups in the portlandite surface. Some OH -groups were observed to be dissolved into the water. When carbon dioxide were injected to a similar portlandite nanopore, we saw that CO_2 were also attracted to the portlandite, but were left on top of the surface.

A carbon dioxide bubble were created close to the left surface in the portlandite nanopore, with water filling the rest of the pore. A stable state were reached after

less than 50 ps , where the combined system behaved as expected from what we experienced from the nanopore system with only water and the nanopore system with only carbon dioxide within the pore. In the transition between the water and the carbon dioxide bubble (water - carbon dioxide interface), a higher concentration of carbon dioxide appears ($\sim 0.125\text{ g/cm}^3$) than in the center of the bubble ($\sim 0.025\text{ g/cm}^3$), but not as high concentration as close to the portlandite surface ($\sim 0.35\text{ g/cm}^3$). This shows that a surface of CO_2 is created, that prevents water from penetrating into the bubble. Some carbon dioxide molecules is observed to diffuse into the water as we would expect.

When it comes to the diffusion of water inside the nanopore as a function of distance from the portlandite surface, we saw that the transport properties were highly reduced close to the surface, but that the diffusion coefficient obtained for bulk water were approached as we move closer to the center of the pore, just as we would expect. However, I believe that better estimates easily can be made with a longer simulation where more origins for the mean square displacement is used. The discrepancy in the diffusion coefficient in the center of the pore is thought to be due to to poor wrapping of the particles in this area, giving a lower number of particles used in the measure of the mean square displacement for these bins, that will ultimately lead to a larger diffusion coefficient. This can be avoided by ensuring that the bin size is re-changed after the number of bins is calculated.

We do not observe great differences in the diffusion coefficient as a function of distance from the portlandite surface for the water-in-nanopore system and the CO_2 -bubble-in-water nanopore system, other than that the diffusion coefficient seems to be a bit lower for the CO_2 -bubble-in-water nanopore system, but this is most certainly because the surface is shifted about an Ångstrom for the CO_2 -bubble-in-water nanopore system, relative to the water-in-nanopore system. The reason for such a shift (location of the surface) is due to different expansion during equilibration of the system. The reason for gaining a lower diffusion coefficient in general inside a nanopore, is that the movement of the particles is constrained in the direction normal to the surfaces.

Last but not least, we looked into the diffusion of the carbon dioxide in the nanopore, but did not manage to get a high resolution result. The results on the diffusion of CO_2 in the carbon dioxide-nanopore system were as expected, with a low diffusion coefficient close to the surface, but higher in the center of the pore. But, even though the density at the center of the nanopore would suggest that we should achieve a larger diffusion coefficient, we have to conclude that the movement of the CO_2 particles is still much dependent on the motion close to the surface. We could perhaps anticipated it, because the CO_2 molecules moves over much larger distances before they hit another CO_2 molecule, or the surface

wall, which suggest that their movement is dependent on interactions that could have happened further away than for molecules in the much denser water. For the CO_2 -bubble-in-water nanopore system we saw that the diffusion coefficient for the carbon dioxide in the center of the pore were close to that obtained for CO_2 in water for the mixed systems.

7.4 Prospects for future work

We have looked into the transport properties of water and carbon dioxide confined within a portlandite nanopore, and seen that the properties is strongly dependent on the distance to the surface at the nano-scale level. This study of the diffusion coefficient as a function of distance to the surface, can easily be enhanced by getting more estimates on the diffusion coefficient at the different locations. We could also look into both larger and smaller pores, but I think that larger pores would be more interesting, because we will then be able to see bulk behavior as well. Larger systems will result in larger and more time-consuming simulations and post-processing of datas.

Calculating the position of the Gibbs surface would be interesting, although I have already seen where it approximately should be for water in the portlandite nanopore from a similar ongoing study. The Gibbs-surface should be located just after the first water-density maxima where the density again becomes that of the bulk density, but is assumed to vary with the wetting ability of the surface material. The Gibbs-surface, is located at z_G , and is the surface interface between two substances, and can be found from:

$$\int_{-\infty}^{z_G} (\rho(z) - \rho_{1b}) dz = \int_{z_G}^{\infty} (\rho(z) - \rho_{2b}) dz$$

where ρ_{1b} is the bulk density at the left side of the surface, and ρ_{2b} is the bulk density at the right, and $\rho(z)$ is the density of the material inwhich you would like to know the location of the surface interface. We should ideally integrate to infinity, but that is not possible in real life, so we have to cut off the integration somewhere where the density is sufficiently close to the bulk density. The method can be studied further in Levitas[31].

When I first started to work on this thesis, I wanted to study density driven flow in a nanopore system. This proved to be too much work after two months of hacking *LAMMPS*. Nonequilibrium molecular dynamics (NEMD) has to be performed, keeping the the density at one location constant, and at another location, some distance apart, at a different density. The number of particles in the system should be constant, so for every particle removed, a particle should

be added elsewhere. The thermostats applied to the system, should only make use of the kinetic energy/velocity in the directions where no flow is anticipated. And we cannot have periodic boundary conditions in the direction of the flow. Using *LAMMPS*, at the moment, we have to chose the z-direction to look at flow, because the *kspace* styles can only be applied in a non-periodic system if the non-periodic direction is the z-direction. Further we cannot have particles passing through the z-boundary at all if we would like to look at density driven flow, for a gravitational driven flow however, this can be allowed.

There are of course many other things that would have been interestin to look into, like crystal growth, and the importance of salt or hydrocarbons present in the nanopore system. I think that molecular dynamics have shown us that it can be used to gain insight, and strengthen our understanding of systems where the nanostructure is important for the large scale behavior.

Part III

Appendices

Appendix A

Force fields

A.1 Force Field parameters

In this section, parameters for different potentials are presented as they were used in this thesis. All the forcefields were tested, but some were decided not to proceed with. All tabulated values are presented for the equations used by *LAMMPS*, so that all the values in the tables can be implemented directly into *LAMMPS* using (what *LAMMPS* call) *real* units. The *Lennard-Jones* potential that is used, is the rightmost equation of eq. (2.11), and the mixing rules are presented below, eqs. (A.1) and (A.2). The Coulomb potential used in *LAMMPS* is just the same as the one presented in eq. (2.12), the bond potential and angle potential are the ones shown in eqs. (2.13) and (2.14).

$$\sigma_{ij} = \frac{1}{2}(\sigma_{ii} + \sigma_{jj}) \quad (\text{A.1})$$

$$\epsilon_{ij} = (\epsilon_{ii}\epsilon_{jj})^{1/2} \quad (\text{A.2})$$

When it comes to the mixing rules presented above in eqs. (A.1) and (A.2), is the arithmetic mixing of force field distances σ , and the geometric mixing of force field energy-adjustments ϵ , numbers that should be used for interactions between mixing of atoms affected by the force field. For example, you have the force field values for carbon, C , and for oxygen, O . Then the *van der Waal* interaction between the oxygen and the carbon is described by the Lennard-Jones potential, using the values obtained from the mixing rules.

A.1.1 ClayFF

To use the Clay Force Field in *LAMMPS*, we have to apply a Lennard-Jones potential along with the Coulomb potential with long-range interactions, where we define a cut-off radius for both. I used the pair style `lj/cut/coul/long` with a cut-off at 12 Ångstrom. Also a harmonic bond and a angle potential is used for the *OH-bonds*. In fact, in the earliest work [12], a modified three-body Weber-Stilinger potential were used to model the harmonic behavior, but was later found to be unnecessary and by changing to a simple harmonic oscillator potential reduce the computational cost enormously. Also, relatively small time steps (0.1 fsec) had to be used for stability. To comparison, using *ClayFF* combined with the shake-algorithm when running for water, we can use a 2 fsec time step, and, according to [12], the systems seems to be equilibrated already after 20 ps . The Clay Force Field model, hereafter referred to as *ClayFF*, were developed by Cygan et al.[12, 13, 14, 46] to effectively model the behavior of clay minerals in molecular dynamics simulations. It is based on an ionic, nonbonded, description of the metal-oxygen interactions associated with hydrated phases, and as already said, it consists of four potentials, the *Lennard-Jones* potential 2.11, the *Coulomb* potential 2.12, a *Bond* potential 2.13 and a *Angle* potential 2.14. The empirical force-field parameters for these potentials, are given for a range of atoms, that are optimized by fitting the parameters to reproduce well-known structures, and the partial charges were calculated using periodic density functional theory.

Tabulated values for the Portlandite used in this thesis is given in table A.1. *ClayFF* values for other materials are tabulated in [12]. A *moltemplate* file for a portlandite unitcell, using *ClayFF* can be found at my github account: <https://github.com/goranbs/Master/tree/master/moleculetemplates>, along with other *moltemplate* files.

A.1.2 Water models

There exists many different water models, some are customarily made for ice, others for vapor, some are flexible, other rigid. What water model to choose depends on what you want to look at, of course it is nice to know what water model to choose for your case! The three-point models SPC and SPCE water models are the ones most widely used along with the four point model TIP-4P, so therefore these models are presented and discussed here. Tabulated values for many different water models are found at London South Bank University web pages¹.

¹Force field models for water [\(28.03.2015\)](http://www1.lsbu.ac.uk/water/water_models.html) http://www1.lsbu.ac.uk/water/water_models.html

Table A.1: ClayFF force field parameters for portlandite $Ca(OH)_2$. [12]. Makes use of harmonic oscillator potentials for bond and angle terms, the Lennard-Jones potential, and Coulomb

Nonbond		Bond	
q_{Ca}	+1.050 e	$K_{1,OH}$	554.1349 Kcal/mol/ \AA^2
q_O	-0.820 e	$r_{o,OH}$	1.0 \AA
q_H	+0.425 e	$K_{2,HOH}$	30.0 Kcal/mol/rad ²
ϵ_{Ca}	5.0298 E-6 Kcal/mol	$\theta_{0,HOH}$	109.47 deg
ϵ_H			
ϵ_O	0.1554 Kcal/mol		
σ_{Ca}	5.562 \AA		
σ_H			
σ_O	3.1655 \AA		

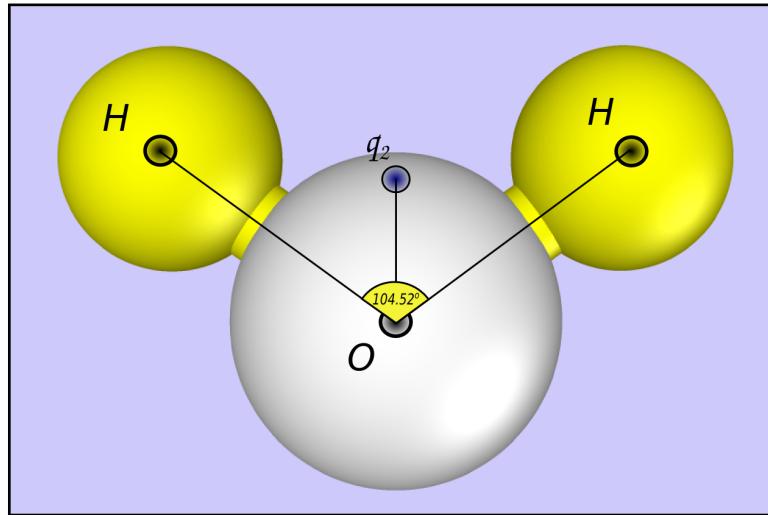


Figure A.1: The water models presented here assign three, and four points in the structure of the water molecule. SPC and SPCE place their charges in the center of the atoms, while the TIP-4P place the charge q_2 , of the oxygen atom, a distance σ_{q2} from the center

SPC and SPCE

Single Point Charge models. The name comes from the fact that each point particle is assigned a charge, located at the center of the particle. The only difference between the SPC and the SPCE model is the net charges of the atoms. Not very surprisingly, this leads to different results when it comes to thermodynamic properties. The SPC model overestimates the diffusion coefficient by almost a factor of 2, while the SPCE model overestimates it by approximately 15%. An-

Table A.2: SPCE and SPC force field parameters. H_2O .[12]

Nonbond		Bond	
q_H	+0.4238 e (+0.41 e SPC)	$K_{1,OH}$	554.1349 Kcal/mol Å ²
q_O	-0.8476 e (+0.82 e SPC)	$r_{o,OH}$	1.0 Å
ϵ_H	0.0 Kcal/mol	$K_{2,HOH}$	45.7696 Kcal/mol rad ²
ϵ_O	0.1553 Kcal/mol	$\theta_{0,HOH}$	109.47 deg
σ_H			
σ_O	3.166 Å		

Table A.3: TIP-4P force field parameters. H_2O .[1], values crosschecked with NAMD and GROMACS input files

Nonbond		Bond	
q_H	+0.52 e	$K_{1,OH}$	450.0 Kcal/mol Å ²
q_O	-1.04 e	$r_{o,OH}$	0.9572 Å
ϵ_H	0.0 Kcal/mol	$K_{2,HOH}$	55.0 Kcal/mol rad ²
ϵ_O	0.1549 Kcal/mol	$\theta_{0,HOH}$	104.52 deg
σ_H		ϕ_{0,HOq_2}	52.26 deg
σ_O	0.9572 Å		
σ_{q_2}	0.15 Å		

other important note, is that the density of water is approximately 0.2% lower than experimental values, for both the models. See [33] for more information on comparing the two models and the TIP-3P model.

A *moltemplate* file for SPCE water is shown in listing B.1, and can be used by *moltemplate* to create a input file for *LAMMPS*.

TIP-4P

The TIP-4P model is a four point model for a three-particle molecule, where the charge of the oxygen atom has been placed at the fourth point, located a distance σ_{q_2} from the oxygen atom, and with an angle ϕ_{0,HOq_2} , between the OH -bond and the Oq_2 -line, as can be seen in fig. A.1, where the fourth point is blue, and marked with the charge q_2 at this site. The reason for placing the charge of the oxygen atom here is because of quantum mechanical simulations, showing the charge distribution for the water molecule. From the electron density plot, the charge minima associated with the oxygen atom is placed at this point.

A.1.3 Models for carbon dioxide

Three models for carbon dioxide have been tested. The *FPF* model is a flexible three-point model, while *TraPPE* and the *EMP2* models are rigid three-point models. When looking at interaction with a surface, using a rigid molecule, it can be difficult to observe whether or not the molecule will attach and form some kind of structure. But again, when simulating systems where no reactions are expected, such a model is computationally faster.

TraPPE

TraPPE - Transferable Potentials for Phase Equilibria², is developed by a statistical mechanics research group at the University of Minnesota. They focus on the development and improvement of transferable force fields. Transferable because the force field parameters should be transferable between different molecules at the same thermodynamic conditions. A transferable model will have a greater power of predicting behavior of new, more complex and different systems, than specialized models that normally only got good predictions for the special case that they were designed for. The Siepmann Group consider it being more interesting, making a good model that can be used for a wide range of applications, than a narrow, but highly accurate model for a special case.

In the *TraPPE* model, the angle of the CO_2 molecule should be held fixed. This is done in *LAMMPS*, using the command shown in listing A.1. The command is a *fix* with the *fix-ID*, RIGID_NVE, which should work on all atoms in the *group*, named *co2*. The *method/style* is *rigid/nve/small* with the required argument *molecule*. First of all, this is for a run in the NVE ensemble (other styles are available for other ensembles), and the *rigid/small* style is best for a system with a large number of small rigid bodies.

```
fix    RIGID_NVE co2 rigid/nve/small molecule
```

Listing A.1: fix for keeping the molecule rigid during the simulation

The *TraPPE* model has force field parameters for alkanes, alkenes, alcohols, thiols, ethers, sulfides, ketons, aldehydes, esters and many more groups of molecules.

²The Siepmann Group: <http://www.chem.umn.edu/groups/siepmann/trappe/molname.php>, available 24.02.2015

Table A.4: TraPPE Force Field parameters for CO_2 .^[37] The model should be used with a cut-off radius of 14 Å, and use Ewald summation or a PPPM calculation

Nonbond		Bond	
q_C	+0.700 e	$K_{1,CO}$	fixed bond length
q_O	-0.350 e	$r_{o,CO}$	1.160 Å
ϵ_C	5.365 E-2 Kcal/mol	$K_{2,OCO}$	rigid angle
ϵ_O	0.157 Kcal/mol	$\theta_{0,OCO}$	180.0 deg
σ_C	2.4945 Å		
σ_O	2.7172 Å		

Table A.5: EMP2 force field parameters for CO_2 .^[7, 37]

Nonbond		Bond	
q_C	+0.6512 e	$K_{1,CO}$	fixed bond length
q_O	-0.3256 e	$r_{o,CO}$	1.160 Å
ϵ_C	5.590 E-2 Kcal/mol	$K_{2,OCO}$	rigid angle
ϵ_O	0.160 Kcal/mol	$\theta_{0,OCO}$	180.0 deg
σ_C	2.456 Å		
σ_O	2.702 Å		

EMP2

The *EMP2* force field for carbon dioxide, is a rigid model, keeping the angle of the carbon dioxide fixed at 180 deg. Locking the angle in *LAMMPS* is done by the command shown in listing A.1. The command is explained in the text about the TraPPE force field, or you can browse the *LAMMPS* manual for more information. The values presented in table A.5 are from Princeton University Colleges webpage³, but are backed up with information from [7, 37].

A.1.4 FPF

The *Flexible Point Field* model for carbon dioxide, were developed to do model CO_2 in a way that can be used in aqueous solutions and mineral systems. In such systems, it is important to have a flexible molecule so that interactions with surfaces and transport rates are predicted at a higher precision. The bond lengths and angles are preserved compared with experiments, even for supercritical CO_2 . But the force constant is believed to be incorrectly predicted because the model

³EMP2 [\(28.03.2015\)](http://paros.princeton.edu/jerring/gibbs_old/results/co2_emp2/co2_emp2.html)

Table A.6: FPF force field parameters for CO_2 . A flexible model for carbon dioxide.[12]

Nonbond		Bond	
q_C	+0.6512 e	$K_{1,CO}$	1009 Kcal/mol Å ²
q_O	-0.3256 e	$r_{o,CO}$	1.162 Å
ϵ_C	0.0559 Kcal/mol	$K_{2,OCO}$	54.0 Kcal/mol rad ²
ϵ_O	0.1597 Kcal/mol	$\theta_{0,OCO}$	180.0 deg
σ_C	2.456 Å		
σ_O	2.702 Å		

Table A.7: Morse force field parameters for SiO_2 .[29]

Morse coefficients	
q_{Si}	1.3 e
q_O	-0.65 e
ϵ_{Si}	0.17733 Kcal/mol
ϵ_O	0.5363 Kcal/mol
ϵ_{Si-O}	45.997 Kcal/mol
σ_{Si}	3.3496 Å
σ_O	3.3774 Å
σ_{Si-O}	1.1504 Å
α_{Si}	2.0446 1/Å
α_O	1.3731 1/Å
α_{Si-O}	2.6518 1/Å

fails to reproduce correct vibrational/frequency spectra. This could lead to incorrect kinetic energy in the system from what is given as the temperature of the system.

A.1.5 Morse

The force field coefficients used in this project for the *Morse* potential force field model are tabulated in table A.7. The values listed in this table is used directly in *LAMMPS* for the “units real” option. The *Morse* potential is a bond potential that is meant to model covalent bonds between atoms. This potential is meant to be used with a cut-off, so that bonds can be broken, and new atoms can be caught in the potential. It is a computationally, much more expensive potential than the harmonic oscillator that it is most often replaced with.

Table A.8: Tersoff force field parameters for CO_2 . Provided from Munetoh et al. and Tersoff 1988 [38, 24] ($\chi_{Si-O} = 1.17945$)

	Si	O
$A(eV)$	1.8308 E+3	1.88255 E+3
$B(eV)$	4.7118 E+2	2.18787 E+2
$\gamma(\text{\AA})^{-1}$	2.4799	4.17108
$\mu(\text{\AA})^{-1}$	1.7322	2.35692
β	1.1000 E-6	1.1632 E-7
n	7.8734 E-1	1.04968
c	1.0039 E-5	6.46921 E-4
d	16.217	4.11127
m	3	3
$\cos(\theta)$	-5.9825 E-1	-8.45922E - 1
$R(\text{\AA})$	2.5	1.7
$S(\text{\AA})$	2.8	2.0
$D(\text{\AA})$	0.2	0.2
$\lambda_1(\text{\AA})^{-1}$	3.2394	3.2394
$\lambda_2(\text{\AA})^{-1}$	1.3258	1.3258
$\lambda_3(\text{\AA})^{-1}$	1.3258	1.3258

A.1.6 Tersoff

The *Tersoff*[24] potential is a *bond order* potential. Unlike a pair potential like the other models presented here, the *Tersoff* potential has a function that weakens the bond ij in the presence of other bonds ik in contact with atom i . This is governed by a function that depends on the angle between the bonds ij and ik .

Tersoff mixing rules, see eq. (A.3). Tersoff parameters must be transformed to *LAMMPS*parameters by the relations $R = \frac{(R'+S')}{2}$ and $D = \frac{(S'-R')}{2}$, where the primes are the Tersoff parameters, and those without are the ones that can be used in *LAMMPS*. Although the Tersoff potential is difficult to implement, it is quite easy to use in *LAMMPS*when you already have the force field parameters.

Example files are available in the *LAMMPS* Starball file structure.

$$\lambda_l^{i,j} = \frac{1}{2}(\lambda_l^i + \lambda_l^j) \quad (\text{A.3a})$$

$$A_{i,j} = (A_i A_j)^{1/2} \quad (\text{A.3b})$$

$$B_{i,j} = \chi_{ij} (A_i A_j)^{1/2} \quad (\text{A.3c})$$

$$R_{i,j} = (A_i A_j)^{1/2} \quad (\text{A.3d})$$

$$S_{i,j} = (A_i A_j)^{1/2} \quad (\text{A.3e})$$

Appendix B

Example code and input scripts

In the first section of this appendix, example code and input scripts are given. The minimization procedure used to minimize the system energy of various silica systems, is explained further, and the input script for this procedure is given. The minimum image convention for triclinic systems is discussed, and the method is provided with a code snippet written in C++ that can be helpful. The way of compiling a new *LAMMPS* source code on the Abel computing cluster is also described in detail.

B.1 Scripts and data files

In this section, some example input scripts and data files for *LAMMPS* are provided.

A moltemplate *.lt*-file for SPCE-water is shown in listing B.1. The file can be imported into other moltemplate files, and used to replicate the water molecule. Further moltemplate files can be found at my github account: <https://github.com/goranbs/Master/tree/master/moleculetemplates>. Here also lies example files on how to build more complex systems with moltemplate.

```
# file "spce_simple.lt"
#
#   H1   H2
#     \ /
#       O
#
SPCE {
```

```

write_once("In Init") {
    units real
    atom_style full
    pair_style lj/cut/coul/long 10.35
    bond_style harmonic
    angle_style harmonic
    kspace_style pppm 0.0001
    pair_modify tail yes
}
# "atom_style full" results in the format:
# atomID molID atomType charge coordX coordY coordZ
write("Data Atoms") {
    $atom:O $mol:. @atom:O -0.8476 0.0000000 0.000000 0.0
    $atom:H1 $mol:. @atom:H 0.4238 0.8164904 0.5773590 0.0
    $atom:H2 $mol:. @atom:H 0.4238 -0.8164904 0.5773590 0.0
}

write_once("Data Masses") {
    # atomType mass
    @atom:O 15.9994
    @atom:H 1.008
}
write("Data Bonds") {
    # bondID bondType atomID1 atomID2
    $bond:OH1 @bond:OH $atom:O $atom:H1
    $bond:OH2 @bond:OH $atom:O $atom:H2
}
write("Data Angles") {
    # angleID angleType atomID1 atomID2 atomID3
    $angle:HOH @angle:HOH $atom:H1 $atom:O $atom:H2
}
# --- Force-field parameters and group definitions ---
write_once("In Settings") {
    # -- Non-bonded (Pair) interactions --
    # atomType1 atomType2 parameter-list (epsilon, sigma)
    pair_coeff @atom:O @atom:O 0.1553 3.166
    pair_coeff @atom:H @atom:H 0.0000 1.833

    # -- Bonded interactions --
    # bondType parameter list (k_bond, r0)
    bond_coeff @bond:OH 554.1349 1.0

    # angleType parameter-list (k_theta, theta0)
    angle_coeff @angle:HOH 45.7696 109.47

    # Group definitions and constraints
    group spce type @atom:O @atom:H
    group wo type @atom:O
    group wh type @atom:H

    fix SHAKE spce shake 1.0E-4 100 0 b @bond:OH a @angle:HOH
}

```

```
}
```

```
} # SPCE
```

Listing B.1: Moltemplate for SPCE-water molecule

In listing B.2, positional data about a bulk triclinic SiO_2 system is provided. The system can easily be replicated in *LAMMPS* to create a large bulk α -quartz using the command `replicate`. This `.data` -file is set up according to `atom_style charge`, and supposed to be used for a *ReaxFF* simulation.

```
# SiO2 unit cell
 9 atoms
 2 atom types

0 5.0152997971 xlo xhi
0 4.3433770318 ylo yhi
0 5.5059700012 zlo zhi
-2.5076498985 0 0 xy xz yz

Masses

1 15.9994
2 28.0855

Atoms

 1 1 1 -1.05 0.24408 1.80187 4.37473
 2 1 1 -1.05 0.82515 3.65382 0.70409
 3 1 1 -1.05 -1.06923 3.23106 2.53941
 4 1 1 -1.05 1.43842 1.11231 1.13124
 5 1 1 -1.05 3.33280 0.68956 2.96656
 6 1 1 -1.05 2.75173 2.54151 4.80188
 7 1 2 2.1 1.31324 2.27460 0.00000
 8 1 2 2.1 -1.19441 2.06877 3.67065
 9 1 2 2.1 2.38881 0.00000 1.83532
```

Listing B.2: Bulk alpha-quartz unit cell. Atom style full

In the short example input script found in listing B.3, some important things are shown. First of all, the script is written from the top down, so the information about the system have to be provided at the top of the file. Some variable names are created. These can be called in two different ways, one that is static, with the dollar sign and the curly braces, which gives you the initial value of the variable, and the other with a prefix `v_` in front of the name. For example you may write `v_Nprint`.

The `dump` command tells what information that should be dumped to a dump file

every `Nstep`-time step. The `thermo_style` command states what is to be printed to screen, and the `thermo` command states how often information is printed to screen.

The velocity command initiates velocities to `all` particles in the system given a temperature, a random seed, and the method used to draw the velocities from some distribution. One may run a `minimize` if the simulation is tough to get started because particles are too close to each other initially. The `minimize` function will then try to move the particles and find a configuration that is close to the original one, but where the total energy of the system is lower than it was initially. We have to specify the ensemble, here an example using `fix` to specify that we would like to run our simulation in the *NVE*-ensemble, holding the volume and the energy of the system constant. The simulation in this ensemble is carried out for `Nstep`-time steps. The input script runs a *ReaxFF* simulation as stated by the `fix qeq` command, and two files have to be added in the input script. These are the `ffield.reax.SiO2` which is called by the `SiO2_REAX.param` file in the `include` statement. The force field file is quite large, so it is not included here, but can be found at my github page: <https://github.com/goranbs/Master/tree/master/moleculetemplates>.

```

boundary      p p p
units         real
atom_style    full
pair_style    lj/cut/coul/long 7.75
neigh_modify  every 1 delay 0 check yes one 3000
pair_modify   tail no

read_data     sio2.data
include       SiO2_REAX.param

replicate 2 2 2

variable Nprint equal 10
variable dt equal 1.0
variable Nstep equal 1000
variable dumpState string dump.alpha-quartz.*.txt

dump 1 all custom ${Nprint} ${dumpState} id mol type x y z

#----- RUNTIME -----
#
fix          QEQ all qeq/reax 1 0.0 8.0 1.0e-6 reax/c

timestep     ${dt}           # fsec 10E-15s
thermo_style custom step temp pe press
thermo       ${Nprint}        # output at every

```

```

velocity      all create 300.0 70690 dist gaussian
minimize      1.0e-4 1.0e-6 100 1000
fix           NVE all nve
run           ${Nstep}

write_data    sio2.data
write_restart  sio2.restart

```

Listing B.3: Short example input script

B.2 Minimization procedure

The minimization procedure explained in this thesis resulted in a *LAMMPS* input script that easily can be added a header part where information about, among other things, the number of time steps are presented. The looping procedure shown in listing B.4 assumes that the keyword `SELF` corresponds to the file name of this script. E.g. if this script is named `in.loop`, then make sure to replace `SELF = in.loop`.

There are several things that needs to be explained in this code snippet. First of all, from the top, `run 0` is provided such that a variable holding the initial potential can be set. A label just specify a line number in the script so that the `jump` command can be used to jump to this line given the name of the file. The command `variable i loop 9` just creates a list [1, 2, 3, 4, 5, 6, 7, 8, 9] where the command `next i` jumps to the next value in the list, `i`.

```

run          0
variable Ep_initial equal ${Ep}
label loopa
variable ii equal 100
variable jj equal 100
variable i loop ${ii}
  variable Epot equal ${Ep}
  timestep 0.1
  fix       NVE all nve
  fix       BER all temp/berendsen ${Tstart} ${Tstart} ${tc}
  run       ${Nsteps_nve}
  timestep 1.0
  run       ${Nsteps_nve2}
  unfix    NVE
  unfix    BER
label loopb
variable j loop ${jj}

```

```

variable T equal ${Tstart}-$j+1 # (T!=0)

fix      NVT all nvt temp ${T} ${T} ${tc}
run      ${Nsteps}
unfix    NVT

print "#-----#"
print "  Temp = $T [K]"
print "  Epot = ${Ep} [kcal/mol]"
print "  Ep_i = ${Epot} [kcal/mol]"
print "  i = ($i / ${ii})  j = ($j / ${jj})"
print "#-----#"

next j
jump    SELF loopb
label   breakb

variable dEp equal abs(${Ep}-${Epot})
if "${dEp}<${Tolerance}" then &
"print '#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-' &
"print '  Ep_f = ${Ep} [kcal/mol],' &
"print '  Ep_i = ${Ep_initial} [kcal/mol],' &
"print '  dEp = ${dEp} , tolerance = ${Tolerance}, ' &
"print '#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-' &
"write_restart Arestart.state_where_tolerance_is_met" &
quit

minimize 1.0e-6 1.0e-6 10000 10000
next i
jump in.loop SELF
label breaka

```

Listing B.4: Minimization procedure in LAMMPS

B.3 Minimum image convention for triclinic systems

For a triclinic system, we also have to make sure to change the tilt factors in the system vectors. The way *LAMMPS* does this, is shown in the code snippet listing B.5.

For a triclinic system, we have like in a Cartesian coordinate system, given three system vectors, **a**, **b** and **c**. Where **b** lies in the xy-plane, with a tilt factor $yx = b_2$ relative to **a**. **c** has a positive unit normal according to the xy-plane and tilt factors $xz = c_1$ and $yz = c_2$ relative to **a** and **b** vector respectively.

The vectors can be expressed like in eq. (B.1).

$$\mathbf{a} = (a_1, a_2, a_3) \quad (\text{B.1a})$$

$$\mathbf{b} = (b_1, b_2, b_3) \quad (\text{B.1b})$$

$$\mathbf{c} = (c_1, c_2, c_3) \quad (\text{B.1c})$$

In the code snippet *domain.cpp*, **delta** holds the positional vectors. The length of the system vectors are $a = xprd, b = yprd, z = zprd$, whereas the tilt factors are, as earlier pointed out, yx, xz, yz .

```

if (zperiodic) {
    if (fabs(delta[2]) > zprd_half) {
        if (delta[2] < 0.0) {
            delta[2] += zprd;
            delta[1] += yz;
            delta[0] += xz;
        }
        else {
            delta[2] -= zprd;
            delta[1] -= yz;
            delta[0] -= xz;
        }
    }
}
if (yperiodic) {
    if (fabs(delta[1]) > yprd_half) {
        if (delta[1] < 0.0) {
            delta[1] += yprd;
            delta[0] += yx;
        }
        else {
            delta[1] -= yprd;
            delta[0] -= yx;
        }
    }
}
if (xperiodic) {
    if (fabs(delta[0]) > xprd_half) {
        if (delta[0] < 0.0) delta[0] += xprd;
        else delta[0] -= xprd;
    }
}
}

```

Listing B.5: code snippet in domain.cpp for minimum image convention for triclinic system

B.4 The Abel computing cluster

Most of the simulations done in this thesis were done on the super computer Abel (at least it was a super computer in 2015). All compute nodes on the cluster have a minimum of 64 GB RAM, 16 physical CPU cores and are connected by FDR (56 Gbps) Infiniband which makes the transferring of information very fast. It existed only an old version of *LAMMPS* on the computer cluster, so then I uploaded the new version locally and compiled an executable compatible with the CentOS operating system at Abel, using Intel CPU's. The steps are presented in the following:

1. download latest *LAMMPS* version
2. `src path/to/lammps username@abel.uio.no:~/lammps`
3. `ssh username@abel.uio.no`
4. `cd lammps/src`
5. now install the packages you want. Follow instructions using the command `make`
6. load necessary modules for the compilation. This is done by the command `module lode <name-of-module>`. You will have to load:
 7. `module load intel`
 8. `module load openmpi.intel`
 9. `module load intelmpi.intel`
10. compile *LAMMPS* for the machine you want. Abel is using CentOS, and you will have to have the package *USER-INTEL*. Then compile with:
 11. `make intel_cpu`

There were an update on Abel that created a temporary problem with the execution of the executable `lmp_intel_cpu`. This can be omitted (but with a executable that runs slower using openMP) with a recompilation of the packages that you want, except the `user-intel`-package! And only loading the `intel` and the `openmpi.intel`-modules. Then you execute the make command `make ompi_icc`, which makes an executable called `lmp_ompi_icc`.

Bibliography

- [1] J. L. Abascal and C. Vega. “A general purpose model for the condensed phases of water: TIP4P/2005.” In: *The Journal of chemical physics* 123.23 (2005), p. 234505. ISSN: 00219606. DOI: [10.1063/1.2121687](https://doi.org/10.1063/1.2121687).
- [2] M. Bentham and G. Kirby. “CO₂ Storage in Saline Aquifers”. In: *Dossier* 60 (2005). URL: <http://science.uwaterloo.ca/~mauriced/earth691-duss/CO2\GeneralCO2Sequestrationmaterilas/CO2\DeepSalineAuifers\benthman\vol60n3.pdf>.
- [3] H J C Berendsen et al. “Molecular dynamics with coupling to an external bath”. In: *The Journal of Chemical Physics* 81.8 (1984), pp. 3684–3690. ISSN: 00219606. DOI: [10.1063/1.448118](https://doi.org/10.1063/1.448118). URL: [http://link.aip.org/link/JCPA6/v81/i8/p3684/s1\&Agg=doi\\$\\backslash\\$npapers2://publication/doi/10.1063/1.448118](http://link.aip.org/link/JCPA6/v81/i8/p3684/s1\&Agg=doi$\\backslash$npapers2://publication/doi/10.1063/1.448118).
- [4] Alexandru Bot et al. “Hydrodynamics in Clay Nanopores”. In: (2011), pp. 16109–16115.
- [5] IC Bourg and CI Steefel. “Molecular dynamics simulations of water structure and diffusion in silica nanopores”. In: *The Journal of Physical Chemistry C* (2012), pp. 2–10. URL: <http://pubs.acs.org/doi/abs/10.1021/jp301299a>.
- [6] W. F. BRACE and J. B. WALSH. “SOME DIRECT MEASUREMENTS OF THE SURFACE ENERGY OF QUARTZ AND ORTHOCLASE”. In: *The American Mineralogist* 47 (1962). URL: https://documents.htracyhall.org/ocr/HTH-Archives/Cabinet3/PatentsResearch\%28StructuresofHighPressureCompounds\%29/Brace,W.F\%28linked\%29/Brace,W.F.-439_OCR.pdf.
- [7] Georgios Chatzis and Jannis Samios. “Binary mixtures of supercritical carbon dioxide with methanol. A molecular dynamics simulation study”. In: *Chemical Physics Letters* 374.1-2 (2003), pp. 187–193. ISSN: 00092614. DOI: [10.1016/S0009-2614\(03\)00640-7](https://doi.org/10.1016/S0009-2614(03)00640-7).

- [8] Kimberly Chenoweth, Adri C T van Duin, and William a Goddard. “ReaxFF reactive force field for molecular dynamics simulations of hydrocarbon oxidation.” In: *The journal of physical chemistry. A* 112.5 (2008), pp. 1040–1053. ISSN: 1520-5215. DOI: [10.1021/jp709896w](https://doi.org/10.1021/jp709896w).
- [9] Kimberly Chenoweth et al. “Simulations on the Thermal Decomposition of a Poly (dimethylsiloxane) Polymer Using the ReaxFF Reactive Force Field Simulations on the Thermal Decomposition of a Poly (dimethylsiloxane) Polymer Using the ReaxFF Reactive”. In: *Journal of American Chemical Society* 1 (2005), pp. 7192–7202. DOI: [10.1021/ja050980t](https://doi.org/10.1021/ja050980t). URL: <http://pubs.acs.org/doi/abs/10.1021/ja050980t>.
- [10] S. J. Choe, Dong Ho Park, and Do Sung Huh. “Adsorption and Dissociation Reaction of Carbon Dioxide on Pt(111) and Fe(111) Surface: MO-study”. In: *Bull. Korean Chem. Soc.* 21 (2000).
- [11] Hitachi High-Technologies Coorporation. *Let's familiarize ourselves with the TEM*. Tokyo: Hitachi High-Technology Cooperation, 2009, p. 57. URL: <http://www.hitachi-hitec.com/em/world>.
- [12] Randall T. Cygan, Jian-Jie Liang, and Andrey G. Kalinichev. “Molecular Models of Hydroxide, Oxyhydroxide, and Clay Phases and the Development of a General Force Field”. In: *The Journal of Physical Chemistry B* 108.4 (Jan. 2004), pp. 1255–1266. ISSN: 1520-6106. DOI: [10.1021/jp0363287](https://doi.org/10.1021/jp0363287). URL: <http://pubs.acs.org/doi/abs/10.1021/jp0363287>.
- [13] Randall T. Cygan, Vyacheslav N. Romanov, and Evgeniy M. Myshakin. “Molecular Simulation of Carbon Dioxide Capture by Montmorillonite Using an Accurate and Flexible Force Field”. In: *The Journal of Physical Chemistry C* 116.24 (June 2012), pp. 13079–13091. ISSN: 1932-7447. DOI: [10.1021/jp3007574](https://doi.org/10.1021/jp3007574). URL: <http://pubs.acs.org/doi/abs/10.1021/jp3007574>.
- [14] Randall T. Cygan et al. “Molecular models and simulations of layered materials”. In: *Journal of Materials Chemistry* 19.17 (2009), p. 2470. ISSN: 0959-9428. DOI: [10.1039/b819076c](https://doi.org/10.1039/b819076c). URL: <http://xlink.rsc.org/?DOI=b819076c>.
- [15] Svenn-Arne Dragly. “Bridging Quantum Mechanics and Molecular Dynamics with Artificial Neural Networks”. PhD thesis. University of Oslo, 2014, p. 191.
- [16] A. Einstein. *Investigations on the theory of the Brownian Motion*. Dover Publications Inc., 1926.
- [17] US. Department of Energy. *Earth Science Division*. 2015. URL: <http://esd.lbl.gov/research/programs/gcs/outreach.html> (visited on 05/13/2015).
- [18] D Frenkel and B Smit. *Understanding molecular dynamics simulations*. 2nd Editio. Academic Press Inc., 2001. URL: <http://www.chim.unifi.it/~procacci/CFS/LIBRI/UnderstandingMolecularSimulations.pdf>.

- [19] Geocaching. *Chattanooga shale*. 2014. URL: <http://www.geocaching.com/geocache/GC1MX4X\chattanooga-shale?guid=2e29bafa-e928-401eb42a-1d5592c80a31> (visited on 05/12/2015).
- [20] P. Hohenberg and W. Kohn. “Inhomogeneous Electron Gas”. In: *Physical Review* 136 (1964).
- [21] William G Hoover. “Canonical dynamics: Equilibrium phase-space distributions”. In: 31.3 (1985), pp. 1695–1697.
- [22] JEOL solutions for Innovation. *Harvard 3D SEM imaging*. 2013. URL: <http://www.jeolusa.com/RESOURCES/ElectronOptics/ImageGallery/tabid/323/galleryType/SlideShow/ItemID/230/Default.aspx> (visited on 05/14/2015).
- [23] Jacob Israelachvili. *Intermolecular & Surface Forces*. 2nd editio. Elsevier Ltd. Academic Press, 1991, p. 437. ISBN: 012375181-0.
- [24] Thomas J. Watson J. Tersoff. “E(\mathbf{r}_{ij})”, in: 37.12 (1988). URL: <http://journals.aps.org/prb/pdf/10.1103/PhysRevB.37.6991>.
- [25] Andrew Jewett. “Moltemplate Manual”. In: (2014), pp. 1–69. URL: <http://moltemplate.org/>.
- [26] W. Kohn and L. J. Sham. “Self-consistent Equations Including Exchange and Correlation Effects”. In: *Physical Review* 385.1951 (1965).
- [27] Chang Lyoul Kong. “Combining rules for intermolecular potential parameters. II. Rules for the Lennard-Jones ($12-6$) potential and the Morse potential”. In: *The Journal of Chemical Physics* 59.5 (1973), p. 2464. ISSN: 00219606. DOI: [10.1063/1.1680358](https://doi.org/10.1063/1.1680358). URL: <http://scitation.aip.org/content/aip/journal/jcp/59/5/10.1063/1.1680358>.
- [28] Anant D. Kulkarni et al. “Oxygen interactions with silica surfaces: Coupled cluster and density functional investigation and the development of a new ReaxFF potential”. In: *Journal of Physical Chemistry C* 117.1 (2013), pp. 258–269. ISSN: 19327447. DOI: [10.1021/jp3086649](https://doi.org/10.1021/jp3086649).
- [29] Sabine Leroch and Martin Wendland. “Simulation of forces between humid amorphous silica surfaces: A comparison of empirical atomistic force fields”. In: *Journal of Physical Chemistry C* 116 (2012), pp. 26247–26261. ISSN: 19327447. DOI: [10.1021/jp302428b](https://doi.org/10.1021/jp302428b).
- [30] Benjamin G. Levine, John E. Stone, and Axel Kohlmeyer. “Fast analysis of molecular dynamics trajectories with graphics processing units-Radial distribution function histogramming”. In: *Journal of Computational Physics* 230.9 (2011), pp. 3556–3569. ISSN: 00219991. DOI: [10.1016/j.jcp.2011.01.048](https://doi.org/10.1016/j.jcp.2011.01.048). URL: <http://dx.doi.org/10.1016/j.jcp.2011.01.048>.

- [31] Valery I. Levitas. “Unambiguous Gibbs dividing surface for nonequilibrium finite-width interface: Static equivalence approach”. In: *Physical Review B* 89.9 (2014), p. 094107. ISSN: 1098-0121. DOI: [10.1103/PhysRevB.89.094107](https://doi.org/10.1103/PhysRevB.89.094107). URL: <http://link.aps.org/doi/10.1103/PhysRevB.89.094107>.
- [32] Oleksandr I. Malyi, Vadym V. Kulish, and Clas Persson. “In search of new reconstructions of (001) α -quartz surface: a first principles study”. In: *RSC Adv.* 4 (2014), pp. 55599–55603. ISSN: 2046-2069. DOI: [10.1039/C4RA10726H](https://doi.org/10.1039/C4RA10726H). URL: <http://xlink.rsc.org/?DOI=C4RA10726H>.
- [33] Pekka Mark and Lennart Nilsson. “Structure and dynamics of the TIP3P, SPC, and SPC/E water models at 298 K”. In: *Journal of Physical Chemistry A* 105 (2001), pp. 9954–9960. ISSN: 10895639. DOI: [10.1021/jp003020w](https://doi.org/10.1021/jp003020w).
- [34] V Marry et al. “How Electrostatics Influences Hydrodynamic Boundary Conditions: Poiseuille and Electro-osmotic Flows in Clay Nanopores.” In: (2013).
- [35] Glenn J Martyna and Michael L Klein. “Nosè-Hoover chains: A canonical ensemble via continuous dynamics”. In: (1992), pp. 2635–2644. URL: http://quantum.ch.ntu.edu.tw/ycclab/wp-content/uploads/2015/02/Nose-Hoover_chains_1992_7627611.pdf.
- [36] Thorsten Merker et al. “Molecular model for carbon dioxide optimized to vapor-liquid equilibria.” In: *The Journal of chemical physics* 132.23 (June 2010), p. 234512. ISSN: 1089-7690. DOI: [10.1063/1.3434530](https://doi.org/10.1063/1.3434530). URL: <http://www.ncbi.nlm.nih.gov/pubmed/20572726>.
- [37] Othonas A Moultos et al. “Atomistic Molecular Dynamics Simulations of CO₂ Diffusivity in H₂O for a Wide Range of Temperatures and Pressures”. In: (2014).
- [38] Shinji Munetoh et al. “Interatomic potential for Si-O systems using Tersoff parameterization”. In: *Computational Materials Science* 39 (2007), pp. 334–339. ISSN: 09270256. DOI: [10.1016/j.commatsci.2006.06.010](https://doi.org/10.1016/j.commatsci.2006.06.010). URL: [#](http://www.sciencedirect.com/science/article/pii/S0927025606002023).
- [39] Sandia National et al. “LAMMPS Users Manual 9”. In: 2003 (2014). URL: lammps.sandia.gov.
- [40] V. Adrian Parsegian. *Van der Waals Forces: A Handbook for Biologists, Chemists, Engineers, and Physicists*. Cambridge University Press, 2006, p. 380. ISBN: 0-521-83906-8. URL: <https://books.google.no/books?id=K4F7uHJVEOoC&printsec=frontcover&dq=van+der+waals&hl=no&sa=X&ei=aGMhVdTUEcniO5OXgaAF&ved=0CB0Q6AEwAA#v=onepage&q=dipole&f=false>.

- [41] Marcos E Perez-Blanco and Edward J Maginn. “Molecular dynamics simulations of carbon dioxide and water at an ionic liquid interface.” In: *The journal of physical chemistry. B* 115 (2011), pp. 10488–10499. ISSN: 1520-5207. DOI: [10.1021/jp203838j](https://doi.org/10.1021/jp203838j).
- [42] Stefano Piana, John L. Klepeis, and David E. Shaw. “Assessing the accuracy of physical models used in protein-folding simulations: Quantitative evidence from long molecular dynamics simulations”. In: *Current Opinion in Structural Biology* 24.1 (2014), pp. 98–105. ISSN: 0959440X. DOI: [10.1016/j.sbi.2013.12.006](https://doi.org/10.1016/j.sbi.2013.12.006). URL: <http://dx.doi.org/10.1016/j.sbi.2013.12.006>.
- [43] S. J. Plimpton. “Fast Parallel Algorithms for Short-Range Molecular Dynamics”. 1995. URL: <http://www.sandia.gov>.
- [44] D. T. Pritchard and J. A. Currie. “Diffusion of coefficients of carbon dioxide, nitrous oxide, ethylene and ethane in air and their measurement”. In: *Journal of Soil Science* (1982). URL: <http://onlinelibrary.wiley.com/doi/10.1111/j.1365-2389.1982.tb01757.x/pdf>.
- [45] Anthony K. Rappé and William a. Goddard III. “Charge Equilibration for Molecular Dynamics Simulations”. In: *The Journal of Physical Chemistry* 95.8340 (1991), pp. 3358–3363. ISSN: 0022-3654. DOI: [10.1021/j100161a070](https://doi.org/10.1021/j100161a070).
- [46] Sandia Report. “Natural Materials for Carbon Capture”. In: November (2010).
- [47] Thomson Reuters. *World Map assessed shale oil*. 2013. URL: <http://blog.thomsonreuters.com/index.php/tag/shale-gas/> (visited on 05/11/2015).
- [48] R. G. Rothwell. “Pressure coring, logging and subsampling with the HYACINTH system”. In: *New Techniques in Sediment Core Analysis*. 2006. URL: https://books.google.no/books?id=D2xQt7xXU1UC&pg=PA151&lpg=PA151&dq=hyperbaric+geology+testing+system&source=bl&ots=y85Mo3_K2v&sig=3BjVi8G76MkPaMJKBCQUOB-qg2E&hl=no&sa=X&ei=chpTVdn_B8W2swGX4CwBA&ved=0CG0Q6AEwCQ#v=onepage&q=hyperbaricgeologytestingsyst.
- [49] Lionel Salem. “Intermolecular Orbital Theory of the Interaction between Conjugated Systems. I. General Theory”. In: *Journal of American Chemical Society* 90.3 (1968), pp. 543–552. ISSN: 0002-7863. DOI: [10.1021/ja01005a001](https://doi.org/10.1021/ja01005a001).
- [50] Daniel V. Schroeder. *No Title*. Addison Wesley Longman, 2000, p. 422. ISBN: 0-321-27779-1.
- [51] AAPG subsurface Science. *Backscatter SEM of shales*. 2012. URL: [http://wiki.aapg.org/Scanning_electron_microscopy_\(SEM\)\#cite_ref-Huangetal_2013_2-0](http://wiki.aapg.org/Scanning_electron_microscopy_(SEM)\#cite_ref-Huangetal_2013_2-0) (visited on 05/14/2015).

- [52] Sumit Sharma and Pablo G Debenedetti. “Evaporation rate of water in hydrophobic confinement.” In: *Proceedings of the National Academy of Sciences of the United States of America* 109.12 (Mar. 2012), pp. 4365–70. ISSN: 1091-6490. DOI: [10.1073/pnas.1116167109](https://doi.org/10.1073/pnas.1116167109). URL: <http://www.ncbi.nlm.nih.gov/articlerender.fcgi?artid=3311341&tool=pmcentrez&rendertype=abstract>.
- [53] Berend Smit and Theo L M Maesen. “Molecular Simulations of Zeolites : Adsorption , Diffusion , and Shape Selectivity Molecular Simulations of Zeolites : Adsorption , Diffusion , and Shape”. In: (2008), pp. 4125–4184. DOI: [10.1021/cr8002642](https://doi.org/10.1021/cr8002642).
- [54] G. L. Squires. *Practical Physics*. Fourth edi. Cambridge University Press, 2001. ISBN: 9780521779401.
- [55] Filip Sund and Natural Sciences. “Water confined in nanoporous silica”. In: June (2014).
- [56] V. Alipour Tabrizy et al. “CO₂ Adsorption Isotherm on Modified Calcite, Quartz, and Kaolinite Surfaces: Surface Energy Analysis”. In: *Petroleum Science and Technology* 31.15 (2013), pp. 1532–1543. ISSN: 1091-6466. DOI: [10.1080/10916466.2011.586962](https://doi.org/10.1080/10916466.2011.586962). URL: <http://www.tandfonline.com/doi/abs/10.1080/10916466.2011.586962>.
- [57] H. F. W. Taylor. *Cement Chemistry*. Thomas Telford Services Ltd, 1997. ISBN: 0 7277 2592 0. URL: https://www.google.no/books?hl=no&lr=&id=1BOETtwi7mMC&oi=fnd&pg=PA1&dq=cement&ots=6YteRt0VxC&sig=1Z1GEtTL9ctWeBDqQin6hlnR8wo&redir=_esc=y#v=snippet&q=portlandite&f=false.
- [58] J. M. Thijssen. *Computational Physics*. second edi. Cambridge: Cambridge University Press, 2007, p. 613.
- [59] Abdulnour Y. Toukmaji and John a. Board. “Ewald summation techniques in perspective: a survey”. In: *Computer Physics Communications* 95.2-3 (1996), pp. 73–92. ISSN: 00104655. DOI: [10.1016/0010-4655\(96\)00016-1](https://doi.org/10.1016/0010-4655(96)00016-1).
- [60] Datchawan Unatrakarn, Asghari Koorosh, and Jose Condor. “Experimental Studies of CO₂ and CH₄ Diffusion Coefficient in Bulk Oil and Porous Media”. In: *Sciedirect* (2011). URL: http://ac.els-cdn.com/S1876610211003006/1-s2.0-S1876610211003006-main.pdf?__tid=01955ba2-f634-11e4-862e-00000aacb361&acdnat=1431166575_e95c0e32f6cff284855817af524f6b79.
- [61] a. C T Van Duin et al. “ReaxFF: A reactive force field for hydrocarbons”. In: *Journal of Physical Chemistry A* 105.41 (2001), pp. 9396–9409. ISSN: 10895639. DOI: [10.1021/jp004368u](https://doi.org/10.1021/jp004368u).

- [62] Loup Verlet. “Computer Experiments on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules”. In: 159 (1967), p. 6. URL: <http://journals.aps.org/pr/abstract/10.1103/PhysRev.159.98>.
- [63] J. Vorholz et al. “Vapor+liquid equilibrium of water, carbon dioxide, and the binary system, water+carbon dioxide, from molecular simulation”. In: *Fluid Phase Equilibria* 170 (2000), pp. 203–234. ISSN: 03783812. DOI: [10.1016/S0378-3812\(00\)00315-0](https://doi.org/10.1016/S0378-3812(00)00315-0).