

MOLECULAR DYNAMICS MODELING OF CLAY-FLUID INTERFACES

by

Jostein Blyverket

THESIS

for the degree of

MASTER OF SCIENCE



Faculty of Mathematics and Natural Sciences
University of Oslo

June 2015

Abstract

In this thesis we implement a new interatomic potential in LAMMPS. The potential is tested and validated against existing models. Promising results are found for the basic validation. Existing molecular dynamic potentials are used to model clay-fluid interfaces. Fluid migration, flow and finally elastic properties of the clay are investigated. This is done to create a framework for further testing and parametrization of our new interatomic potential. The simulation consists of modeling molecular dynamic properties of portlandite, an important constituent in concrete. Little to no studies have been conducted on this material on a molecular level.

Acknowledgements

After two years at Computational Physics I am finishing my master thesis, it has been a lot of hard work, but also a lot of fun. First I want to thank everybody at Computational Physics, I do not think there is any place at UiO like the Computational Physics group. Then I would like to thank Filip Sund, for all the help you gave me the first year of my master. Thanks to Stian Roti Svendby, for five years of collaboration, good luck with your master thesis. To Gørjan Brekke Svaland, for five years at physics, and now two years at Computational Physics, thank for all the help you have given me. To my supervisor Anders Malthe-Sørensen, thank you for always keeping the door open and for inspiring talks. Thanks to Alexandru Boțan for helping me out with LAMMPS and molecular dynamics in general. A big thanks to my father, mother and two brothers, who always support me. And finally to Martine, for being the best person I know of. I love you, and I'm looking forward to getting you back home from London.

Collaboration Details

I have worked closely with Gørjan Brekke Svaland, on setting up molecular systems. Data files and LAMMPS input script are available if you send me an email: josteibl@fys.uio.no

Contents

1	Introduction	7
1.1	Gas Bearing Shales	7
1.1.1	Shale Gas Extraction	9
1.1.2	Environmental Concerns	9
1.1.3	Ethical Aspects	10
1.2	Physics of Tight Rocks	11
1.2.1	Why Molecular Dynamics?	12
1.3	Molecular Dynamics Studies of Nanostructured Tight Rocks	12
1.4	Goals and my Contribution	14
1.5	Structure of the Thesis	15
I	Theory	17
2	Molecular Dynamics	19
2.1	Equations of Motion	19
2.2	Observables	20
2.2.1	Velocity	20
2.2.2	Energy	20
2.2.3	Temperature	21
2.2.4	Pressure	21
2.3	Ensembles	21
2.3.1	Ergodicity	22
2.3.2	Equipartition Theorem	22
2.4	Thermostats	22
2.4.1	Andersen Thermostat	23
2.4.2	Berendsen Thermostat	23
2.4.3	Nosè-Hoover Thermostat	23
2.5	Integration Scheme	24
2.5.1	Velocity Verlet	24
2.5.2	Other Integration Schemes	26
2.6	Periodic Boundary Conditions	26
2.6.1	Minimum Image Convention	27

2.7	Long Range Interactions	28
2.7.1	Ewald Summation	28
2.7.2	Other Long Range Solvers	32
3	Interatomic Potentials	35
3.1	Basic Potential	35
3.2	Clay Force Field	37
3.2.1	Lorentz-Berthelot mixing rules	38
3.3	Water Models	39
3.3.1	Other Water Models	39
3.3.2	SHAKE Algorithm	40
3.4	Stillinger-Weber Potential	41
3.5	Vashishta-Kalia-Rino-Ebbsjø Potential	42
3.5.1	Bond Order Scheme	43
3.5.2	Note on the Bond Order Scheme	44
3.6	Reax Force Field	44
4	LAMMPS	47
4.1	Installing LAMMPS	47
4.1.1	Compiling LAMMPS	48
4.2	LAMMPS Input Script	48
4.3	LAMMPS Structure	53
4.3.1	Class Hierarchy	54
4.3.2	The Potential Class	54
4.3.3	The LAMMPS Neighbor List	55
4.3.4	Pressure Calculations in LAMMPS	55
5	Hydrodynamics	57
5.1	Fluid Dynamics	57
5.1.1	Flow Between Parallel Plates, Analytic Solution	58
5.1.2	Boundary Conditions	59
5.1.3	Breakdown of Continuum	60
5.1.4	Slip Corrections	60
5.2	Gravity driven Flow	61
5.3	Pressure driven Flow	61
5.3.1	Review of Schemes	62
6	Material Properties	63
6.1	Basic Linear Elasticity Theory	63
6.1.1	Stress and Strain	64
6.1.2	Poisson's Ratio	64
6.1.3	Stiffness Matrix	65
6.1.4	Compliance Matrix	66

6.2	Fracture Mechanics	67
6.2.1	Breakdown of Continuum in Fracture Mechanics	67
II	Implementations in LAMMPS and Simulation Procedures	69
7	Implementation	71
7.1	Vashishta Potential	71
7.1.1	Two-body Potential	71
7.1.2	Three-body Potential	74
7.1.3	Bond Order Scheme	76
7.1.4	Assigning Parameters to Elements	77
7.2	Region	77
7.3	Debugging LAMMPS	78
8	Simulation Procedure	79
8.1	Crystal Structures	79
8.1.1	Portlandite	79
8.1.2	Silica	80
8.2	Constructing a Nanopore	82
8.3	Initiating Flow	84
8.4	Elasticity	85
8.4.1	Applying Strain in Molecular Dynamics	85
8.4.2	Note on Elasticity Simulations	87
III	Numerical Simulations and Results	89
9	Structural Properties	91
9.1	Radial Distribution Function	91
9.1.1	Measurements of $g(r)$	92
9.2	Density	96
9.2.1	Water and Clay Density	96
9.3	Tetrahedral Order Parameter	98
9.3.1	Bulk Water Order Parameter	98
9.3.2	Pore Order Parameter	100
10	Diffusion and Bulk Viscosity	101
10.1	Diffusion	101
10.1.1	Bulk Water Diffusion	102
10.1.2	Diffusion in Nanopores	104
10.2	Equilibrium Computations of Bulk Water Viscosity	106

10.2.1	Green-Kubo Computation of Viscosity	106
10.2.2	Equilibrium Viscosity Results	107
11	Non-Equilibrium Molecular Dynamics	109
11.1	Hydrodynamics	109
11.1.1	Gibbs Dividing Surface	110
11.1.2	Shear Viscosity	111
11.1.3	Estimates of the Shear Viscosity	111
11.1.4	Slip Length	112
11.2	Material Properties	113
11.2.1	Stress and Strain Relations	113
11.2.2	Yield Fracture	114
11.2.3	Initial Crack Fracture	116
11.2.4	Determine the Stress Intensity Factor K	117
IV	Discussion, Conclusion and Future Work	121
12	Discussion, Conclusion and Future Work	123
12.1	Discussion and Conclusion	123
12.1.1	Structural Properties	123
12.1.2	Diffusion and Viscosity	124
12.1.3	Hydrodynamics	125
12.1.4	Material Properties	126
12.2	My Contribution and Future Work	127
12.2.1	Outstanding Questions	128
12.3	Molecular Dynamics as a Method for Modeling Shales	129
V	Appendices	131
Appendix A		
SHAKE Algorithm		133
Appendix B		
Building LAMMPS on a HPC-cluster		137

List of Figures

1.1	Shale gas resources, world map	8
1.2	Shale gas extraction	9
1.3	SEM images of shale structures	11
1.4	Results from other molecular dynamics studies of shale constituents	13
2.1	PBC and minimum image convention	27
2.2	Gauss and point charge distribution	29
3.1	Lennard-Jones interatomic potential	36
3.2	Harmonic potential	38
3.3	Water model, molecular dynamics	39
3.4	SHAKE algorithm	40
3.5	Bond order scheme	43
4.1	Bulk water	53
4.2	LAMMPS class hierarchy	56
5.1	Gibbs dividing surface, principle	59
5.2	Slip boundary condition	60
6.1	Stress components	65
6.2	Fracture tip stress	68
7.1	Three-body configurations	74
8.1	Portlandite unit cell	80
8.2	b-silica unit cell	81
8.3	Annealing of silica, temperature profile	82
8.4	Setting up a nanopore	83
9.1	Radial distribution function, water models	93
9.2	Radial distribution function, silica	94
9.3	Annealing silica, system setup	95
9.4	Water density, nanopores	97
9.5	Tetrahedral network	98

2 List of Figures

9.6	Tetrahedral order parameter, three water models	99
9.7	Tetrahedral order parameter, inside nanopore	100
10.1	Bulk water mean square displacement	103
10.2	Diffusion in three nanopores	105
10.3	Green-Kubo estimates of bulk water viscosity	108
11.1	Gibbs dividing surface, 4 nm pore	110
11.2	Velocity profiles, nanopore flow	111
11.3	Slip length, 4 nm	112
11.4	Stress-strain curves, ClayFF	115
11.5	Stress-strain curves, ReaxFF	116
11.8	Stress intensity factor, ClayFF and ReaxFF	117
11.6	ClayFF fracture simulation	118
11.7	ReaxFF fracture simulation	119

List of Tables

9.1	Oxygen-Oxygen Pair Distribution Functions for All Water Models Using 1000 Molecules Simulations	92
9.2	Oxygen-Hydrogen Pair Distribution Functions for All Water Mod- els Using 1000 Molecules Simulations	92
9.3	Initial and equilibrium density of the atomic systems.	96
10.1	Bulk Water Diffusion Coeffiecient	104
10.2	Bulk Water Viscosity	108
11.1	Young's modulus and Poisson's ratio for Portlandite	114

List of Listings

1	Velocity update Verlet, equation 2.6	25
2	Position update Verlet, equation 2.7	25
3	Force update Verlet, equation 2.8	25
4	Velocity update Verlet, equation 2.9	26
5	Periodic Boundary Conditions	27
6	Minimum image convention	28
7	Two-body potential calculations	73
8	Three-body part implemented in LAMMPS, for the usc-potential	75
9	Three-body part implemented in LAMMPS, for the usc-potential	76

Chapter 1

Introduction

Shales are sedimentary rocks formed from deposits of mud, silt, clay and organic matter. Shale gas mainly consists of methane, trapped inside low permeability rocks. Methane CH₄, or natural gas, is seen as a cleaner alternative to higher emitting fuels in the transition period towards a low carbon fuel driven world. First and foremost it is seen as a replacement for coal energy, because of its low emission of greenhouse gas, compared to coal. Countries, such as China with hundreds of coal power plants are also rich on shale gas(figure 1.1). Therefore it is suggested to replace the coal extraction with natural gas. Another country with a large energy demand is the United States. In North America shale gas is abundant and they have started exporting gas instead of importing it. In year 2000 less than 1% of the gas production in the U.S came from shale gas, by year 2010 over 20 % of the gas production was shale gas. In western and Eastern Europe large reserves of unconventional gas or shale gas have been uncovered. However, European countries are not experiencing an onshore “shale gas revolution”as in the U.S, and there are several reasons for this. Clay content in European shales are higher, making fracking more difficult. European clay tends to be more ductile than its U.S counterparts. Making it bend instead of fracturing and creating new pathways for the gas. This in turn results in U.S fracking technology not being applicable to European geology. Political regulations stricter in Europe, environmental regulations were in place before shale gas extraction began, while in the U.S extraction started before regulations were set.

1.1 Gas Bearing Shales

Shales consists of fine grained clay minerals. We often hear about oil and gas-bearing shales. These shales consist of light crude oil and methane, the latter called shale gas and former called shale oil. Shales are reckoned as a low-permeability rock. Therefore shale gas and oil where not extracted on a larger

¹source: <http://matplotlib.org/basemap/>

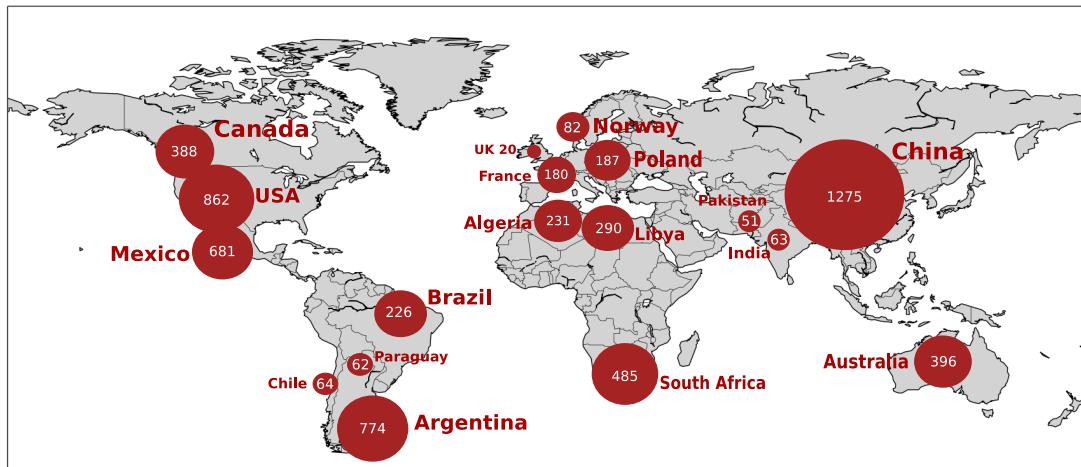


Figure 1.1: Shale gas resources by country given in trillion cubic feet ($1 \text{ f}^3 = 0.028 \text{ m}^3$). Notice that countries in the middle east and Russia are left out. Data from U.S energy administration [68]. Worldmap plotted with matplotlib¹

scale before the 1980's. Then it became profitable, because of the increasing price of fossil-energy and the technological development, such as directional drilling and fracking. The problem when it comes to extraction of oil and gas from shales is the low permeability. Conventional oil and gas production has for the last 100 years of exploration been the main focus of oil and gas companies. This is typically called free gas, trapped inside porous pockets of rocks, like sandstone, carbonates and siltstones. However most of the growth in extraction is from unconventional resources, such as tight gas, coal bed methane, gas hydrates and shale gas. These are found in a large volume, but are difficult to extract. This is in contrast to the conventional resources, which are found in small volumes, but are easy to extract.

There is a difference between tight and shale gas. The tight gas is dispersed within low porosity silt and sand areas. In the U.S tight gas is defined as having less than 10 per cent porosity and less than 0.1 millidarcy permeability [52]. Shale gas is a description of natural gas locked inside tiny pockets within layered sedimentary rocks, such as shales. A common picture is to think of it as the bubbles of air trapped in a loaf of bread as it bakes [52]. Both these forms of natural gas are extracted in a similar manner, because of the low permeability of the rock, hydraulic fracturing is applied. This increases the permeability because you induce fractures where the gas can flow. Still, even though the permeability of the rocks increases, the porosity remains the same. Because of the low permeability, new technology and knowledge is needed to extract the gas in an efficient way.

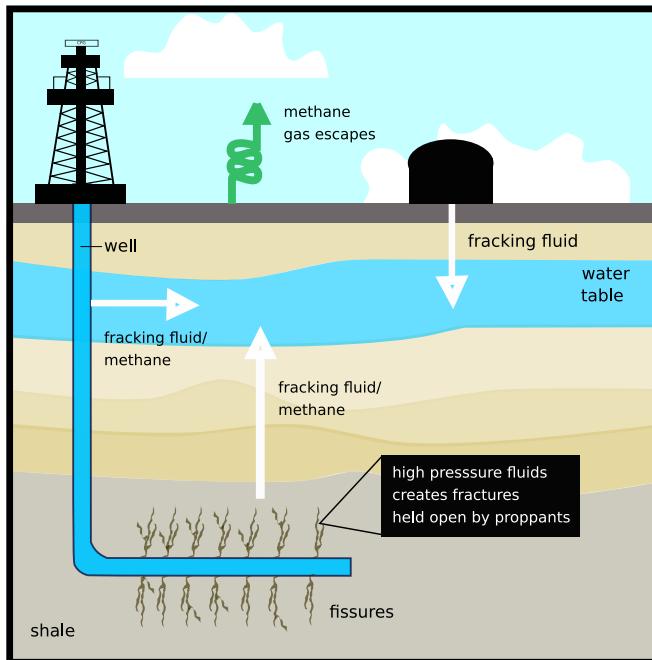


Figure 1.2: Shale gas extraction process. Possible leaking pathways for methane and fracking fluid. Created with inkscape²

1.1.1 Shale Gas Extraction

Gas bearing shales are vertically thin, but expand several kilometers horizontally. During the exploration stages vertical wells are drilled, confirming economic viability. The wells are then drilled horizontally into the shale layer, and cased with cement for completion. Explosive charges are deployed and triggered, resulting in vertical fractures along the bore-hole. Fluid consisting of water, sand (proppants) and chemicals is then injected under high pressure into the well. Opening pre-existing and creating new fractures, the proppants help keeping the fractures open. More fluid is injected to maintain the high pressure and opening fractures. When depressurized the fracturing fluid flows back to the surface and shale gas is allowed to flow into the production well, see figure 1.2.

1.1.2 Environmental Concerns

In recent years there has been increasing protest against shale gas industry and fracking in the U.S. Communities claim the groundwater is contaminated by fracking fluid and methane gas. The main concern is that fractures expand longer than the gas bearing shale, resulting in methane and fracturing fluid leakage, contaminating the groundwater. However, Fisher and Warpinski [17] found that

²source: <https://inkscape.org/en/>

fractures rarely propagates more than 200 – 400 m and that faults where not a source of unconstrained fracture propagation. This fracture length is in safe distance from the vertical distance between groundwater aquifers and shales.

These studies were conducted on U.S shales, and a different geological setting in Europe calls for similar research. In Britain environmental regulations does not permit drilling below freshwater aquifers. If the policy where to change, considerations should be given to composition of fracturing fluid, evidence of fracture height growth, hydrological conditions for fluid flow and a better understanding of shales and overlying geology. There have been concerns about stress on local water resources because large amounts of water are needed for the fracking fluid. Moore [53] reported that the amount of water needed to operate a hydraulically fractured shale gas well for a decade can be compared to the amount of water needed to water a golf court for a month, or run a 1,000 MW coal-fired power plant for 12 hours.

1.1.3 Ethical Aspects

Research has in general proven that the side effects of shale gas extraction are equal to other industries, e.g. wastewater leakage or methane leakage. The role of shale gas and climate change have been subject to numerous studies, Broderick et al. [8] concluded that shale gas only accounted for 0.29 – 2.9 % more CO₂ emissions than conventional gas production. Other studies revealed that the emissions where higher than coal [26], this study has been criticized for neglecting methane leakage by coal production.

With the development of new technology and methods to extract shale and tight gas come questions about the environmental effects. This is typical for any rapid growing area of science; are the consequences properly accounted for? In shale gas extraction horizontal wells are drilled and water is injected with a high pressure, with the purpose of fracturing and increasing the permeability of the rocks. This method has been used for several years in the conventional oil and gas industry. The new concerns are that since much of the gas bearing shales are in the vicinity of areas with high human settlements; this fracturing is in various ways polluting the groundwater. What concerns the opposition is that the contents of the fracturing liquid are often not known. This is due to competition between companies and technology kept secret to ensure advantages on other companies. The claim is that this liquid contains chemicals which contaminate the groundwater, and due to the large volume of shale gas basins this affects a lot of people. The other claim is that methane is leaking and contaminating the groundwater. There are those who think the extraction of unconventional gas is a step back for a world in desperate need for energy resources competing against fossil-energy, and in the long run replacing it. Some studies are concerned that the development of shale gas delays the initiative to invest in low-carbon

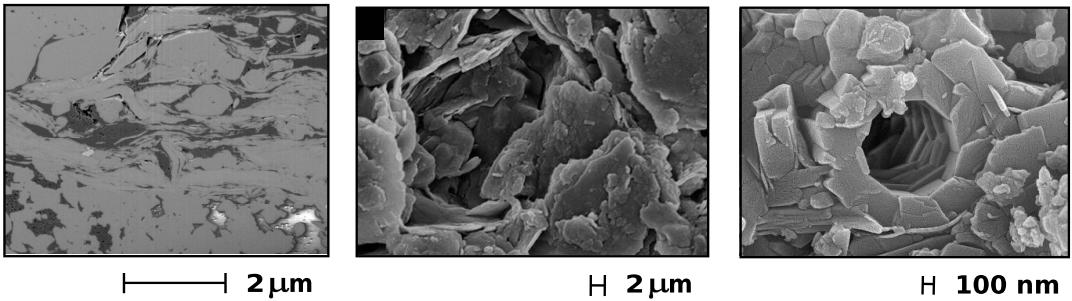


Figure 1.3: (Left) Back scattered SEM image of Barnett Shale, dark objects are Kerogen [54], (middle) SEM image of shale from China [11], (right) SEM of nanosized pore [45].

technologies and renewable energy [57, 8]. Others will say that the extraction of natural gas is replacing the more polluting coal production. After all shale gas is the most environmental friendly fossil-fuel.

1.2 Physics of Tight Rocks

Understanding how systems behave on multiscale is important in several fields of engineering. In this thesis we study water-clay systems. We could have chosen to study it on a coarser scale. For instance by solving the Navier-Stokes equation for fluid flow, or applied finite element methods to the elasticity and fracturing simulations. What we then have to realize is that we lose information of small scale processes; on the contrary from small scale processes we are not able to draw conclusions on a larger scale. If we were to use continuum mechanics, atomistic details would have been lost. Further by using molecular dynamics we lose information about the long time and length scale.

The most detailed methods are based on solving Schrödinger's equation. They are often called ab-initio methods and keep track of the wave functions for all electrons in a system. For large systems, this becomes computationally expensive. By doing approximations when solving the wave functions, density functional theory and other, alike methods are able to study a handful of atoms on your laptop. Doing more approximations e.g. by setting the atoms to be hard spheres, and setting fixed charges, we arrive at molecular dynamics. The next scale from here is coarse grained particle dynamics, such as dissipative particle dynamics and coarse grained molecular dynamics. A group of atoms is then set to behave as one, and numerical simulations can be carried out for microseconds. These methods are off-lattice methods, with a discrete nature. There is other, so called on lattice methods such as the Lattice Boltzmann method. Lattice sites are then

occupied or available, and trajectories after collisions are restricted to certain directions. The Lattice Boltzmann method is a mesoscale method, found to solve the Navier-Stokes equation. Other mesoscale methods are direct simulation Monte Carlo, where statistics control collisions. This can be used to study gas flow in the fractures seen in figure 1.2. By coupling these methods together a multiscale understanding of fluid migration in tight rocks can be created.

1.2.1 Why Molecular Dynamics?

Above we stated that shale basins stretches several kilometers horizontally and fractures from fracking expands hundreds of meters vertically. So why did we chose molecular dynamics? Well, recent studies have found that surface forces and interactions on a short scale alter the system behavior on a large scale. Eventually we can use properties found on the shorter scale as input parameters at a coarser scale, building our way up to a complete understanding of a system. Figure 1.3 illustrates shale structures at micro and nanoscale. To the left a SEM image of the Barnett shale, the center image is a closer view of shale structures, before we to the right see nanoscale structure of shales. Because of the low permeability and high surface area (due to small pores) little is known of fluid migration and transport in shales. Low permeability requires long lasting experiments, because diffusion and flow rates are relatively small. Molecular dynamics happens to be in nanometer range, making it suitable for modeling shale systems.

1.3 Molecular Dynamics Studies of Nanostructured Tight Rocks

We have chosen molecular dynamics as our method for studying shale properties. On a molecular level, shales are a mixture of a variety of minerals. Such as clay, quartz and calcite. In our study we model clay and quartz minerals.

Fluid flow and migration through small channels has become increasingly popular to study the last decade. The advent of supercomputers has aided this development. New technologies have emerged from molecular dynamics studies, such as design of nanoporous membranes for reversed osmosis [20]. Geological storing of CO₂ has in the last couple of years resulted in numerous molecular dynamics studies of clay/quartz minerals [6, 14, 43, 61, 12, 71, 5]. In these papers, properties of water confined in nanopores, swelling of clay and the possibility of wetting minerals with supercritical CO₂ is investigated. These studies are however limited, because they only consider static interatomic potentials, resulting in a lack of accuracy due to the inability to model simple chemical reactions. The Vashistha and reax force field (ReaxFF) potential have allowed for stud-

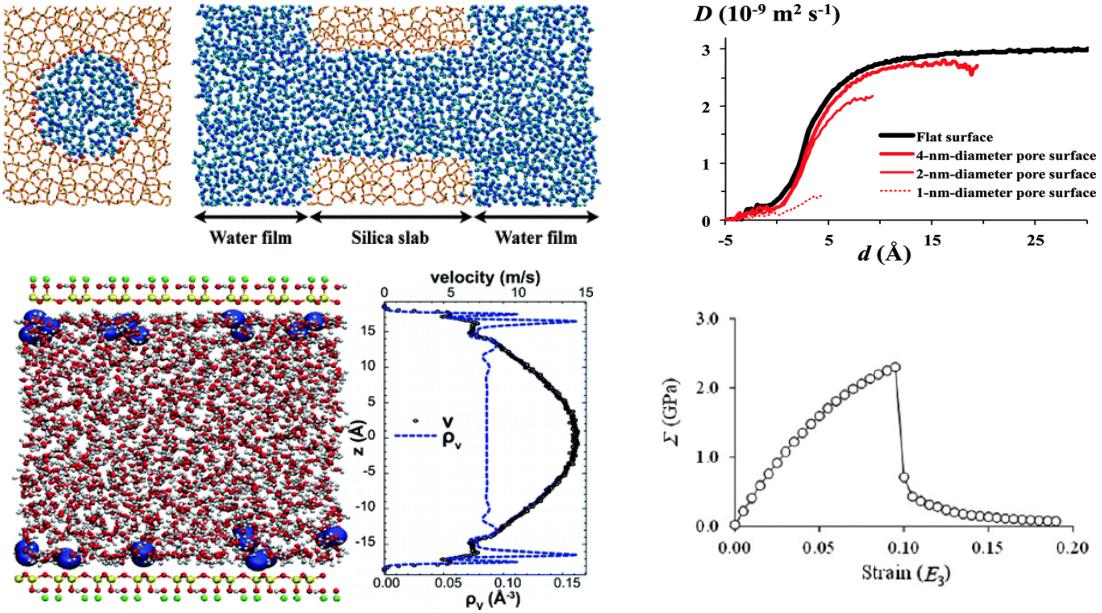


Figure 1.4: Upper left Bourg [6] studied water confined in nanopores, their diffusion results upper right. (lower left) Botan [4] found the water velocity profile. (lower right) Hantal [23] found elastic properties of Illite.

ies where interactions between water and mineral are plausible [51, 47]. The Vashistha-potential is however closed source, while the ReaxFF demands a large number of variables. In figure 1.4 results from other studies of mineral/water systems are depicted [6, 4, 23]. These results proved valuable guidelines for our own numerical simulations.

Finally, experimental methods should be mentioned. After all, making our models fit with experiments are what we strive for. Elastic properties of clay can be found by X-ray spectroscopy; however this is difficult because of the small crystal size of clays. Slip-length is investigated experimentally by two main methods, surface force apparatus (SFA) [29] and quartz-crystal microbalance [21] (QCM). In the SFA a liquid film is trapped between two mica sheets³. The film thickness can be adjusted from nanometers to Ångströms. Frictional forces are measured moving one of the surfaces. In the QCM method, liquid is absorbed on a surface and subject to vibrational motion causing liquid slip, just like marbles on a vibrating plate. Surface force apparatus measurements of slip using water on a mica surface are reporting no-slip, but for experiments with a coating on the mica surface and a water/oil liquid mixture, they observe slip in the range 5 – 80 nm [67]. It is said that experimental and a numerical treatment is vital to understand clay structures, molecular dynamics as a method has proven

³Mica is a silicate having nearly perfect basal cleavage

extremely helpful in understanding the atomistic behavior of clay minerals [13].

1.4 Goals and my Contribution

Little to no numerical research has been done on portlandite, $\text{Ca}(\text{OH})_2$. This mineral is not considered as clay; anyway all our numerical simulations setups can easily replace portlandite with another type of clay mineral. Portlandite was seen as a starting point for creating a new reactive force field, due to its simplicity compared to other clay minerals. Structure and elastic properties of portlandite are of great importance in concrete and cement industry [33]. It is known to occupy about 20 – 25% of the volume fraction of the total mass in hydrated cement. The presence of portlandite in cementitious materials influences mechanical properties, such as strength, elasticity, cracking, shrinkage and creep. $\text{Ca}(\text{OH})_2$ happens to be the calcium counterpart to brucite, $\text{Mg}(\text{OH})_2$. Brucite, has proven promising for geological carbon capture [25]. A dissolution process occurs when it comes in contact with water and dissolved CO_2 . This process is also believed to induce further fracture growth, and thereby creation of new surface area for reactions and CO_2 sequestration. Relatively little research has been undertaken on how hydraulic fracturing could affect the rate at which contaminants migrate vertically from shale formations [42]. Therefore we seek to increase the atomistic understanding of clays and fluid/clay interaction. By studying portlandite, this is an important constituent in concrete, and easily replaced by more relevant clay minerals. This is done by

- (a) Implementing a reactive force field in LAMMPS and validating it against existing models.
- (b) Study properties fluids confined in a nanopore, such as structure and diffusion.
- (c) Study fluid flow, with non-equilibrium molecular dynamics to reveal how the slip-length alters the flux through a nanopore
- (d) Crude estimations of elastic properties of clay, in which direction is the clay strongest?
- (e) Study fracture mechanics of clays. How does the choice of force field influence our results?
 - (a) The potential we implement is not previously found in LAMMPS, because the parameters are closed source. We happen to be in the position of these parameters and if they were not closed source we would have sent our implementation to the LAMMPS developers. (b) Has never been done for portlandite, and is of great interest due to weathering processes of concrete. (c) Is interesting for fluid flow

inside shales. (d) Never done for portlandite on molecular level, and again it is of great interest to know the strengths and weaknesses of cement constituents. (e) Same as (d). Note that even though portlandite is not considered clay we will continue calling it a clay mineral. For the very reason stated above, that our methods can be applied to any other clay mineral.

1.5 Structure of the Thesis

My thesis is divided into four parts. Part I concerns the theoretical background of molecular dynamics, hydrodynamics and elasticity. Part II regards the implementation of a molecular dynamics potential in LAMMPS and procedures for the various simulations. Part III is divided into chapters for the different properties we found (diffusion, viscosity etc.). Results from these simulations follow shortly after the method is presented. This is done to avoid a long chapter on methodology, before we give the results. Instead we give the methodology and the results in a consistently manner. In part IV we discuss our results and draw some conclusion, we look back to see if molecular dynamics answered our questions about modeling tight rocks. In detail the chapters are structured in the following way

- Chapter 1 Introduction and motivation.
- Chapter 2 Basic theory of molecular dynamics.
- Chapter 3 Interatomic potentials in molecular dynamics, a description of the potentials used in this thesis and their pro's and con's.
- Chapter 4 LAMMPS, the simulation tool, how to execute simulations and the basic structure of this software.
- Chapter 5 Hydrodynamics, basic concepts and derivations of equations used later in the thesis.
- Chapter 6 Material Properties, elasticity theory and basic fracture mechanics in molecular dynamics.
- Chapter 7 Implementation of an interatomic potential in LAMMPS and how to setup a simulation with this potential.
- Chapter 8 Simulation procedure, system setups, how to perform non-equilibrium molecular dynamics simulations in LAMMPS.
- Chapter 9 Structural properties of bulk water and clay systems. Validation of the implemented potential.
- Chapter 10 Bulk water and nanopore diffusion. Bulk viscosity measurements.

- Chapter 11 Non-equilibrium molecular dynamics, hydrodynamics, elasticity and fracture simulations.
- Chapter 12 Discussion and future work.

Methods on how to perform fluid flow, elasticity calculations and implementing potentials in LAMMPS are not easily available in literature. Chapter 7, 8, 9, 10 and 11 cover such methods, and hopefully others can use this thesis as a reference.

Part I

Theory

Chapter 2

Molecular Dynamics

In molecular dynamics we estimate the trajectories of particles by integrating Newton's classical laws. From particle trajectories and velocities, we sample information about an N -particle system. Using theory from statistical mechanics, properties such as energy, temperature and pressure can be quantified. Limitation in computational power sets a threshold to the number of particles N , and the total simulation time. Modern day supercomputers, such as the Anton supercomputer have been able to simulate a protein folding process, lasting in the order of milliseconds [50]. When it comes to number of particles, the Gauss Centre for Supercomputing reported a simulation of 4×10^{12} particles, which are four trillion particles [19]. For our purpose we have used a time-step of 1 fs (10^{-15} s), for at most 3 ns and a total number of particles $N = 4 \times 10^4$.

In this chapter we will cover the very basic of molecular dynamics and statistical physics. We explain how statistical physics is related to molecular dynamics, and how this make us able to characterize the mechanics of an N -body system. Some technicalities considering a molecular dynamics simulation is presented e.g. how to account for finite system sizes, and how do we keep the temperature constant? We outline how to compute the trajectories of the particles, at the end of the chapter we look into methods for handling long range forces. Interatomic potentials will be discussed in chapter 3.

2.1 Equations of Motion

In molecular dynamics we assume a hard sphere model for each of the atoms. From the potential energy we find the forces between these hard spheres. When we know the forces acting on a particle, we integrate Newton's classical equations to find the configurations of the particles at a later time step. Interatomic potentials are often written as $U(\mathbf{r})$, where \mathbf{r} is the distance between the interacting particles. The potential U comes in various forms and parametrizations. One of the most successful and widely used potentials is the, Lennard-Jones potential,

see chapter 3. By differentiating the potential energy $U(\mathbf{r})$, we find the forces acting on particle i from

$$\mathbf{F}_i = -\nabla U_i . \quad (2.1)$$

The acceleration is given as $\sum_i \mathbf{F}_i = m_i \mathbf{a}_i$, where \mathbf{a}_i is the acceleration and m_i is the mass of the object. If we integrate this equation of motion, we find the velocity \mathbf{v}_i and the position \mathbf{r}_i . We are then able to track the trajectories of all the atoms in time. When we solve equation 2.1 for N atoms we can impose restrictions to the system, e.g. constant volume and constant temperature. These restrictions are named ensembles in statistical mechanics, and will be discussed in section 2.3.

2.2 Observables

Observables are the quantities we measure in molecular dynamics. From statistical mechanics we find properties of an N-body system, by sampling individual particle properties and average it over the total number of particles. In this section we explain basic observables, from which we can find more complex quantities, such as the self-diffusion of a collection of particles. These more complex measurements are outlined in chapter 9.

2.2.1 Velocity

There is little use in knowing a single particle velocity and trying to characterize a system from this. We sample over all particles N to find a system average. The velocity \mathbf{v} of the system (if there is a flow or drift) is given by

$$\mathbf{v} = \frac{1}{N} \sum_N \mathbf{v}_i .$$

If we only measure the velocity at a time t , the motion of the atoms might seem chaotic. By averaging this quantity in time, we are able to measure if there is a flow or drift in our system.

2.2.2 Energy

A molecular dynamics system is said to have two forms of energy, kinetic and potential. These two are closely related, in the sense that we would like the total energy to be conserved. If the kinetic energy drops, the potential energy increases. The kinetic energy at a time-step t , is given as

$$E_k = \sum_N \frac{1}{2} m_i v_i^2 ,$$

where m_i is the mass of atom i , and v_i^2 is the magnitude of the velocity. The potential energy $U(\mathbf{r})$ depends on the force field chosen for the simulation, various examples are listed in chapter 3.

2.2.3 Temperature

Temperature is closely related to the kinetic energy of a system. The relationship between the temperature T and the kinetic energy E_k , is given from the equipartition theorem

$$\langle E_k \rangle = \frac{3}{2} N k_B T,$$

where E_k is the kinetic energy of all the particles, N is the number of particles, k_B is Boltzmann's constant and T is temperature. Temperature is dependent on the average kinetic energy of the particles

$$T = \frac{2\langle E \rangle_k}{3Nk_B}. \quad (2.2)$$

The above relationship is how we compute temperature in molecular dynamics.

2.2.4 Pressure

Pressure measurements in molecular dynamics are done by adding the virial term to the expression for an ideal gas. The virial relates the average total energy of a system with the total potential energy. Pressure estimates are therefore closely related to the potential energy $U(\mathbf{r})$, and then again the position \mathbf{r} of the particles. This makes it crucial to have the potential energy terms correct in order to find correct pressure values. The pressure is computed in the following way

$$P = \frac{Nk_B T}{V} + \frac{\sum_i^N \mathbf{r}_i \cdot \mathbf{F}_i}{dV}, \quad (2.3)$$

here V is the total volume of the system and d is the dimensionality of the system, 2 for 2D-simulations and 3 for 3D-simulations.

2.3 Ensembles

In molecular dynamics we talk about different ensembles, such as the NVE, NVT or NPT-ensemble. That is, constant energy, volume or pressure. Equilibrium molecular dynamic is when we keep the number of atoms N , fixed during a simulation. If we also hold the volume and energy constant throughout the simulation time, we sample from the microcanonical ensemble. By applying what is called a thermostat, we are able to control the temperature in a molecular

dynamics simulation. There are several types of thermostats and how they work is covered in section 2.4. Keeping the particle number, volume and temperature fixed is the NVT-ensemble, or canonical ensemble. Note that in a numerical simulation, the temperature does not at all seem to be constant. There are fluctuations around the mean, this works because of something named ergodicity, which will be explained in the next subsection. Holding the pressure constant instead of the volume is done in the NPT-ensemble, this is still the canonical ensemble but now the volume is allowed to fluctuate.

2.3.1 Ergodicity

We mentioned in the above section that the temperature T is not constant. If we measure the temperature at one point in our simulation, it could be 300 K and later 304 K. What the ergodicity principle is telling us, is that we can average the temperature in time. Because the configurations of the atoms at a time t , does not depend on the initial configuration at $t = 0$. There is full correspondence between the initial phase space coordinates and those that specify the system at a later time t [18]. So we can average over all initial phase space coordinates (Monte Carlo simulations) or we can average over the time evolved phase space coordinates (Molecular dynamics).

2.3.2 Equipartition Theorem

An important result in statistical physics is the equipartition theorem. It couples the kinetic energy E_k and temperature T . By adjusting the kinetic energy we are then able to control the temperature, and perform isothermal simulations. The relationship is

$$\langle E \rangle = \frac{d}{2} N k_B T ,$$

for an ideal gas with no interactions. However, this also works for particles constrained by a potential field $U(\mathbf{r})$. Here d is the degrees of freedom for a particle.

2.4 Thermostats

In molecular dynamics it is possible to adjust and control the temperature T , of a particle system. There are several ways to do this. We will shortly describe the Andersen, Berendsen and the Nosè-Hoover thermostat. They range from simplistic velocity alterations, like the Berendsen thermostat, to more sophisticated operations on the Hamiltonian ¹ of a system with the Nosè-Hoover method.

¹Hamiltonian is a representation of the energy of a system

2.4.1 Andersen Thermostat

The Andersen thermostat simulates collisions between atoms in the system and a heat bath. Colliding atoms will gain new normally distributed velocities, with a standard deviation $\sqrt{k_B T_{\text{bath}}/m}$. We generate a random number in the range $[0, 1]$ for all atoms. If this number is less than $\frac{\delta t}{\tau}$ (τ is the mean time between collisions), the atom is assigned a new velocity. There are pros and cons regarding the Andersen thermostat, since the velocities can be changed in any direction, we risk to get an unphysical behavior. E.g. two atoms on the course of colliding might have their velocities changed in a direction opposite to the collision direction, when we really just wanted a dampening factor, or rescaling of the current velocity. Because of this, the Andersen thermostat is poor at lattice vibrations, but it is quite good at shaking molecules out of position. Imagine initializing a box of water molecules in a regular structure, by using the Andersen thermostat we can shake the molecules out of this ordering.

2.4.2 Berendsen Thermostat

The Berendsen thermostat on the other hand, rescales the absolute velocities of the atoms. This removes unphysical behavior as we saw in the Andersen thermostat, because now colliding particles get their velocities rescaled in magnitude, instead of in direction. The rescaling works as follows

$$\gamma = \sqrt{1 + \frac{\Delta t}{\tau} \left(\frac{T_{\text{bath}}}{T} - 1 \right)}, \quad (2.4)$$

here Δt is the time-step, τ is the characteristic collision time, T is the temperature of the system and T_{bath} is the temperature of the heat bath. The γ -value is then multiplied by all the velocities.

2.4.3 Nosè-Hoover Thermostat

The Nosè-Hoover thermostat has a non-Hamiltonian equation of motion, designed to generate positions and velocities from an isothermal NVE and isobaric NPT-ensemble. The fact that the equations of motion are non-Hamiltonian, is a result from the alteration of the Hamiltonian of the system. The new equations of motion is written as

$$\mathbf{F}_i = -\mathbf{F}_i - \zeta m_i \mathbf{v}_i \quad (2.5)$$

Where ζ is a friction coefficient. Since the equations of motion are no longer on the form of 2.1, the integration scheme also changes.

The Nosè-Hoover thermostat differs from the Andersen and Berendsen that it alters the Hamiltonian of the system. Hoover found that this can only be done on the above form to sample the canonical ensemble correct. One artifact

considering the above equation is that it cannot be derived from a Hamiltonian. The implication of this is that we cannot use statistical mechanics as before to find dynamic properties of the system. This becomes a problem when we apply an external force field \mathbf{F} , as we do for simulation of flow in the nanopores. According to Hoover, we are then unable to sample the canonical ensemble correct. Martyna et al. [38], found a solution to this problem in 1992; they introduced chains of thermostats allowing for a system with more than one conservation law.

2.5 Integration Scheme

In molecular dynamics, we integrate Newton's equations of motion. To do this, we need an integration scheme, preferably a scheme that conserves the energy of the system. The velocity Verlet algorithm is designed to do this, as well as being relatively simple. It is often considered the best algorithm for molecular dynamics simulations. We might be tempted to use the well-known, simple Euler algorithm, but it turns out that this suffers from large energy drift due to the over-estimation of the solution. It is therefore particularly bad for oscillatory movement, which is bad if we want to simulate solids in molecular dynamics. We present the velocity Verlet algorithm and a C++ implementation in the next section.

2.5.1 Velocity Verlet

Here we state the scheme for the integration of the equations of motion in LAMMPS. Our own implementation of velocity Verlet algorithm is used to clarify the integration steps. The scheme is derived from a Taylor expansion of the equations of motion, and can be found in [18] chapter 4.2. For each atom i the steps are as follows

$$\mathbf{v}_i(t + \Delta t/2) = \mathbf{v}_i(t) + \frac{\mathbf{F}_i(t)}{2m} \Delta t \quad (2.6)$$

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

$$\mathbf{F}_i(t + \Delta t) = -\nabla_i U_i(\mathbf{r}(t + \Delta t)) \quad (2.8)$$

$$\mathbf{v}_i(t + \Delta t) = \mathbf{v}_i(t + \Delta t/2) + \frac{\mathbf{F}_i(t + \Delta t)}{2m} \Delta t \quad (2.9)$$

An implementation of the scheme can be stated in the following way. Starting with the velocity update in 2.6.

```
for (uint dim = 0; dim < 3; dim++) {
    velocity[dim] = atom.getVelocity()[dim] + (atom.getForce()[dim]/2.0)*dt;
}
atom.setVelocity(velocity);
```

Listing 1: Velocity update Verlet, equation 2.6

A new velocity for all three dimensions are found from the old velocity and the forces experienced in the previous time step. Next step is to update the positions from this new velocity, equation 2.7

```
for (uint dim = 0; dim < 3; dim++) {
    atom.m_position[dim] += atom.getVelocity()[dim]*dt;
}
```

Listing 2: Position update Verlet, equation 2.7

Now we arrive at the time-consuming part in molecular dynamics, that is, calculating the forces. Step 2.8 in the velocity Verlet algorithm is be implemented as

```
forceFactor = 24*(2*R12i - R6i)*R2i;

for (uint dim = 0; dim < 3; dim++) {
    force[dim] = forceFactor*Rij[dim];
}

atoms[i].addForce(force);

// Newtons 3'law
for (uint dim = 0; dim < 3; dim++) {
    force[dim] = -force[dim];
}
atoms[j].addForce(force);
```

Listing 3: Force update Verlet, equation 2.8

In this listing i and j are the $i'th$ and the $j'th$ atoms in the system. At the end we update the velocities from the velocity obtained in 2.6 and the force found in 2.8.

```

for (uint dim = 0; dim < 3; dim++) {
    velocity[dim] = atom.getVelocity()[dim] + (atom.getForce()[dim]/2.0)*dt;
}
atom.setVelocity(velocity);

```

Listing 4: Velocity update Verlet, equation 2.9

The above implementation has been tested in my own molecular dynamics code for a LJ-potential. It turns out the velocity Verlet algorithm is not only simple; it is also one of the most accurate algorithms. In the next section we give a brief outline of other integrators.

2.5.2 Other Integration Schemes

There are other methods capable of obtaining the same precision as the velocity Verlet algorithm. The Leap Frog method is derived from the original Verlet scheme and therefore computes the same trajectories. The disadvantage with this method is that the velocity and position are not known at the same time, which gives us little control over the energy of the system. Some methods compute the same trajectories as the velocity Verlet method but give a better estimate of the velocities. Like the Beeman algorithm, where a more accurate velocity computation provides a better calculation of the energy. For completeness, we mention the higher-order schemes which are based on a higher-order derivative of the particle coordinates. But even though these would allow for a longer time step without loss of accuracy, they are computational expensive [18].

2.6 Periodic Boundary Conditions

In continuum fluid dynamics, boundary conditions are handled by setting the velocity at the boundary equal to zero. If we solve the wave equation, we apply Dirichlet or Neumann boundary conditions. Because of the discrete nature of molecular dynamics, the effects of a finite system size are handled in an alternative way, compared to the continuum methods given above. When the particles are at an edge of the simulation box, and passes through the system, we update the position by subtracting/adding a system size length L to the position. According to what boundary the particle crossed, if it was $x = 0$, we add a system length L_x . This is called periodic boundary conditions, and numerically we write

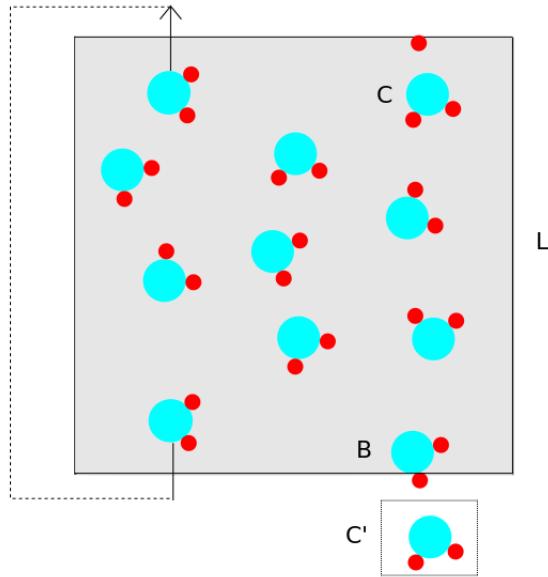


Figure 2.1: Periodic boundary condition and minimum image convention illustrated.

```
for (uint dim = 0; dim < 3; dim++) {
    if (atom.getPosition()[dim] > L[dim]) atom.m_position[dim] -= L[dim];
    else if (atom.getPosition()[dim] < 0) atom.m_position[dim] += L[dim];
}
```

Listing 5: Periodic Boundary Conditions

Here `dim`, is the dimension of our system.

2.6.1 Minimum Image Convention

Now that we have imposed periodic boundary conditions on the particles, we have to make sure that there is no *empty* space outside the simulation box. This is handled by the minimum image criteria, and is best explained by looking at figure 2.1. Atom *B* is at the edge of the simulation box, it does not pass through the box because it sees an image of atom *C* on the other side of box, *C'*. This keeps it from escaping the box, because it allows for interactions across the system. One condition has to be satisfied though, the cutoff r_c needs to be smaller than half the system size L . The shadow interaction between atom *B* and *C* can be implemented as

```

for (uint dim = 0; dim < 3; dim++) {
    Rij[dim] = R[i][dim] - R[j][dim];

    if (Rij[dim] < -L[dim]/2.0)      Rij[dim] += L[dim];
    else if (Rij[dim] > L[dim]/2.0) Rij[dim] -= L[dim];
}

```

Listing 6: Minimum image convention

In listing 6, we compute the interatomic distance r_{ij} and test if the distance is less than $L/2$ or if it is larger than $L/2$. According to this, we add or subtract a system size of L , respectively. Eventually this solves the problem of having a surface on each edge of the system box, and therefore only a finite system size. Care has, to be taken when doing simulations where an object in the simulation box might interact with its periodic image.

2.7 Long Range Interactions

Long range forces in molecular dynamics include Van der Waals and Coulomb interactions. These are non-bonded forces which make them major contributors to the dynamics of particles in a molecular dynamics simulation. If we take a look at the Coulomb potential, we find that the energy decays as $\frac{1}{r}$, compared to other potentials such as the LJ which goes as $\frac{1}{r^6}$. So the contribution is immense even at a long range r . The cutoff introduced for the LJ-potential is surely too short for the Coloumb term, and we would get an unphysical jump in the energy at the cutoff distance r_{cut} . To handle long range forces several screened potentials have been presented, such as the Yukawa and Debye. These potentials are less computational expensive, but fails in accuracy over the long range solutions of Ewald summation methods.

The most used of these long range solvers is the Ewald summation. It was introduced in 1921 by Paul Peter Ewald [16]. Since the Ewald summation is so important for the stability and structure of a molecular dynamics system, we will show in detail how it works.

2.7.1 Ewald Summation

A set of point charges in molecular dynamics contributes to the potential as

$$U(r) = \sum_{i=1}^N q_i \phi(r_i) \tag{2.10}$$

here q_i is the point charge value, and ϕ is the electrostatic potential of atom i . The electrostatic potential of point i , because of all the other particles j , can

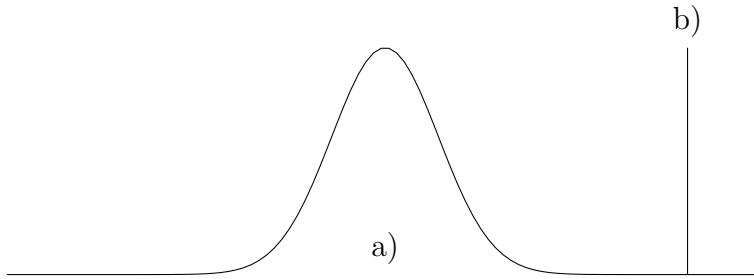


Figure 2.2: (a) Gauss charge density. (b) Point charge

then be expressed as

$$\phi(r_i) = \sum'_{j,\mathbf{n}} \frac{q_j}{|\mathbf{r}_{ij} + \mathbf{n}L|}. \quad (2.11)$$

In equation 2.11 we sum over all the particles j and their periodic images \mathbf{n} . Since the particles are point charges, this is a poorly convergent sum of δ -functions. To continue on from here we make some assumptions

1. We assume that i interact with all its periodic images, but not with itself.
2. Around charge q_i there is a charge cloud, because of charges with opposite sign $-q_i$.
3. This *cloud* is represented as a Gaussian function, which rapidly decays to 0 when r is large.

These assumptions reveals that we have to compute three contributions to the potential, 1. Because of the point charge q_i , 2. Because of the screened cloud charge and 3. Because of the background charges. Now we will address the three different contributions.

We assume that the screened charge contribution takes the form

$$\rho_{\text{Gauss}}(r) = -q_i(\alpha/\pi)^{3/2} \exp(-\alpha r^2) \quad (2.12)$$

at a point r in space, where α is a constant. Comparing the δ and Gaussian charge distribution is done in figure 2.2.

Poisson's equation and Fourier Transform

Poisson's equation gives us the potential contribution of a charge distribution $\rho(r)$. We can write this as

$$-\nabla^2\phi(\mathbf{r}) = 4\pi\rho(\mathbf{r}), \quad (2.13)$$

in Gauss units. The simplest solution of this equation is the single point charge situated in the origin

$$\phi(\mathbf{r}) = \frac{z}{4\pi|\mathbf{r}|}. \quad (2.14)$$

We have a whole set of point charges distributed as

$$\rho(\mathbf{r}) = \sum_{i=1}^N q_i \delta(\mathbf{r} - \mathbf{r}_i). \quad (2.15)$$

Just a sum of δ -functions spread out in space. The potential at a random point in space follows

$$\phi(\mathbf{r}) = \sum_{i=1}^N \frac{q_i}{4\pi|(\mathbf{r} - \mathbf{r}_i)|}. \quad (2.16)$$

Like any real function we can write this in Fourier space using the relation

$$f(\mathbf{r}) = \frac{1}{V} \sum_{-\infty}^{\infty} \tilde{f}(\mathbf{k}) e^{i\mathbf{k}\cdot\mathbf{r}}, \quad (2.17)$$

V is the volume of the system, $\mathbf{k} = 2\pi/\mathbf{l}$ where $\mathbf{l} = \{l_x, l_y, l_z\}$. The Fourier coefficients is found by

$$\tilde{f}(\mathbf{k}) = \int_V d\mathbf{r} f(\mathbf{r}) e^{-i\mathbf{k}\cdot\mathbf{r}}. \quad (2.18)$$

Let us apply this trick to the Poisson's equation and write it in Fourier space.

$$-\nabla^2 \phi(\mathbf{r}) = -\nabla^2 \left(\frac{1}{V} \sum_k \tilde{\phi}(\mathbf{k}) e^{i\mathbf{k}\cdot\mathbf{r}} \right) \quad (2.19)$$

$$= \frac{1}{V} \left(\sum_{\mathbf{k}} k^2 \tilde{\phi}(\mathbf{k}) e^{i\mathbf{r}\cdot\mathbf{k}} \right) \quad (2.20)$$

If we then take the Fourier transform of the charge distribution

$$\rho(\mathbf{r}) = \frac{1}{V} \sum_k \tilde{\rho}(\mathbf{k}) e^{i\mathbf{r}\cdot\mathbf{k}} \quad (2.21)$$

Now we can substitute equation 2.19 and 2.21 into equation 2.13, and get

$$k^2 \tilde{\phi}(k) = 4\pi \tilde{\rho}(k), \quad (2.22)$$

hence the name *kspace* solver in LAMMPS. Let solve this for our cloud charge distribution instead of the δ -distribution.

Solving Poisson's equation with Fourier series

We can now, as mentioned above solve the charge distribution assumed present in our system. This is done with a Fourier form of the Poisson's equation 2.12. The potential at r_i due to the charge distribution $\rho_i(\mathbf{r})$ is given as

$$\rho(\mathbf{r}) = \sum_{j=1}^N \sum_{\mathbf{n}} q_j (\alpha/\pi)^{3/2} \exp[-\alpha|\mathbf{r} - (\mathbf{r}_j + \mathbf{n}L)|^2]. \quad (2.23)$$

To use the Poisson's equation on Fourier form we Fourier transform the above equation

$$\rho_i(\mathbf{k}) = \int_V d\mathbf{r} e^{-i\mathbf{k}\cdot\mathbf{r}} \rho(\mathbf{r}) \quad (2.24)$$

$$= \int_V d\mathbf{r} \exp(-i\mathbf{k}\cdot\mathbf{r}) \sum_{j=1}^N \sum_{\mathbf{n}} q_j (\alpha/\pi)^{3/2} \exp[-\alpha|\mathbf{r} - (\mathbf{r}_j - \mathbf{n}L)|^2] \quad (2.25)$$

$$= \int_{\text{all space}} d\mathbf{r} \exp(-i\mathbf{k}\cdot\mathbf{r}) \sum_{j=1}^N q_j (\alpha/\pi)^{3/2} \exp[-\alpha|\mathbf{r} - \mathbf{r}_j|^2] \quad (2.26)$$

$$= \sum_{j=1}^N q_j \exp(-i\mathbf{k}\cdot\mathbf{r}_j) \exp(-k^2/4\alpha). \quad (2.27)$$

This can be inserted in the Poisson's equation on Fourier form

$$\tilde{\phi}(\mathbf{k}) = \frac{4\pi}{k^2} \sum_{j=1}^N q_j \exp(-ik\cdot\mathbf{r}_j) \exp(-k^2/4\alpha). \quad (2.28)$$

Now we can compute the contribution to the potential U from ϕ , but first we find $\phi(\mathbf{r})$.

$$\phi(\mathbf{r}) = \frac{1}{V} \sum_{\mathbf{k}\neq 0} \tilde{\phi}(\mathbf{k}) \exp(i\mathbf{k}\cdot\mathbf{r}) \quad (2.29)$$

$$= \sum_{\mathbf{k}\neq 0} \sum_{j=1}^N \frac{4\pi q_j}{k^2} \exp[i\mathbf{k}(\mathbf{r} - \mathbf{r}_j)] \exp(-k^2/4\alpha) \quad (2.30)$$

and hence

$$U = \frac{1}{2} \sum_i q_i \phi(\mathbf{r}) \quad (2.31)$$

$$= \sum_i \sum_{j=1}^N \frac{4\pi q_i q_j}{V k^2} \exp[i\mathbf{k}\cdot(\mathbf{r}_i - \mathbf{r}_j)] \exp(-k^2/4\alpha) \quad (2.32)$$

$$= \frac{1}{2V} \sum_{\mathbf{k}\neq 0} \frac{4\pi}{k^2} |\tilde{\rho}(\mathbf{k})|^2 \exp(-k^2/4\alpha) \quad (2.33)$$

where we have used the definition

$$\tilde{\rho}(\mathbf{k}) \equiv \sum_{i=1}^N q_i \exp(i\mathbf{k} \cdot \mathbf{r}_i).$$

Correcting the self Contribution and Real Space sum

In the above contribution to the potential U we have included a term due to the cloud charge of q_i and the point charge q_i located at the center of the Gaussian 2.2. To correct for this we find the potential energy at the center of the Gaussian as

$$\rho_{\text{Gauss}}(\mathbf{r}) = q_i (\alpha/\pi)^{3/2} \exp(-\alpha r^2). \quad (2.34)$$

Using Poisson's equation and real space this time we find

$$U_{\text{self}} = (\alpha/\pi)^{1/2} \sum_{i=1}^N q_i^2. \quad (2.35)$$

This should be subtracted from the real space and Fourier contributions to the Coulomb energy. The real space sum is the point charges screened by opposite point charged Gaussians and give us the term

$$U_{\text{short-range}} = \frac{1}{2} \sum_{i \neq j}^N q_i q_j (\sqrt{\alpha} r_{ij}) / r_{ij}. \quad (2.36)$$

So the total contribution to the potential energy for the electrostatics can now be written

$$U = \frac{1}{2V} \sum_{\mathbf{k} \neq 0} \frac{4\pi}{k^2} |\tilde{\rho}(\mathbf{k})|^2 \exp(-k^2/4\alpha) \quad (2.37)$$

$$- (\alpha/\pi)^{1/2} \sum_{i=1}^N q_i^2 \quad (2.38)$$

$$+ \frac{1}{2} \sum_{i \neq j}^N q_i q_j (\sqrt{\alpha} r_{ij}) / r_{ij}, \quad (2.39)$$

we observe that the mid term is not dependent of r and as long as the charges are constant so is this value.

2.7.2 Other Long Range Solvers

The Ewald method sketched above scales as $\mathcal{O}(N^{3/2})$. To overcome this scaling, other methods, such as the particle-particle-mesh (PPPM) method and the

fast-multiple method were developed. They scales as $\mathcal{O}(N \log N)$ and $\mathcal{O}(N)$, respectively. In our simulations we used the PPPM-method, it maps the atoms to a 3D mesh and exerts a fast Fourier transform to solve Poisson's equation on the mesh, then interpolates these values back to the real system. The derivation of these mesh-schemes is beyond the scope of this thesis.

Chapter 3

Interatomic Potentials

Atomic potentials have been developed to study a wide range of materials, and to evaluate their microscopic structure, physical and thermodynamical properties. Clay introduce a daunting challenge for the computational physicist due to the complexity of clay minerals, and the experimentalist faces challenges due to the ultra-fine grain size of these phases [13]. The clay force field (ClayFF) was derived from a combination of quantum chemical calculations and empirical methods. This allowed the authors to conserve important physical quantities that other models were lacking, like swelling of clay. This force field is based on an ionic description of the metal-oxygen interaction, preserving the swelling attribute. Long range molecular dynamic forces, as van der Waals and Coulomb interactions are found for interfaces between the clay and fluid phases. Most of our simulations are performed using the ClayFF.

We look at different types of water models, the SPC, SPC/E and usc-model(Vashistha-potential). SPC, is an abbreviation for single point charge, referring to the point charge located on each atom. Even though water is a simple molecule, consisting of only three atoms, molecular simulations of water are not trivial. This is because of how the long range forces, as electrostatics affects water and its dynamic properties. We give a brief outline of how many-body potentials are constructed, and the Vashistha-potential is presented, so is the belonging bond-order scheme.

3.1 Basic Potential

The most widely used potential form in molecular dynamics, is the Lennard-Jones (LJ) potential. As we will see later, its applicability ranges from modeling noble gases to highly complex clay constituents. The LJ-potential was developed in the 1920's by, J.E. Jones, and the object of the study was to explain experimental data with the equation of state of a gas [30]. Some of this data had been explained by van der Waals models, Jones extended this further. Today the repulsive and attractive form is written as one, and called the LJ-potential. It is a two-body

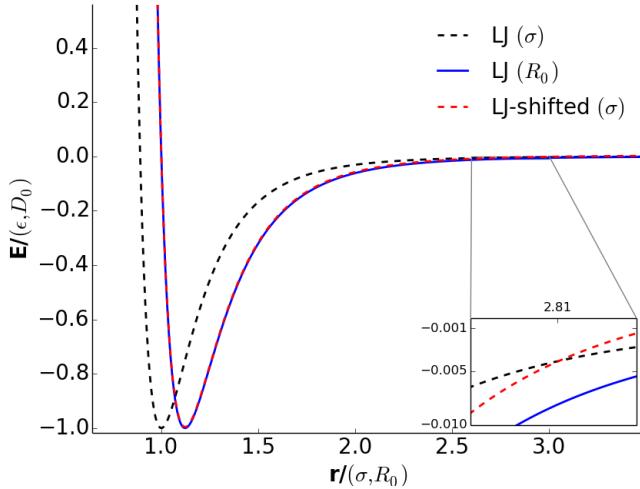


Figure 3.1: Comparing the two potential forms in equation 3.1, and the shifted LJ-potential. The shifted potential has the effect of shortening the range of the potential. As we see from the zoomed figure, it decreases faster towards zero before we set it to zero at the cutoff length.

potential, meaning that it gives you the energy between two atoms i and j as a function of their separation, r . The functional form is

$$E_{\text{LJ}} = 4\epsilon \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 \right] = D \left[\left(\frac{R_0}{r} \right)^{12} - 2 \left(\frac{R_0}{r} \right)^6 \right], \quad (3.1)$$

where ϵ is the strength of the interaction, σ and R is the point of the lowest potential value. The relation between these two potential forms is found in chapter 3.2. As seen from figure 3.1, the potential does not go to zero in the vicinity of the cutoff value r_c . The cutoff is best described

$$E(r) = \begin{cases} V(r), & r \leq r_c \\ 0, & r > r_c. \end{cases}$$

This sudden truncation of the potential can cause unphysical effects, such as a jump in energy for atoms moving past the r_c -distance. Often the LJ-potential is shifted at r_{cut} , to avoid these errors, the potential is then on the form

$$E(r) = \begin{cases} V(r) - V(r_c), & r \leq r_c \\ 0, & r > r_c. \end{cases}$$

The potential is evaluated at the cutoff distance r_c (often 3σ), and we subtract this value from the potential value at a distance r , see figure 3.1.

3.2 Clay Force Field

The clay force field was developed to represent crystalline compounds and its interfaces with fluids. It is a general force field, in the sense that it models multiple clays very well. The atoms are point particles, with a point charge and are allowed full translational movement. Interactions between metal and oxygen are based on the LJ-potential, and electrostatics on the Coulomb form. The force field parameters are found from a combination of DFT¹ quantum mechanical computations, and some of the structure is found from experiments [13]. To model the OH-bond stretch and angle between H-O-H, two harmonic terms were added. In the following section we present the force field, its parameters can be found in the appendices.

The total energy of the atomic system is written on the form

$$E_{\text{total}} = E_{\text{coul}} + E_{\text{vdW}} + E_{\text{bond stretch}} + E_{\text{angle bend}} \quad (3.4)$$

The three first terms in 3.4 are two-body interactions, while the last term is a three-body evaluation. The point charge interaction E_{coul} is the well-known Coulomb potential for point charges

$$E_{\text{coul}} = \frac{e^2}{4\pi\epsilon_0} \sum_{i \neq j} \frac{q_i q_j}{r_{i,j}}, \quad (3.5)$$

where e is the charge of an electron, q is the partial charge derived from quantum mechanics, r_{ij} is the vertex from atom i to atom j and ϵ_0 is the dielectric constant. The covalent bond between oxygen and hydrogen is modeled as a harmonic potential, the strength of the potential depends on the atomic distance r_{ij} , from the equilibrium distance r_0

$$E_{\text{bond stretch}} = \frac{1}{2} k_{\text{bond}} (r_{ij} - r_0)^2. \quad (3.6)$$

Here k , is the strength of the harmonic oscillator. In order to model the H-O-H angle in water and metal-oxygen angle in clays, we define the three-body potential

$$E_{\text{angle bend}} = \frac{1}{2} k_{\text{angle}} (\theta_{ijk} - \theta_0)^2. \quad (3.7)$$

In equation 3.7, the angle bend is a harmonic oscillator around the equilibrium angle θ_0 . The strength of the potential is controlled by the prefactor $k_{\text{angle bend}}$.

Two-body repulsion and attraction are modeled by the LJ, $\frac{1}{r^{12}}$ and, $-\frac{1}{r^6}$ term, respectively.

$$E_{\text{vdW}} = \sum_{i \neq j} D_{ij} \left[\underbrace{\left(\frac{R_0}{r_{ij}} \right)^{12}}_{\text{Repulsion}} - 2 \underbrace{\left(\frac{R_0}{r_{ij}} \right)^6}_{\text{Attraction}} \right]. \quad (3.8)$$

¹Density functional theory, described in the introduction

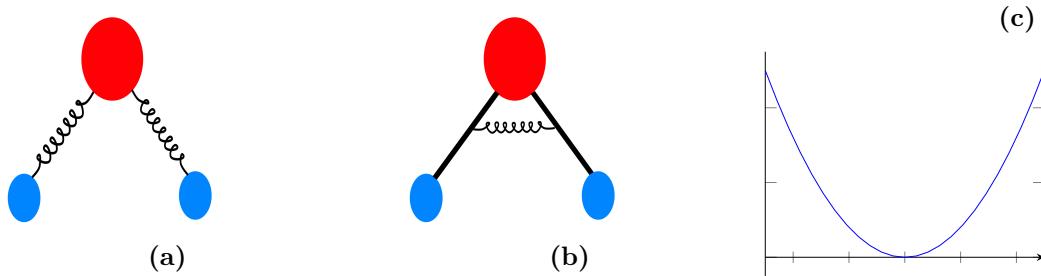


Figure 3.2: (a) Harmonic oscillator potential for covalent oxygen-hydrogen bonds. (b) Harmonic potential for the angle bend (c) potential form of the harmonic potentials.

We find how R_0 is related to σ . We know that R_0 describes where the potential has a minimum, so if we differentiate the left side of equation 3.1, we get

$$F = -4\epsilon_{ij} \left[12 \left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - 6 \left(\frac{\sigma_{ij}}{r_{ij}} \right)^6 \right] \frac{r_i}{r^2}. \quad (3.9)$$

After some rearranging

$$\sigma = \frac{R_0}{2^{1/6}}. \quad (3.10)$$

Now we have an expression for how, R_0 and σ are related in the two potential forms of the LJ-potential.

3.2.1 Lorentz-Berthelot mixing rules

Interactions between molecules or atoms of different types need to be modeled in an efficient way. In the clay force field, and among other LJ-based molecular dynamics potentials, we use a mixing of parameters between atom i of type 1, and atom j of type 2. This mixing is called Lorentz-Berthelot mixing rule. The distance parameter σ_{ij} is calculated as an arithmetic mean

$$\sigma_{ij} = \frac{1}{2}(\sigma_i + \sigma_j). \quad (3.11)$$

The evaluation of the energy term is given as a geometric mean

$$\epsilon_{ij} = \sqrt{\epsilon_i \epsilon_j}. \quad (3.12)$$

These mixing rules assures us that we do not need to find parameters for interactions between all particles explicitly, i.e. from quantum mechanical calculations of the interaction between i and j . We are then able to simulate complex and ill-defined structures without running the risk of over parametrization, slowing the computations down. These mixing methods are not without faults [3], but as mentioned they are effective.

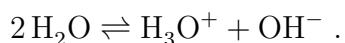


Figure 3.3: Illustration of a water molecule in molecular dynamics. The red sphere is the oxygen with its point charge and the two white spheres are hydrogens. They are in a distance $r = 1.0 \text{ \AA}$ from the oxygen atom. The angle spanned by the two vertices' (covalent bonds) is $\theta = 109.47^\circ$

3.3 Water Models

Unfortunately, properties of water is hard to study in molecular dynamics. This is due to the influence of long-range electrostatics. There are different models, each with their pros and cons, some model diffusion quite well, but are not that good at water-solid interfaces. Again we have to decide what properties are important for our simulation and our results. The potential forms for water that will be used here are the SPC, SPC/E and usc-water model. In the SPC and SPC/E model, the bonds and angles are constrained with the SHAKE algorithm in LAMMPS. Charge values are the only thing that differs between SPC and SPC/E, this will have an effect on the dynamics of the water.

The covalent bonds between oxygen-hydrogen are static in the SPC and SPC/E-model, but in the usc-model we can get bond breaking and dissociation of water according to



We are then able to model some simple chemistry, e.g. the forming of silanol groups Si-O-H, at the surface of silica-water interfaces. The excess negative charges forming on the oxygen atom, as seen from figure 3.3, make hydrogen bonding between molecules plausible in molecular dynamics. Hydrogen bonds are crucial to determine properties of water, because strong hydrogen bonding gives a more tetrahedral structure and lower diffusion coefficient.

3.3.1 Other Water Models

There are a wide variety of water models, some of the most well-known are the TIP-models. They introduce a dummy atom in between the triangle formed by the hydrogen and oxygen atoms. In the TIPS2-model, the charge q is moved of the oxygen atom and towards the two hydrogen atoms [31]. The TIP4P-model,

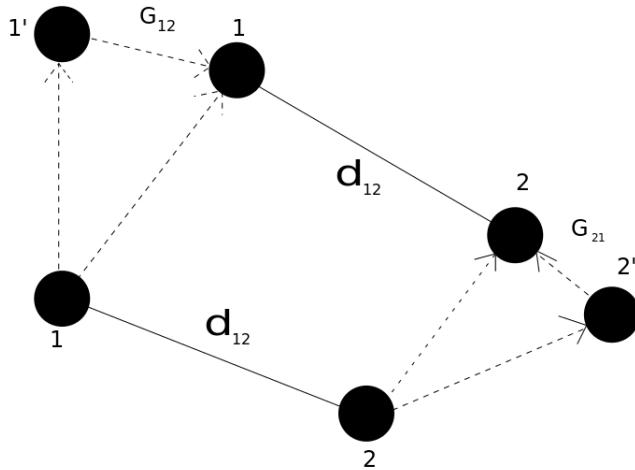


Figure 3.4: Illustration of the SHAKE algorithm. Atom 1 moves to the position $1'$ and the bond distance d_{12} is not conserved. This is handled by the constraint and the SHAKE algorithm which gives us the force G_{12} needed for bond length conservation.

has three fixed point charges q , and one LJ center displaced towards the hydrogen atoms [1]. Five site models have also been developed, TIP5P. Here there is no charge on the oxygen atom; the two hydrogen atoms are given a positive charge. The two lone pair atoms are given the opposite charge of the hydrogens. The LJ interaction is only operating between pair of oxygen atoms [35].

3.3.2 SHAKE Algorithm

Due to the flexibility of the oxygen-hydrogen bond in the SPC and SPC/E-models, the time step is forced to be low. If the time step is too large, we risk getting a system with non-constant energy or lose hydrogen from an oxygen atom, because it has moved too far in one time step. One way to handle this flexibility is to set the distance d_{12} in figure 3.4 as a constant. But then we are neglecting the force required to keep d_{12} constant, this is bad for energy conservation in the system. These problems are handled with the SHAKE algorithm. This algorithm was proposed as early as 1976, and was used to simulate liquid butan [49]. We give a brief outline of the method.

Consider two atoms 1 and 2 at a distance d_{ij} apart. If we apply a holonomic² constraint to the motion of the particles, we need to find the force G_{ij} to hold d_{ij} constant. This procedure is shown in the appendix V. A brief outline of the

²A particle on a sphere is subjected to a holonomic constraint unless there is a force e.g gravity which can make it fall off the sphere.

algorithm as seen in figure 3.4. It consists of moving the atoms to the positions $1'$ and $2'$ due to the unconstrained forces. Then find the constrained forces G_{ij} so that the bond distance d_{ij} is maintained constant.

3.4 Stillinger-Weber Potential

The Stillinger-Weber potential is briefly discussed due to its similarities with the usc-potential. When proposed by Frank Stillinger and Thomas Weber in 1984, it was meant to describe solid and liquid forms of Si. It differs from other potential forms, with the fact that it models the geometry of the system and only consists of a reduced pair potential. This make the transferability to other systems than the potential originally was designed for non-trivial. The form of the potential as implemented in LAMMPS is

$$E = \sum_i \sum_{j>i} \phi_2(r_{ij}) + \sum_i \sum_{j\neq i} \sum_{k>j} \phi_3(r_{ij}, r_{ik}, \theta_{ijk}) . \quad (3.13)$$

Where the two-body part takes the form

$$\phi_2(r_{ij}) = A_{ij}\epsilon_{ij} \left[B_{ij} \left(\frac{\sigma_{ij}}{r_{ij}} \right)^{p_{ij}} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^{q_{ij}} \right] \exp \left(\frac{\sigma_{ij}}{r_{ij} - a_{ij}\sigma_{ij}} \right) . \quad (3.14)$$

The parameters A , B , p , q and a where in the original paper optimized so that a diamond structure where the most stable structure at low pressure. Later they were further optimized to fit the melting point and liquid structure of silica. The three-body part is on the form

$$\begin{aligned} \phi_3(r_{ij}, r_{ik}, \theta_{ijk}) &= \lambda_{ijk}\epsilon_{ijk} [\cos(\theta_{ijk}) - \cos(\theta_0)]^2 \times \\ &\exp \left(\frac{\gamma_{ij}\sigma_{ij}}{r_{ij} - a_{ij}\sigma_{ij}} \right) \exp \left(\frac{\gamma_{ik}\sigma_{ik}}{r_{ik} - a_{ik}\sigma_{ik}} \right) . \end{aligned} \quad (3.15)$$

Again the λ and the γ parameters where optimized to the above mentioned criteria. The two parameters σ and ϵ are the only two variables with a dimension, distance and energy, respectively. The other variables are dimensionless.

The SW-potential was developed to study semiconductors, later it has been parametrized for coarse grained water in the mW-model. Because the molecular structure of water and silica are much alike, they both prefer a tetrahedral coordination of atoms. And it was found to model properties of water quit well [40]. Here the tetrahedral structure is being favored by the three-body term that penalizes non-tetrahedral angles. There are no electrostatic forces involved, which make the model bad for study interfaces between systems of different kinds. However, the time step in the mW-model can be as large as 10 fs.

3.5 Vashishta-Kalia-Rino-Ebbsjø Potential

The Vashishta-potential [70] (usc-potential³), is a molecular dynamics potential modeling the structure of silica. It was proposed by P. Vashishta, R. Kalia, J. Rino and I. Ebbsjø in 1990. Later, Vashishta extended the potential to model interfaces between silica and water. The implementation of this potential is closed source, but the potential form and its structure are publicly available. Pairwise interactions are modeled with a repulsive, Coulomb, dipole and van der Waals term. A three-body term accounts for the bending of covalent bonds. To describe interactions between water and silica, the potential has a bond-order part, which weights the oxygen-oxygen interaction in water or silica. Because this potential does not incorporate any static bounded interactions, it can model reactions and fracturing of silica. This is a property the ClayFF lacks, because atoms are bonded to each other throughout the whole simulation. ClayFF also lack the weighting of interactions according to the surroundings of atom pairs. Here we present the form of the usc-potential, used for both water and silica. The total energy of the system can be written as

$$E_{\text{tot}} = \sum_{i < j} V_{ij}(\mathbf{r}_{ij}) + \sum_{i < j < k} V_{ijk}(\mathbf{r}_{ijk}), \quad (3.16)$$

where $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$, \mathbf{r}_i is the position of the i 'th atom. The two-body terms are on the form

$$V_{ij}(r) = \underbrace{\frac{H_{ij}}{r^\eta}}_{\text{Repulsion}} + \underbrace{\frac{Z_i Z_j}{r} e^{-r/r_{1s}}}_{\text{Coulomb}} - \underbrace{\frac{D_{ij}}{2r^4} e^{-r/r_{4s}}}_{\text{Dipole}} - \underbrace{\frac{w_{ij}}{r^6}}_{\text{vdW}}. \quad (3.17)$$

Here H_{ij} is the strength of the steric repulsion, Z_i is the charge of atom i , r_{1s} is the screening length of the Coulomb interaction, D_{ij} is the strength of the dipole-dipole interaction, r_{4s} is the screening length of the dipole interaction and w_{ij} is the van der Waal strength. These parameters can be found in [70].

To account for bond bending and oxygen-hydrogen angles as well as silica-oxygen angles, a three-body term is added. It has the form

$$V_{ijk}(r_{ij}, r_{ik}) = B_{ijk} \exp\left(\frac{\xi}{r_{ij} - r_0} + \frac{\xi}{r_{ik} - r_0}\right) \frac{(\cos \theta_{ijk} - \cos \theta_0)^2}{1 + C_{ijk}(\cos \theta_{ijk} - \cos \theta_0)^2}. \quad (3.18)$$

In equation 3.18, B_{ijk} is the strength of the three-body term, ξ and C_{ijk} are constants, and θ_{ijk} is the angle formed by \mathbf{r}_{ij} and \mathbf{r}_{ik} .

The two-body part is truncated at r_c and shifted for $r < r_c$. Since the cutoff is relative short for this potential the energy gap at the truncation length is vital for the behavior of the potential. This can be stated as

$$V_{ij} = V_{ij}(r) - V_{ij}(r_c), \quad (3.19)$$

³usc: University of Southern California

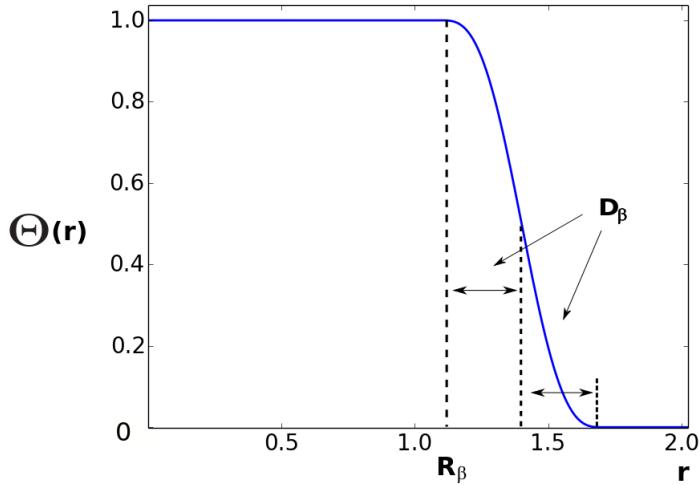


Figure 3.5: Plot of the bond order cutoff scheme. A continuous derivative at the cutoff prevents a sudden jump in energy.

where we calculate the potential value at r_c and subtract this from the potential value at r .

3.5.1 Bond Order Scheme

The interaction between pair of atoms in silica and water is not handled by the Lorentz-Berthelot mixing rule as in the ClayFF. This rule is probably the most efficient way of handling interactions, but it does not capture chemical reactions. There are several bond-order potentials e.g. Tersoff, REBO and the ReaxFF potentials [62, 7, 69]. These potentials have an advantage in comparison with other potentials because they can model different bonding states of an atom. They are all based on the same principle; bonding strength depends on the environment of the atom considered. In the usc-potential O-O interactions are handled according to the number of surrounding H and Si-atoms. The bond order scheme in the usc-potential is given as

$$V_{ij}(r_{ij}) = V_{ij}^{(\text{silica})}(r_{ij})(1 - f_{ij}) + V_{ij}^{(\text{water})}(r_{ij})f_{ij}, \quad (3.20)$$

where $V_{ij}^{(\text{silica})}$ and $V_{ij}^{(\text{water})}$ are the oxygen-oxygen two-body part in pure silica and water. Here f_{ij} is a weight factor defined as

$$f_{ij}(n_i^{\text{Si}}, n_i^{\text{H}}, n_j^{\text{Si}}, n_j^{\text{H}}) = \frac{n_i^{\text{H}} + n_j^{\text{H}}}{n_i^{\text{Si}} + n_i^{\text{H}} + n_j^{\text{Si}} + n_j^{\text{H}}}, \quad (3.21)$$

this is a smooth function of the number of Si and H-atoms surrounding two oxygen atoms. The variable η_i^β , where β is the Si or H-type atom, is the number

of atoms around the i 'th oxygen atom. To avoid a step function we want to model the number n_i^β as a continuous function Θ . This function is given as

$$n_i^\beta = \sum_{k \in \beta} \Theta(r_{ik}), \quad (3.22)$$

and varies smoothly from 0 to 1, depending on the distance between the i 'th oxygen and its k 'th type neighbor

$$\Theta(r) = \begin{cases} 1 & (r \leq R_\beta - D_\beta) \\ 1 - \frac{r-R_\beta+D_\beta}{2D_\beta} + \frac{\sin(\pi(r-R_\beta+D_\beta)/D_\beta)}{2\pi} & (R_\beta - D_\beta \leq r < R_\beta + D_\beta) \\ 0 & (R_\beta + D_\beta \leq r) \end{cases}$$

A plot of $\Theta(r)$ is shown in 3.5. The values of R_β and D_β are closed source.

3.5.2 Note on the Bond Order Scheme

The bond order part might seem a bit cryptic at first sight, but let us clarify how it is supposed to work. In equation 3.20, f_{ij} is a weight factor; it determines whether the interaction is pure silica or pure water. If there are no H-atoms in the vicinity of an oxygen-oxygen pair, it is interpreted as a pure silica interaction, $f_{ij} = 0$ and there is no water contribution to the two-body potential. If we consider an oxygen-oxygen interaction in a region of bulk water, we find the value of f_{ij} to be 1, and there is no silica contribution to the oxygen-oxygen interaction.

3.6 Reax Force Field

The reax force field (ReaxFF) was developed by van Duin et al. [69] in 2001. It was developed for reactive systems, and incorporates a bond order distance scheme to model the dissociation of bonds. The force field was originally created to study complex reactions in hydrocarbons. At the time it was about 10000 times faster than quantum chemical calculations. Since 2001 new parameters have been developed for other systems, such as clay-zeolite composites [47].

In ReaxFF, the overall energy of the system is described by 13 terms; the version implemented in LAMMPS is on the form.

$$\begin{aligned} E_{\text{system}} = & E_{\text{bond}} + E_{\text{lone pair}} + E_{\text{over}} + E_{\text{under}} + E_{\text{valence}} + E_{\text{pen}} + E_{\text{coa}} + \\ & E_{\text{C2}} + E_{\text{tors}} + E_{\text{conj}} + E_{\text{H-bond}} + E_{\text{vdW}} + E_{\text{Coulomb}} \end{aligned}$$

Here E_{bond} is the bond order energy between a pair of atoms, the *lone pair* accounts for the valence electrons not shared with other atoms e.g. 2 for oxygen.

The over and under coordination term penalizes deviation from the optimal number of lone pairs. E_{valence} is the valence energy contribution from valence angle terms. The pen term handles the stability of a configuration where double bonds are sharing an atom in a valence angle. coa is a three-body conjugation term, then there is a correction for a C₂ molecule. The remaining parts of the potential constitutes of a torsion term, four body conjugation term, a hydrogen bond term and finally, non-bonded interactions by vdW and Coulomb. For a more thorough dissection of the reax potential, see van Duin's supplementary material [15].

Chapter 4

LAMMPS

LAMMPS (Large-scale Atomic/Molecular Massively Parallel Simulator) is a molecular dynamics simulation package developed at Sandia National Laboratories [48]. It is written in object-oriented C++, making it easy to add extensions. The source repository contains around 940 files, if we assume an average line number in each file of about 100 (I believe this is a low shot) it adds up to 94000 lines of code. In addition you have source files in different additional packages e.g. the *REPLICA* package, which are not counted in. Among all the files there are a variety of thermostats, potentials and general tools it would have taken years to implement on your own. This is why we argue for the advantages of knowing how to develop new LAMMPS features.

LAMMPS is not unique; there are other molecular dynamics software's such as GROMACS, OpenMD, Amber, CHARMM and NAMD. Not all of these are open source packages. We have however chosen to work with LAMMPS, because of its reputation and extensive user-manual.

There is an excellent guide to pitfalls and usage of the LAMMPS code ¹. But it will not help you if you are interested in extending or adding features to the LAMMPS package. We will therefore in detail outline the implementation of the usc-potential in LAMMPS(chapter 7.1). In this chapter we dissect an input script and illustrate how the class hierarchy in LAMMPS is structured. But first, a brief overview of installation of the software.

4.1 Installing LAMMPS

LAMMPS is downloaded from this link ², on an Ubuntu machine we just type

```
$ sudo apt-get install lammps
```

¹<http://lammps.sandia.gov/doc/Manual.html>

²<http://lammps.sandia.gov/download.html>

in a terminal. We will then in our home directory, see a repository named lammps-(date of latest version). For guides on how to build and install additional packages on a high performance computing cluster, see appendix B.

4.1.1 Compiling LAMMPS

To run the software we need an executable, when the download is complete we compile the source code by the following commands, if we are on an ubuntu machine.

```
~/lammps-version/src$ make ubuntu
```

For checking and installing additional packages we type

```
~/lammps-version14/src$ make package-status
```

This gives us a list of the packages available in our version of LAMMPS, and if they are installed or not. For all the options just type

```
~/lammps-version/src$ make package
```

which lists all the additional package alternatives. For more information about the available packages and requirements we encourage the reader to check the LAMMPS manual, linked on the previous page.

4.2 LAMMPS Input Script

LAMMPS is unable to run if we don't provide any input script. From this script we control the whole simulation. Molecular topology, units, potential forms, ensembles and so on. The input varies from simulation to simulation, and it can be a handful getting the overview in the beginning. To ease this somewhat steep learning curve, we will stepwise introduce you to an input script for simulating bulk water with the SPC-model. Below we have presented the whole input script.

```
Lammps input script

units           real
dimension       3
boundary        p p p
atom_style      full

kspace_style    pppm 1.0e-4

read_data       water.data

variable        nx equal 6
variable        ny equal 6
variable        nz equal 6

replicate       ${nx} ${ny} ${nz}

group           wo type 1
group           wh type 2
group           spc tyle 1 2

pair_style      lj/cut/coul/long 10

pair_coeff      1 1 0.1553 3.166
pair_coeff      1 2 0.0 0.0
pair_coeff      2 2 0.0 0.0

bond_style      harmonic
bond_coeff      1 554.1349 1.0

angle_style     harmonic
angle_coeff    1 45.7696 109.47

velocity        all create 300.0 104112 dist gaussian

neighbor        2.0 bin
neigh_modify    every 1 delay 0 check yes one 3000

fix             constrain all shake 1.0e-4 20 0 b 1 a 1
fix             1 all npt temp 300.0 300.0 100 iso 1 1 1000.0

dump            1 all custom 50 bulkWater-*.txt id type mol x y z
restart         100000 bulkWater-*.restart

timestep        1
run             200000
```

Each line in the above input script is evoking a process in LAMMPS. In the following we will briefly explain what happens.

```
units           real
```

This part means we are using real units in LAMMPS, the distance is then in Ångstrøm, energy is in kcal/mol and temperature in K. Keeping track of the units is vital when we ask LAMMPS to dump data for post-processing.

```
boundary       p p p
```

The boundary line is important for how LAMMPS is setting the boundaries, **p** is periodic and since we got three of them, our system is periodic in all three directions. This can be set to **f** for non-periodic and fixed, and **s** for non-periodic. Non-periodic boundary conditions are often used in simulations where we want to model a surface.

```
atom_style     full
```

Here we are setting what attributes that belong to the atoms. Be careful to use attributes that you need for your simulations, because it cannot be changed after it is set. The choice of style affects which quantities that is stored for each atom.

```
kspace_style   pppm 1.0e-4
```

For long range interactions we have to specify a long range solver. The above line shows how this is done for the particle-particle-mesh method. Another alternative is the Ewald method.

```
read_data      water.data
```

In this example, we define the molecular topology of a water molecule in the **water.data** file. How this file is setup can be found in the LAMMPS manual.

```
variable       nx equal 6
```

The **variable** command is used to set how many replicas we want of the water molecule in the x, y and z-direction.

```
replicate      ${nx} ${ny} ${nz}
```

To replicate the molecule in three-directions we use the **replica** command, LAMMPS has to be built with the additional package *REPLICA* for this command to work.

```
group          wo type 1
```

When the atomic system is setup, we group the different types of atoms in groups. By setting a group, we can make alterations to each of the specific groups, such as setting the velocity for oxygen atoms and not for hydrogen atoms.

```
pair_style      lj/cut/coul/long 10
```

In this line we declare what kind of interatomic potential we want to use. For the SPC-model this is a short range LJ-potential and a long range Coulombic term. The 10 after the declaration is the cutoff length in the units we have chosen. There are a variety of interatomic potentials in LAMMPS, which again can be used to model different systems. Notice that adding potentials to our input script, usually means that we must supply more parameters. As for the Coulomb interaction we must set the charges of the atoms, and decide a cutoff radius for the Coulomb force. If this is not set, LAMMPS use the same cutoff as for the vdW interaction.

```
pair_coeff      1 1 0.1554 3.66
```

From chapter 3.1, we saw that the LJ-potential contains a set of parameters σ and ϵ , these needs to be set explicit for the system we want to study.

```
bond_style      harmonic
bond_coeff      1 554.1349 1.0
```

From the molecular topology file we have explicit set the bonds and angles between types of atoms. In the above lines we decide what bond style and angle

style to be used. Here k is the strength and 1.0 is the equilibrium length of this harmonic potential.

Then we need to set the initial velocity of our atoms, this is done by the command

```
velocity           all create 300.0 10345 dist gaussian
```

Here we assign all the particles a velocity corresponding to 300 K. The number 10345 is a seed needed for the random number generator. We also set the distribution of the velocities to have a Gaussian form.

Further we have two commands, setting the neighbor list of each atom, and updating the neighbor list of an atom. These are given by

```
neighbor          2.0 bin
neighbor_modify   every 1 delay 0 check yes one 3000
```

The `neighbor` command sets parameters that affects the building of neighbor lists in LAMMPS. All atom pairs within a cutoff distance, equal to the force cutoff plus the skin distance are stored in the list. The number is given in length units. The `neighbor_modify` command has more input parameters, I will only go through what is set in the example above. `every 1` means that we build the list one step after the delay has passed. And as we see here, we have no delay, this means that we update the neighbor list every time step. In LAMMPS the computation of neighbor lists are not that expensive, and we are recommended to do it as in the above example. If the `check` keyword is set to yes then we only build the neighbor list on a particular time step, or if the atom has moved more than half the skin distance.

To handle the intra-atomic oscillations we impose the SHAKE algorithm to the harmonic bond and angle term. This algorithm is covered in the appendices. It allows for a larger timestep, because we constrain the bond length.

```
fix               constrain all shake 1.0e-4 20 0 b 1 a 1
```

To get the system running we need to invoke an integrator, this is done by the `fix` command, for the NPT-ensemble the input is given as

```
fix               1 all npt temp 300.0 300.0 100 iso 1 1 1000.0
```

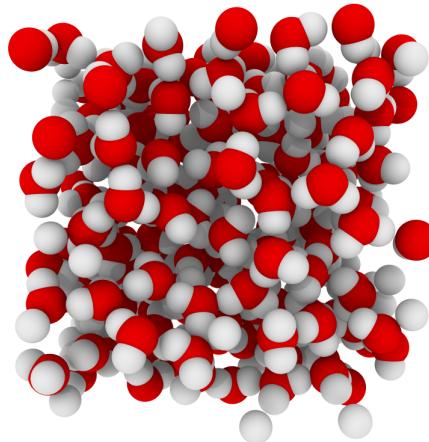


Figure 4.1: Bulk water from the outlined input script

It's actually not the Verlet class that calculates the trajectories of the atoms, but the fix and the style of the fix. For this particular fix we run the NPT-ensemble, with an isobaric pressure of 1 atm. The pressure is adjusted every 1000 step.

```
dump          1 all custom 50 bulkWater-*.txt id type mol x y z
```

The `dump` command allows us to post-process atom trajectories or other outputs. Here we write the x , y and z positions for all atoms every 50 time step, and dumps it to a text file. We also need to specify the time step, and for how long we want the simulation to run.

```
timestep      1
run           200000
```

In real units, this give us a time step of 1.0 fs for 5000 steps, which gives us a simulation time of 5000 fs.

A result of the above simulation can be seen in figure 4.1. Hope this motivates you to dig deeper into LAMMPS.

4.3 LAMMPS Structure

The LAMMPS program package is written in C++, it is structured in an object oriented fashion, a feature that makes it easy to extend the software. A schematic class hierarchy can be found in figure 4.2. To give a quick summary of the figure, we see that there is a superclass `LAMMPS_ST`. There are a range of subclasses to

this parent class, some of the most important are the neighbor list class, atom class, force class and the update class. These will all be explained in more detail, for those who are interested in developing new features in LAMMPS.

4.3.1 Class Hierarchy

Each box in figure 4.2 is a class and has a pair of associated source files in the `src` directory. The `LAMMPS.cpp/h` file is the parent class, it holds an instance of LAMMPS, and is callable by e.g. `main.cpp`, which calls it and passes through the input file. Other top level classes are given their own box; these classes are visible to all other classes and functions. In the boxes to the right we find virtual parent classes. They can have several child classes. Such as `Pair` has `lj/cut/coul/long` or `lj/cut/coul/debye`, those are pair potentials.

The `memory` class allocates memory for vectors and arrays. In the `universe` class, partitions of processors are set up so that multiple simulations can be run. Atom information, such as charge and molecule number is stored in the `atom` class. Integrating the equations of motion is done by the `update` class; this also holds a minimization tool. Keeping track of neighbor atoms is done in the `neighbor` class, it builds and stores neighbor lists. When a neighbor list is required, the `neighbor request` class gives you the required neighbor list of an atom. The `Force` class computes interatomic forces. We will dissect some of the more important classes in LAMMPS, by the means that these are the classes most important to understand; if you want to implement a new potential.

4.3.2 The Potential Class

We specify the interatomic potential we want to use by the `pair_coeff` command. It is relative easy to implement a new interatomic potential, if you just know how the different parts of LAMMPS are working. There is a vast number of potentials in LAMMPS, LJ, Coulomb, Tersoff, ReaxFF you name it. The potential energy of an atom is calculated from the interatomic terms, to find the force between atoms we have to differentiate the potential energy terms. Potentials we have encountered in LAMMPS, does not compute a numerical derivate from the potential energy values. LAMMPS discerns between many-body potentials and two-body potentials. A `kspace_style` is often not needed for a many-body potential, e.g. usc or Stillinger-Weber. According to what type of potential we are implementing, we advise you to choose an existing two-body potential for a two-body potential implementation, and the same goes for a many-body potential.

4.3.3 The LAMMPS Neighbor List

In LAMMPS we have access to different neighbor lists. These lists are built so that they fit different interatomic potentials, and how these potentials demands different sums over neighboring particles. The two main neighbor lists are the full neighbor list and the half neighbor list. They are different in the way that for the half neighbor list, atom i stores the information about atom j , atom j does however not store that i is a neighbor. For the basic two-body LJ-potential, this makes the computation more efficient. The half neighbor list is not able to sum over correct neighbors in a many-body potential, such as the SW or usc-potential, the reason for this is that we need to know that j is i 's neighbor. When you ask for a full neighbor list and still have to compute a pairwise force, it is normal to skip half the list. It is important to remember that we have to ask LAMMPS explicit for a certain neighbor list.

4.3.4 Pressure Calculations in LAMMPS

LAMMPS calculates the pressure in the same way we stated in chapter 2.2. For these computations LAMMPS has an own built in functions called `ev_tally`, which takes the force as an input argument for computations of the virial. It also finds the pressure tensor that is the pressure in different directions, such as P_{xx} or P_{xy} , all in all nine values for the pressure. This comes in useful when we want to find the stress as a function of applied strain.

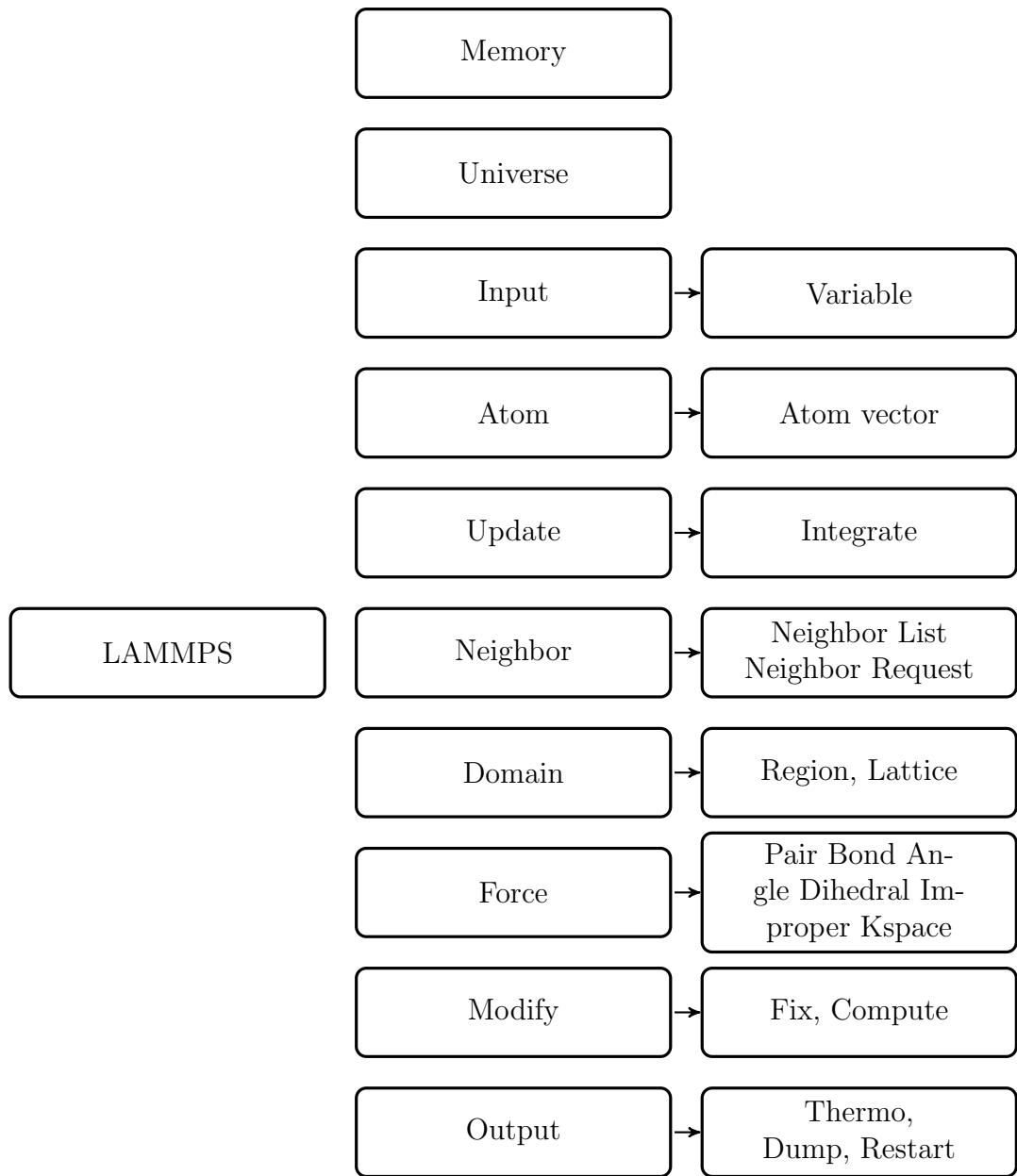


Figure 4.2: LAMMPS class hierarchy. Some of the more important classes are explained in the text.

Chapter 5

Hydrodynamics

Computer simulations have made it applicable to study fluid flow in micro-/nanochannels, these systems are difficult to study experimentally and new knowledge of how fluids behave at small scales is continuously being revealed. This has led to opportunities in the development of micro and nanofluidic devices. Exploration of oil and gas confined in tight rocks has opened a field of study on how micro/nanochannels affect fluid flow, and how this flow can induce stress and alter the porosity of the rock. Optimizing desalination membranes, for max flux of water and holding back solutes, can be studied by molecular dynamics simulations. These are just a few examples of applications of fluid flow at nanoscale.

In nanosized pores the hydrodynamic results as we know it breaks down, we can no longer assume no-slip boundary conditions, and little is known about how surface effects alters the flow in tight pores. The Navier-Stokes equation, which describes macroscopic flow needs to be modified to account for slip boundary conditions. In this chapter we state the basic theory behind fluid dynamics and how some of this theory is applied to nanoscale flow. We discuss how the flow is initiated, gravity or pressure driven, and the difficulties of discrete modeling of fluid flow.

5.1 Fluid Dynamics

In continuum fluid dynamics the Navier-Stokes equation has successfully described macroscopic flow for decades. One of its most general forms is presented below

$$\rho_m \frac{D\mathbf{v}}{Dt} = \rho_m \mathbf{F} - \nabla P + \eta \nabla^2 \mathbf{v} + \eta \nabla(\nabla \cdot \mathbf{v}) \quad (5.1)$$

Here \mathbf{v} is the velocity, P is the pressure, ρ_m is the density and η is the viscosity. This equation can be solved for a plane Poiseuille flow between two parallel plates; this is the analytical solution of the flow problem addressed in this thesis. From

equation 5.1, we find an expression of the viscosity η , by assuming the following,

- the velocity \mathbf{v} at the boundary is zero.
- the viscosity η and the density ρ_m is constant.
- there is only one non-zero velocity component in the system, and that is in the direction of the flow \mathbf{v} .
- there is no variation in the velocity in y -direction.

From these assumptions, we find that the left hand side of equation 5.1 is zero, so is the non-linear term $\eta \nabla(\nabla \cdot \mathbf{v})$. Since we are using a constant force instead of a pressure difference $\Delta P = 0$, we are left with

$$-\rho_m F_x = \eta \nabla^2 \mathbf{v} .$$

Where F_x is the applied force in the x -direction and $\nabla^2 \mathbf{v} = \frac{\partial^2 v_x}{\partial z^2}$ describes the velocity profile in the direction of the force. After some basic algebra we obtain for the viscosity

$$\eta = -\frac{F_x \rho_m}{\left(\frac{\partial^2 v_x}{\partial z^2}\right)}. \quad (5.2)$$

There are other methods to calculate the viscosity [4], but in this theses we will use equation 5.2.

5.1.1 Flow Between Parallel Plates, Analytic Solution

A continuum description of the flow profile between two parallel plates (Poiseuille flow) is derived from the NS-equation. In our case there is no pressure gradient, only a pure gravity driven flow. NS is then again written as

$$-\rho_m F_x = \eta \nabla^2 \mathbf{v} , \quad (5.3)$$

we let the fluid flow be in the x -direction, and only depend on the distance z between the plates. The equation is then on the form

$$\frac{d^2 v_x}{dz^2} = -\frac{\rho_m}{\eta} F_x. \quad (5.4)$$

We note that the left hand side is only dependent on z , and the right hand side only depends on x . This is only true if both sides are constants. For the velocity we now arrive at

$$v_x = -\frac{\rho_m}{2\eta} z^2 + az + b , \quad (5.5)$$

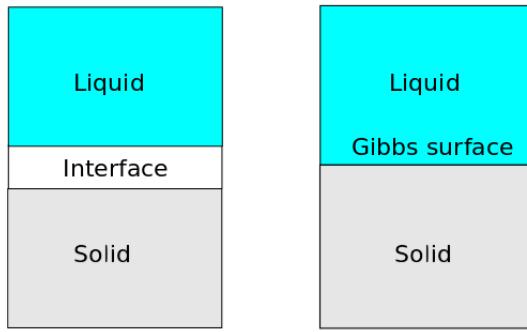


Figure 5.1: Principle of Gibbs dividing surface. To the left we have the real system, and to the right we have an ideal system.

where a and b are constants, which we find from the no-slip boundary conditions $v_x(0) = 0$ and $v_x(H) = 0$. The velocity profile is then a parabola on the form

$$v_x(z) = \frac{\rho_m}{2\eta} z(H - z) . \quad (5.6)$$

Equation 5.6 is only true for no-slip boundary conditions, and no variations in shear viscosity in space or time. This expression is expected to break down for channels on nanometer scale.

5.1.2 Boundary Conditions

Because of the discrete nature of molecular dynamics it is not trivial to determine where a surface, z_{surf} is located. The particles are point particles, and a point does not really have an extension in space. There are several methods proposed to measure the surface, to mention some

1. Green-Kubo relation for the location of the fluid-solid interface.
2. We can split the pore into a region where the fluid phase is where the fluid behaves as in bulk, and the solid phase where it is discrepancy from bulk.
3. Gibbs dividing surface gives us a measure of where the liquid/solid interface is, by integrating the density. This is described as

$$\int_0^{z_{\text{surf}}} [\rho_{\text{bulk}} - \rho(z)] dz = \int_{z_{\text{surf}}}^{+\infty} \rho(z) dz \quad (5.7)$$

We chose to use equation 5.7, and Gibbs dividing surface to solve the surface location problem. This gives us an interpretation of the surface as in figure 5.1. The principle behind this figure and the Gibbs dividing surface can be explained

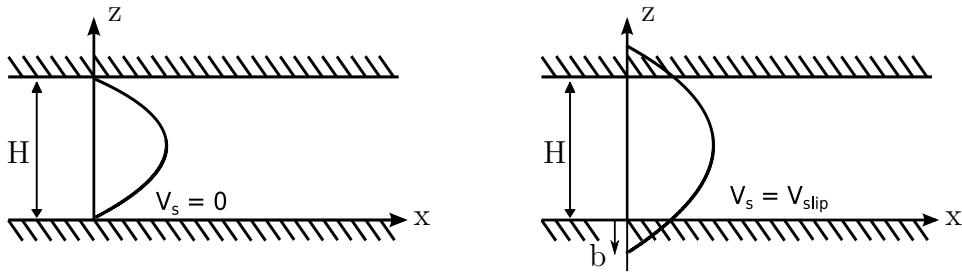


Figure 5.2: Velocity boundary condition and slip length

in the following way. To the left in figure 5.1 we see the real system, by real we mean that the bulk density of the liquid does not transfer directly to the bulk density of the solid phase. There is an inhomogeneous layer between, with fluctuating density. In the figure to the right, we have an ideal system. The bulk density of the liquid phase transfers directly to the bulk density of the solid phase. The distance of the inhomogeneous phase is now zero. Gibbs surface is the plane where this is true.

5.1.3 Breakdown of Continuum

Why should there be any problem obtaining the correct velocity profile in nanopores with continuum equations, you might ask? Well, for pore sizes above 10 nm the deviation between continuum and molecular dynamics simulations are small when predicting the Poiseuille flow rate [24]. However, for pore sizes below 10 nm the errors are larger, this is explained by the fluctuating density of the fluids at the boundary. This in turn is caused by the high confinement of molecules close to the surface. We might think that when the molecules are forced to be more rigid at the boundary, the NSE with its non-slip boundary conditions should work just fine, but for the derivation of the velocity profile we assumed a constant density and viscosity in the whole system.

5.1.4 Slip Corrections

We usually assume no-slip or stick boundary conditions, this corresponds to a vanishing fluid velocity at the fluid-solid interface. But because of the fluid density fluctuations at the fluid-solid interface, we cannot assume a homogeneous viscosity across the pore. The slip length is expressed as

$$b = \pm \frac{v_x(z_{\text{surf}})}{\left(\frac{dv_x}{dz}\right)_{z_{\text{surf}}}}, \quad (5.8)$$

here v_x is the velocity at the Gibbs dividing surface. We note that b is positive when the liquid velocity at the wall is positive. Negative values of b corresponds

to back-flow near the wall. The calculation of b is limited to pores where the gap is much larger than b [46]. In figure 5.2, the slip length b is illustrated for a pore with height H . Notice that the velocity at the boundaries now is non-zero.

5.2 Gravity driven Flow

Gravity or acceleration driven flow, is the simplest way to model fluid flow in molecular dynamics. We simply add a constant force \mathbf{F}_i to all the particles. If we keep adding a constant force to the particles without there being any form of friction slowing the particles down, they will continue to accelerate. If we look at flow between two parallel plates or in a carbon nanotube for that sake, the particles in the wall will exert a force on the moving molecules. After a while this force and the added force will cancel each other out, and you get $\sum F = 0$. We are then able to measure a flow profile between the plates.

The reason we often want to avoid inducing an acceleration driven flow is due to the complexity of the material, of which our molecules are passing. Flow between two parallel plates should work fine, but if the plates are rib roughened we get non-physical behavior where the particles are colliding with the wall. Other examples are flow of water past a protein; the water will just keep pressing against the protein molecule instead of flowing around it.

5.3 Pressure driven Flow

Pressure driven flow, is flow driven by a difference in pressure between two regions A and B . Creating a pressure difference in molecular dynamics is not as trivial as for an ideal gas. In addition to the kinetic energy, the system also depends on potential energy terms e.g. LJ or Coulomb. An insertion-deletion procedure of particles is no longer trivial, because we have to compute a potential energy terms as well. Some of them decaying as r^{-1} , resulting in a greater computational effort needed. LAMMPS provides features for inserting particles like the `fix gcmc`. This fix attempts insertion of water molecules according to a Monte Carlo scheme discussed in Frenkel [18]. Just recently they updated the fix to handle long range forces. Limitations of insertion-deletion methods are listed below.

- A semi-permeable membrane at $x = L$ and $x = 0$ is needed, or else the particles would chose the easiest path to lower pressure.
- The temperatures in the insertion-deletion regions have to be individually controlled, due to fluctuating particle numbers.
- To get a steady velocity profile the duration of the insertion-deletion would have to last for a long time. Most likely several nanoseconds.

- The reservoirs would have to be large, so that deletion of one water molecule is not resulting in an unreasonable large pressure difference.
- Large reservoirs imply more computational cost, for an already computational expensive method.

Flow by true pressure difference is in other words not easy. As we will see in the next section, a number of schemes have tried to solve this.

5.3.1 Review of Schemes

Several schemes have been proposed to obtain pressure driven flow. Li et al. proposed a semi-permeable membrane at $x = 0$ reflecting particles with a probability p . Keeping the particle number and temperature constant [34]. Although the method is simple and easy to implement, it suffers from the fact that it is not translational invariant in the x-direction. Li's membrane method has been successfully applied to simulate water flow through a carbon nanotube [63]. To et al. derived a method where they used that a constant pressure difference is related to a constant difference in the squared velocities at the inlet and outlet of a system [66]. McNamara et al. presented a scheme where they added a pressure force per particle from a continuum distribution of pressure to the equations of motion [39].

Chapter 6

Material Properties

In this chapter we explain basic elasticity and fracture theory. For solids, molecular dynamics simulations are an excellent method to investigate what happens at the atomic scale when a material is being deformed or even fractured. Information of how the fracture tip evolves at nanoscale, and how impurities and other factors influence fracture propagation can be gathered. Molecular dynamics simulations of elastic and fracture properties of silicates have been subject to numerous studies the last decade [44, 41, 9], but for clay materials there has been little to no research [23]. This is probably due to the complexity of clay minerals compared to silica structures, and the possibility for applications. Silicates being important in electronic devices and solar cells, while clays just recently have become interesting due to shale oil and gas. This has resulted in the development of molecular dynamics potentials parametrized for clays, like ClayFF and ReaxFF. With these force fields we are able to compute elasticity and fracture mechanics of clays to a certain extent. By elasticity, we here mean deformation to the point that the deformation is reversible when the forces of deformation are removed [64]. The theory of elasticity is developed for continuous media, but we will assume that it applies at atomic scale. This is a valid assumption as long as we consider homogeneous materials with some sort of isotropic property.

6.1 Basic Linear Elasticity Theory

How a system responds to stress and strains can be expressed by numbers that relates the deformations of the system to the stresses applied. Young's modulus and Poisson's ratio are the two we will focus on in this thesis. Young's modulus relates the stress applied, to the resulting strain. While Poisson's ratio tells you how much a system expands or contracts, in the direction perpendicular to the applied stress. There are other material properties that could be computed, such as the bulk modulus. We limit ourselves to introduce the theory of Young's modulus and Poisson's ratio.

6.1.1 Stress and Strain

When we apply force on a body, we observe to some extent a deformation, a change in shape and volume. This deformation is mathematically described as strain, ϵ . When a body is deformed, every point in that body changes its location with a distance $d\mathbf{r}$. The stress σ , the body experiences when subject to a strain ϵ , exhibits a linear relationship. Stress is said to work in the i-direction on the j-plane of an object. From this we observe that stress with repeating indices' e.g. σ_{ii} , are normal stress, and shifting indices' are shear stress τ_{ij} , see figure 6.1. The same accounts for the strain, i.e. normal and shear strain. Strain applied in the x-direction is expressed as

$$\epsilon_x = \frac{\sigma_x}{E} ,$$

where E is Young's modulus. So by straining the material in the x-direction and evaluating the stress in same direction, we find Young's modulus of a material. Young's modulus, or the modulus of extension, gives us information about the ease of which the material stretches. The basic law describing how a material deforms and returns to its original shape due to external forces is *Hooke's law*, it is defined in the forms of stress and strain as explained above.

6.1.2 Poisson's Ratio

The relation listed above is experimentally established [64], and known as Hooke's law. When a body is strained in the x-direction as above, a contraction in the other two directions follows. This relation is expressed as

$$\epsilon_y = -\nu \frac{\sigma_x}{E}, \quad \epsilon_z = -\nu \frac{\sigma_z}{E} ,$$

where the prefactor ν , is the Poisson's ratio. If the material is isotropic, we can find the relationship between shear strain and shear stress from the two constants, E and ν . For other non-isotropic materials the coupling is a bit more complicated.

Isotropy

There are several forms of isotropy. In this thesis we talk about isotropic and transverse isotropic materials. It is important to state the difference between an isotropic and a transverse isotropic (TI) material, because properties of a system become directional dependent according to type of isotropy. Glass is an isotropic material; all properties are the same in the x, y and z-direction. A TI material has isotropic properties in two directions, but it is anisotropic in the third direction.

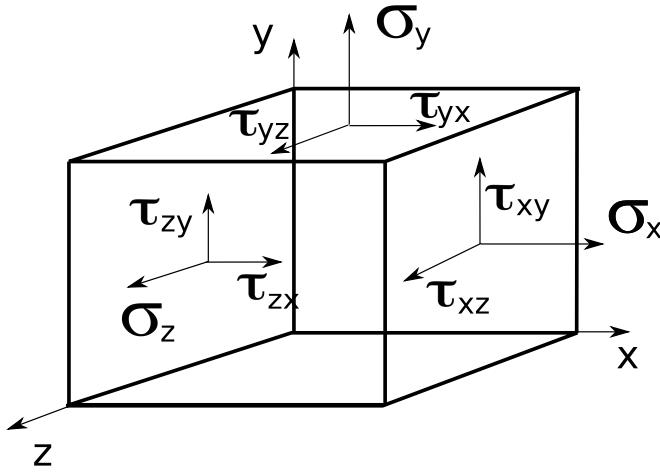


Figure 6.1: Directions of stress. Three positive and three negative contributions for the three planes, results in a stress tensor with 81 components.

6.1.3 Stiffness Matrix

The stiffness tensor is a relationship between the applied strain and resulting stress (or opposite) in all directions of a material. From figure 6.1, we see that there are 81 different coefficients. A generalized Hooke's law is written on the form

$$\sigma_{ij} = C_{ijkl}\epsilon_{kl}, \quad (6.1)$$

here E_{ijkl} are the elasticity constants. This matrix would have a total of 81 elements, luckily symmetry arguments helps us get rid of some elements. For a system under shear, $\tau_{ij}^{\text{TOP}} = \tau_{ij}^{\text{BOTTOM}}$, from Newton's second law. Shear stress with indices' ij and ji is set to $\tau_{ij} = \tau_{ji}$, from conservation of torque. We are then left with six components giving 36 elements in the matrix. Next we replace the quadruple subscripts with the easier notation C_{ij} . For isotropic materials the elasticity constants are directional independent. For transverse isotropic materials, such as clays, the constants are directional dependent. A TI material is symmetric around an axis which is normal the plane of isotropy. Hence, we need a new expression for the stiffness tensor C_{ij} . For symmetric and thermodynamic reasons the stiffness tensor reduces to 21 independent variables. For a TI material it can be written as

$$\begin{bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_{xz} \\ \tau_{yz} \\ \tau_{xy} \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & 0 & 0 & 0 \\ C_{12} & C_{11} & C_{13} & 0 & 0 & 0 \\ C_{13} & C_{13} & C_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{44} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{66} \end{bmatrix} \begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_z \\ \gamma_{xz} \\ \gamma_{yz} \\ \gamma_{xy} \end{bmatrix}$$

Here σ is the stress in either the x, y or z-direction, τ is the shear, ϵ is the strain in the three directions and γ is the strain due to shear stress. The physical significance of this matrix is that the strain and stress in one direction is coupled to the resulting strain in the two other directions. If we do matrix multiplication we see that $\sigma_x = C_{11}\epsilon_x + C_{12}\epsilon_y + C_{13}\epsilon_z$. Hence, if we were to find the stress in the x-direction this depends on the resulting strain in the two other directions.

6.1.4 Compliance Matrix

If we invert the stiffness tensor we find the compliance matrix. For a TI material it is on the form

$$\begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_z \\ \gamma_{xz} \\ \gamma_{yz} \\ \gamma_{xy} \end{bmatrix} = \begin{bmatrix} 1/E & -\nu/E & -\nu'/E' & 0 & 0 & 0 \\ -\nu/E & 1/E & -\nu'/E' & 0 & 0 & 0 \\ -\nu'/E' & -\nu/E' & 1/E' & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu' & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu' & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{bmatrix} \begin{bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_{xz} \\ \tau_{yz} \\ \tau_{xy} \end{bmatrix}$$

In the above setup of the compliance matrix we have assumed an isotropic xy-plane and rotational symmetry around the z-axis. We are left with five independent elasticity constants.

- E is Young's modulus in the isotropic direction.
- E' is Young's modulus in the direction perpendicular to the isotropic plane.
- ν is Poisson's ratio in the isotropic plane.
- ν' is Poisson's ratio in the direction perpendicular to the isotropic plane.
- μ is the shear modulus in the isotropic direction.
- μ' is the shear modulus in the plane perpendicular to the isotropic plane.

The shear modulus μ , can be expressed as a function of the other variables $\mu = \frac{E}{2(1+\nu)}$. Giving us five independent constants. To find μ' we have to shear the system.

How the elastic constants are coupled, is now seen by multiplying the compliance matrix. For the strain in the x-direction we get, $\epsilon_x = \frac{\sigma_x}{E} - \frac{\nu}{E}\sigma_y - \frac{\nu'}{E'}\sigma_z$. Note here that the directions do not seem to be independent of each other, seemingly increasing the difficulty of finding the constants. In chapter 11.2, we see that there are ways to avoid this.

6.2 Fracture Mechanics

Linear elasticity was defined as deformation of a system to the extent that the system returned to its original shape after deformation. In fracture mechanics, we deform a system beyond the point of reversible deformation. Study of fracture propagation has for a long time been considered a difficult task, due to the non-equilibrium state of the system. Atoms close to the fracture tip are under an enormous amount of stress and moving so fast that the concept of temperature is not defined clearly [28]. The advent of supercomputers has increased the interest in studying fracture phenomena's at nanoscale. Visualization techniques have allowed scientist to follow the fracture tip propagation and visualizing the stress field at nanometer resolution. Molecular dynamics potentials are developed to account for the bond breaking and forming under system strain, e.g. the reax or usc-potential. Here we present basic concepts of fracture mechanics.

Modes of Fracture

There are three main modes of fracture, mode-I opening mode, mode-II in plane shear and mode-III out of plane shear. For our simulations we only perform mode-I fractures. In fracture mechanics we distinguish between two types of fracture propagation.

There are two types of fractures, brittle and ductile. Brittle fracture is believed to involve bond breaking at the atomic scale. Due to the bond breaking, the fracture snaps through the material causing a sudden drop in the stress intensity. Glass is known to break in a brittle manner. Ductile fracturing on the other hand constitutes of a deformation of the material. The material remains in the deformed state until increased stress is applied. In ductile fracture voids are created in the vicinity of the fracture tip. As the stress increases these voids grow into adjacent voids and the fracture propagates [2].

6.2.1 Breakdown of Continuum in Fracture Mechanics

Shortly after World War-I, Griffith used energy balance and thermodynamics, and laid the foundation in the field of fracture mechanics [22]. The theories were applied to continuum solids, and described how an isolated crack behaves subject to a stress. For an isotropic linear elastic material, it can be shown that the a closed form solution of the stresses in the body, is on the form

$$\sigma_{ij}(r, \theta) = \frac{K}{\sqrt{(2\pi r)}} f_{ij}(\theta) . \quad (6.2)$$

Where, f_{ij} is a dimensionless function of the angle θ , from the fracture tip to the region we analyze. K is a measure of the intensity of the stress field near the crack tip. The problem with this equation and the continuum description

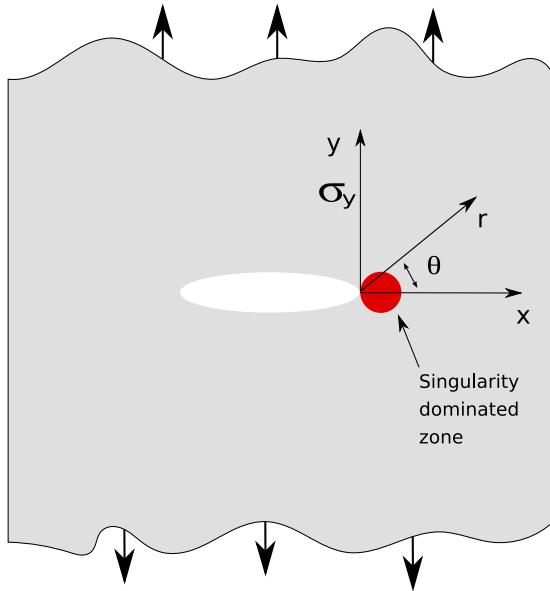


Figure 6.2: Singularity in continuum description of stress around the fracture tip.

of fracture is that for $r \rightarrow 0$, we have a singularity, i.e. the stress around the crack tip is in the order of infinity. This is why an atomic description of fracture mechanics has become more popular. For the situation depicted in figure 6.2, where we strain the system normal to the fracture plane, i.e. mode-I fracture. The closed form solution is

$$\sigma_y = \frac{K}{\sqrt{(2\pi r)}} \cos\left(\frac{\theta}{2}\right) \left[1 + \sin\left(\frac{\theta}{2}\right) \sin\left(\frac{3\theta}{2}\right) \right], \quad (6.3)$$

now $f_{ij}(\theta)$ is a cos function. For $\theta = 0$, the stress in the x and y-direction is equal leading to,

$$\sigma_x = \sigma_y = \frac{K}{\sqrt{(2\pi r)}}. \quad (6.4)$$

The stress intensity factor K , turns out to be very valuable, if this number is known, it is possible to solve for components of stress, strain and displacement as a function of r and θ [2]. However, it is inapplicable to determine the value of K at the point of singularity, unless we perform numerical simulations, like molecular dynamics. Later we will see how the stress intensity factor can be found for a system with finite width.

Part II

Implementations in LAMMPS and Simulation Procedures

Chapter 7

Implementation

In this chapter we will outline the implementation of the usc-potential in LAMMPS. As discussed earlier there are some challenges related to extending LAMMPS. It is hard to get an overview of all the source code in LAMMPS, when there are new additions monthly. Knowing what you want to do, and what parts of LAMMPS are important for you is the key. This is easier said than done, and we will supply you with a couple of tips on where to start. As stated in the goals for this thesis, we wanted to develop a reactive force field, with parametrization for clay minerals. By doing this in LAMMPS, we save a lot of work and it will be easier for others to use our code. We do not have to worry about getting the thermostats or integrator to work, all we have to focus on is implementing the potential form correct. In the following we will try to dissect the code we have written and propose it as an example of how to implement a many-body potential in LAMMPS.

7.1 Vashishta Potential

The theory and potential form of this potential is found in chapter 3. Here we will explain how the different parts of the potential were implemented in LAMMPS. We briefly discuss how we solved some challenges encountered in the implementation. We start with the two-body interaction and proceed to the three-body part. There will be listings of code where we want to state explicit how certain things were done.

7.1.1 Two-body Potential

The two-body part of the usc-potential is found in chapter 3. It consists of a repulsive part, Van der Waal, Coulomb and dipole term. A many-body potential requires a full neighbor list from LAMMPS. For the two-body part, there is no need to calculate the $j \rightarrow i$ interaction, when we already have computed the $i \rightarrow j$

interaction. Instead we use Newton's third law and skip half of the neighbor list. This is done in the following way

```

for (jj = 0; jj < jnum; jj++) {
    j = jlist[jj];
    qj = q[j];
    j &= NEIGHMASK;
    jtag = tag[j];

    if (itag > jtag) {
        if ((itag+jtag) % 2 == 0) continue;
    } else if (itag < jtag) {
        if ((itag+jtag) % 2 == 1) continue;
    } else {
        if (x[j][2] < ztmp) continue;
        if (x[j][2] == ztmp && x[j][1] < ytmp) continue;
        if (x[j][2] == ztmp && x[j][1] == ytmp && x[j][0] < xtmp) continue;
    }
}

```

This is the inner j-loop over neighbor atoms of atom i , which is the outer loop. Since we are using the full neighbor list from LAMMPS, where each atom knows its neighbors we need to skip half of them. In the full neighbor list i, j and j, i are listed. To save computational effort we only want to compute $i \rightarrow j$, and use Newton's third law for the $j \rightarrow i$. First we ensure that only half of the neighbor list is used. The `itag` and `jtag` are the ID's of the atoms. The atom ID's could be the same if we look at a small periodic system with a large cutoff, due to shadow atoms, hence the last `if` tests. The next part is to find the interatomic distance \mathbf{r}_{ij} ,

```

delx = xtmp - x[j][0];
dely = ytmp - x[j][1];
delz = ztmp - x[j][2];
rsq = delx*delx + dely*dely + delz*delz;

```

where `delx` is the distance between two atoms in the x-direction. Then for the atoms of `itype` and `jtype` we find the belonging parameters

```
ijparam = elem2param[itype][jtype][jtype];
```

Depending on the atom types we now are in possession of the two-body potential parameters. From this we calculate the two-body forces

```

void Pair_usc::twobody(Param *param, double rsq, double &fforce,
                      int eflag, double &eng, double &qtmp, double &qj)
{
    // Declaration of variables
    double r, rcut...

    // Evaluating the two-body part at rc
    coulcutfactor = exp(-rcut*r1sinv)*qtmp*qj*qqr2e;
    dipolecutfactor = -exp(-rcut*r4sinv);
    repulscut = param->eta*param->H*rcut9inv*rcut2inv;
    coulcut = coulcutfactor *(r1sinv + rcutinv)*rcut2inv;
    dipolecut = dipolecutfactor*param->D*(r4sinv*rcut4inv+4.0*rcut5inv)*rcutinv;
    vdwcut = -6.0*param->w*rcut8inv;

    // Repulsive force
    forcerepuls = param->eta*param->H*r9inv*r2inv;

    // Coulomb calculations
    coulscreenfactor = exp(-r*r1sinv);
    forcecoul = coulscreenfactor*qtmp*qj*qqr2e*(r1sinv+rinv)*r2inv;

    // Dipole calculations
    dipolefactor = exp(-r*r4sinv);
    forcedipole = -param->D*dipolefactor*(r4sinv*r4inv+4.0*r5inv)*rinv;

    // VdW calculations
    forceVdW = -6.0*param->w*r8inv;

    // Truncate potential at rc
    forcerepuls = forcerepuls - repulscut;
    forcecoul = forcecoul - coulcut;
    forcedipole = forcedipole - dipolecut;
    forceVdW = forceVdW - vdwcut;

    fforce = forcerepuls + forcecoul + forcedipole + forceVdW;
    if (eflag) eng = evdW + ecoul + erepuls + edipole;
}

```

Listing 7: Two-body potential calculations

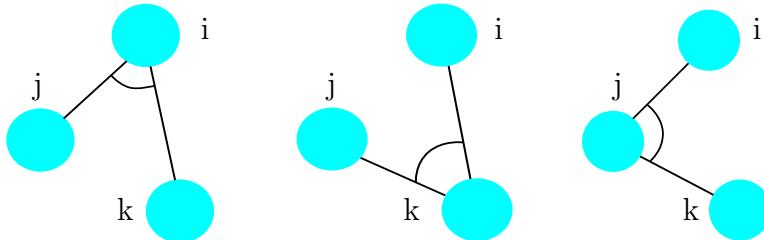


Figure 7.1: Looping through three-body configurations. The parameters need to be set explicit for the three interactions as stated in the potential file.

Here we have shifted the potential as explained in section 3.1. We found that this was crucial for our potential, probably due to the short cutoff length. Parameters, such as charge and potential strength is passed through to the two-body function, and the force is returned to the integrator by using Newton's third law. We

```
f[i][0] += delx*fpair;
f[i][1] += dely*fpair;
f[i][2] += delz*fpair;
f[j][0] -= delx*fpair;
f[j][1] -= dely*fpair;
f[j][2] -= delz*fpair;
```

multiply the pair contribution to the force by the direction of the interaction. In listing 7, we also compute the contribution to the potential energy. This is done by evoking the `eflag` tag. Hence we are able to plot the potential energy as a function of time.

7.1.2 Three-body Potential

For non-static molecular potentials we need a three-body part which depends on the distance \mathbf{r}_{ij} and \mathbf{r}_{ik} , that is from atom $i \rightarrow j$ and atom $i \rightarrow k$. A double loop over j and k is needed for each atom i . The angles θ_{ijk} are then found(figure 7.1). From the angles we compute the three-body force. We used the Stillinger-Weber, three-body potential as a template for the usc, three-body implementation.

```

void Pair_usc::threebody(Param *paramij, Param *paramik, Param *paramijk,
                         double rsq1, double rsq2,
                         double *delr1, double *delr2,
                         double *fj, double *fk, int eflag, double &eng)
{
    // Declaring variables
    double r1, rinvsq1...

    r1 = sqrt(rsq1);
    rinvsq1 = 1.0/rsq1;
    rainv1 = 1.0/(r1 - paramij->r0);
    gsrainv1 = paramij->zeta * rainv1;
    gsrainvsq1 = gsrainv1*rainv1/r1;
    expgsrainv1 = exp(gsrainv1);

    r2 = sqrt(rsq2);
    // Same as the for r1

    rainv12 = 1.0/(r1*r2);
    cs = (delr1[0]*delr2[0] + delr1[1]*delr2[1] + delr1[2]*delr2[2]) * rainv12;
    delcs = cs - paramijk->costheta;
    delcssq = delcs*delcs;
}

```

Listing 8: Three-body part implemented in LAMMPS, for the usc-potential

```

facexp = expgsrainv1*expgsrainv2;

facrad = paramijk->B * facexp*delcssq;
frad1 = facrad*gsrainvsq1;
frad2 = facrad*gsrainvsq2;
facang = 2.0*paramijk->B* facexp*delcs;
facang12 = rinv12*facang;
csfacang = cs*facang;
csfac1 = rinvsq1*csfacang;

fj[0] = delr1[0]*(frad1+csfac1)-delr2[0]*facang12;
fj[1] = delr1[1]*(frad1+csfac1)-delr2[1]*facang12;
fj[2] = delr1[2]*(frad1+csfac1)-delr2[2]*facang12;

csfac2 = rinvsq2*csfacang;

fk[0] = delr2[0]*(frad2+csfac2)-delr1[0]*facang12;
fk[1] = delr2[1]*(frad2+csfac2)-delr1[1]*facang12;
fk[2] = delr2[2]*(frad2+csfac2)-delr1[2]*facang12;

if (eflag) eng = facrad;
}

```

Listing 9: Three-body part implemented in LAMMPS, for the usc-potential

In listing 8, we first declare variables. Then we pre-compute terms from equation 3.18. Note that `paramij` contains the i and j interaction parameters, while `paramik` contains the i and k interactions. Depending on the type of the center atom `paramijk` determine if the ijk configuration has a three-body interaction. For an Si-O-Si interaction, θ_0 is 109.47° , while for an O-Si-Si interaction $\theta_0 = 0$ and $B_{ijk} = 0$, hence no three-body interaction.

7.1.3 Bond Order Scheme

The implementation of the bond order scheme is the most challenging part of the usc-potential, it was challenging because we need to find neighboring atoms of two oxygen pairs in silica or water. This is not trivial when we want it to be computational efficient. A bond order scheme is implemented, but it is still in its preliminary phase. The idea behind our bond order implementation is to create a new neighbor list. This list holds all the oxygen-oxygen interactions. A loop through this list, while scanning for the number of Si and H neighbors we find number of neighbors, Θ . Summing over Θ and using equation 3.21, we find the weight factor f_{ij} .

7.1.4 Assigning Parameters to Elements

For many-body potentials there are often numerous variables to keep track of. Depending on the form of the interaction, if it is a two-body or three-body interaction, and which atom types that are interacting. In the usc-potential we solve this by assigning a data structure `struct` and name it `Param`. It contains a set of parameters which is assigned at the start of the simulation. Every element i and its interactions with j and k needs to be set explicit. For water we need to set eight different types of interaction parameters, for H-H-H, H-O-H etc. This is set in a potential parameter file as the one listed below. Unfortunately, the usc-potential parameters are closed source, so we do not reprint them here.

```
# Water parameters for the usc-potential

#i j k costheta H r1s r4s D w r0 zeta B globalCut cutsq eta

O O O
H H H
O O H
O H H
H O O
H O H
H H O
O H O
```

For more than two atom types, the potential parameter file handling becomes tedious. For three different elements we get 27 entries etc.

7.2 Region

In the theory of fracture mechanics we operate with an elliptic shaped crack. In this way, we can control the direction of the fracture, by adjusting the a-direction of an ellipse. In LAMMPS there are different region commands, together with a delete atoms command, we use this to create the initial fracture. We want to write something like this

```
region 1 ellipse xc yc zc a b c r side in
delete_atoms region 1 mol yes
```

The first integer number is the name of the region. Then we pass to LAMMPS what kind of region we want. `xc`, `yc` and `zc` is the center of the ellipse. While `a`, `b` and `c` are the lengths in the x , y and z -direction. We define a radius `r`, a

`side in` tells LAMMPS to tag the atoms inside this ellipse. From the definition of an ellipse we have

$$\left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 + \left(\frac{z}{c}\right)^2 = 1 \quad (7.1)$$

In the subclass `RegionEllipse` of the `Region` class we write

```
int RegEllipse::inside(double x, double y, double z)
{
    double delx = x - xc;
    double dely = y - yc;
    double delz = z - zc;

    double aa = a*a;
    double bb = b*b;
    double cc = c*c;

    double r = sqrt((delx*delx)/(aa) + (dely*dely)/(bb) + (delz*delz)/(cc));

    if (r <= radius) return 1;
    return 0;
}
```

Here we see how we have changed the radius `r` from a sphere to define an ellipse. Molecules inside the ellipse are tagged and deleted by the `delete atoms` command.

7.3 Debugging LAMMPS

Finding errors and origins of segmentation faults is eased by the use of a debugger. For our implementation we chose to use the `gdb` debugger. It runs on most UNIX like systems and works for several programming languages. A short example of how to use the debugger is outlined below

```
$ gdb lmp_ubuntu
```

This command starts the debugging session. To run the executable we supply an input script in the debug shell.

```
(gdb) run -in in.yourinscript
```

To exit the debugger we type `quit`.

Chapter 8

Simulation Procedure

In this chapter we explain our setup of the various systems and the numerical procedures of our simulations are outlined. We start by introducing crystal structures of portlandite and silica, before we move on to the setup of a nanopore. We discuss how water is inserted in this pore by a voxelation method. To compare our implementation of the usc-potential with other silica potentials, we perform annealing of silica. A heat/cool method for annealing of β -cristobalite to amorphous silica is therefore presented. A method for fluid flow based on acceleration driven flow is presented. We also describe how to strain a molecular system to estimate elasticity constants.

8.1 Crystal Structures

Portlandite $\text{Ca}(\text{OH})_2$, is a layered clay-mineral consisting of calcium, oxygen and hydrogen. It is not one of the most abundant clay-minerals, but we chose portlandite due to its low number of atoms, compared to other clay types. Because of the relative high content of portlandite in hydrated cement, it is interesting to study water migration and elastic properties of this mineral. Another reason was that the low complexity eased the implementation of a reactive ClayFF, keeping new parameters to a minimum. The usc-potential is a silica-water potential, constructed in such a way that we obtain the tetrahedral structure of water and silica. Amorphous silica was compared and annealed with the usc and ClayFF-potential, the crystal structure of β -cristobalite is therefore outlined.

8.1.1 Portlandite

As mentioned portlandite is a layered clay mineral, low on constituents compared to other clay minerals. The experimental structural characterization of clay is limited, due to the ultrafine grain size of these phases. Hydroxide phases are even harder to characterize experimentally, and explains why numerical simula-

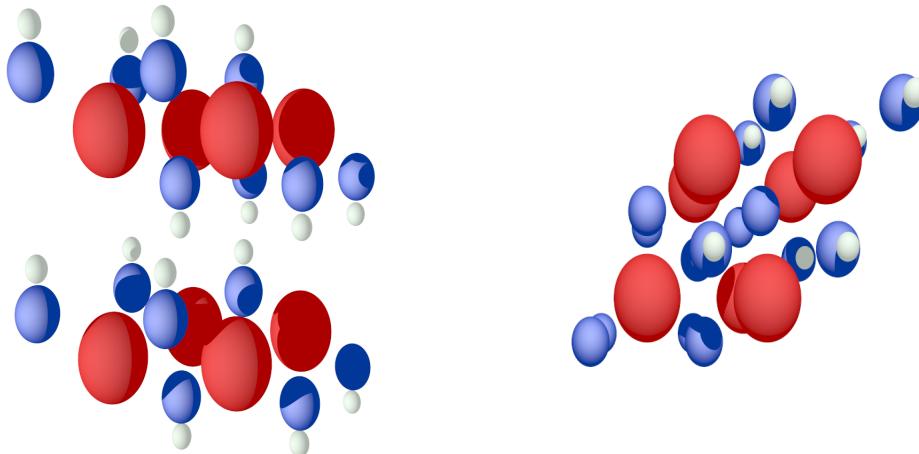


Figure 8.1: Left figure side, view of portlandite unit cell. Right image top view of portlandite unit cell. Crystal structures from [13]

tions are becoming more and more popular to determine structural and interface properties of clay-minerals. Even though the portlandite unit cell is small compared to other clays, it consists of 40-atoms. Portlandite is a transverse isotropic material, meaning that it is isotropic in one plane, but anisotropic normal to the plane of isotropy. In figure 8.1, a side view and a top view of the portlandite ($\text{Ca}(\text{OH})_2$) unit cell is depicted. There are four red calcium atoms in the 001 plane at different heights z . They share four hydroxide molecules with the other calcium atoms. Two of the calcium-oxygen bonds are covalent, and handled with an angle term, in ClayFF. The two other nearest hydroxyl oxygen atoms are not bonded. Hydroxide interaction, OH, is handled by a static harmonic bond term in ClayFF. The interlayer hydrogen atoms form hydrogen bonds with the adjacent clay layers.

8.1.2 Silica

Quartz is one of the most abundant minerals found on earth. It comes in different structures, α -quartz, β -quartz, α -cristobalite or β -cristobalite. In our simulation we use the cubic β -cristobalite (from now named b-cristobalite) structure and quenched this to arrive at an amorphous state, often called vitreous silica or a-silica. The melt quench method is described, so is the initial and final structure of silica.

Crystal Structure

The unit cell of b-cristobalite is cubic and has a lattice constant $a = 7.16 \text{ \AA}$. From this and the position of the oxygen-atom relative to the silica-atom, we can build our system with LAMMPS. The b-cristobalite is a tetragonal structure, with

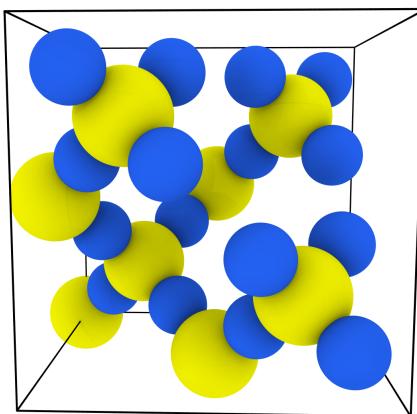


Figure 8.2: b-cristobalite unit cell. Unit cell structure from [72]

four oxygen atoms surrounding one silica atom. Its unit cell structure consists of 8 silica atoms and 16 oxygen atoms, figure 8.2. When silica is heated to its melting point and then cooled, we get a system of amorphous silica. This system is isotropic, arranged in a tetrahedral structure, one silica atom surrounded by four oxygen atoms.

Annealing Silica

We want to compare structural properties of amorphous silica, the usc and ClayFF parameters for silica is then applied. In order to create amorphous silica we have to use a melt and quench method. One stepwise procedure is found in [70]. The method in this paper is time-consuming, from experience we know that a simple heating and melting procedure produce satisfying results. This method consists of gradually heating the b-cristobalite until we reach $T_{\text{anneal}} = 5000 \text{ K}$, and SiO_2 appeared in a liquid phase. Then the annealing was reversed, lowering the temperature until we reached 300 K. For the ClayFF potential the heating and melting can be summarized as

- An energy minimum of the b-cristobalite is found by Newtons method in LAMMPS.
- The system is heated from 300 K to 1000 K over a period of 100 ps.
- The time step is 0.5 fs and the pressure P is set to 0 atm.
- We relax the system for 100 ps, at 1000 K
- Then the system is heated to 2000 K, followed by another 100 ps relaxation.

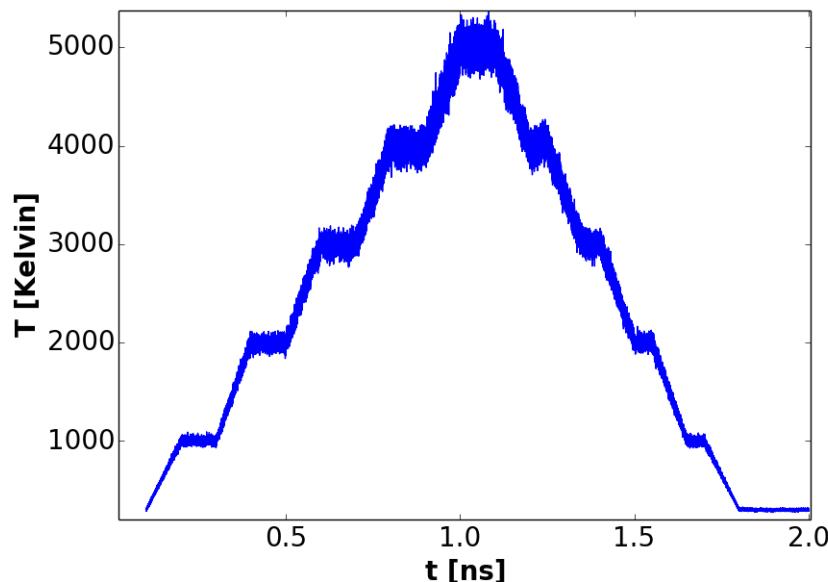


Figure 8.3: Temperature profile showing the annealing procedure for obtaining amorphous silica.

- This procedure is applied until we reach 5000 K, then we reverse the steps until we reach 300 K.

The same steps are performed with the usc-potential, with a time step of 0.05 fs and keeping the volume constant. In figure 8.3, the heating and cooling of the b-cristobalite is illustrated for the usc-potential. The time step used here was 0.05 fs, causing it to be a long lasting procedure. Note larger temperature fluctuations for higher temperatures. This is most likely because of the liquid state of the silica. This allows for more translational movement, compared to lower temperatures where the silica solidifies.

8.2 Constructing a Nanopore

The portlandite slab is created from a unit cell using the open source molecular builder, *moltemplate*¹. Two slabs of equal size are constructed and placed at a height H apart. Leaving room for the insertion of water molecules, see figure 8.4.

For bulk water systems without any other constituents, placing water molecules is trivial. For a nanoporous system though, this is not that easy. Luckily we are not considering complex pore geometries, because of the layered properties of clay. Here we explain the voxelation method we used to get correct water density in the pore. The method outlined below is adopted by a description in [59].

¹<http://www.moltemplate.org/index.html>

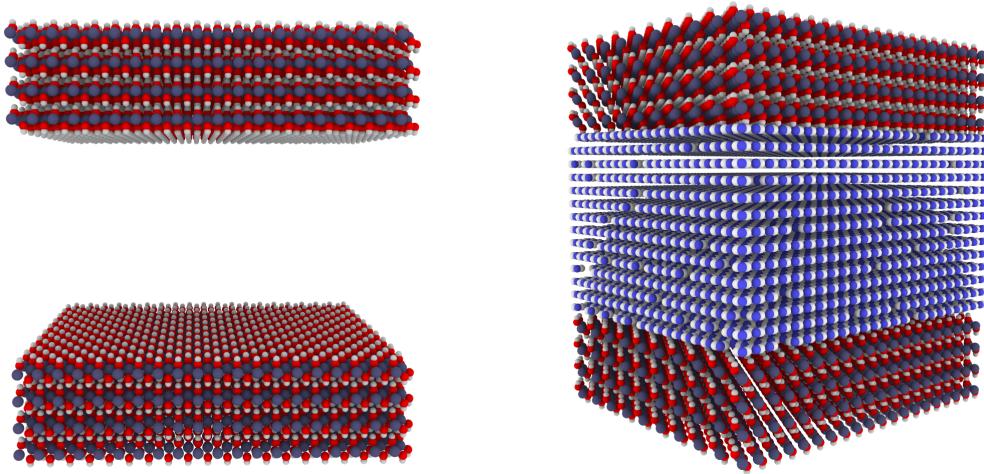


Figure 8.4: (a) A 4nm portlandite clay pore. To insert water with correct density inside the pore we follow a voxelation method. (b) Water is distributed in the voxels to obtain the wanted density.

Finding Correct Voxel Size

The volume between the two slabs of clay is given as

$$V_{\text{pore}} = L_x L_y L_z ,$$

where L_x and L_y are the system sizes, while L_z is the distance between the inner surface of the clay slabs. We want to find how many water molecules to insert, so we need the “volume” of a water molecule. Our voxel size is defined from this volume, given as

$$V_{\text{water}} = \frac{M[\text{g/mol}] N/N_A [\text{mol}^{-1}]}{\rho[\text{g/cm}^3]} = \frac{M}{\rho N_A} [\text{cm}^3] .$$

For one water molecule we obtain

$$l = \left(\frac{M}{\rho N_A} \right)^{1/3} [\text{cm}] .$$

Which gives us a voxel size

$$l = \left[\frac{18.001[\text{g/mol}]}{0.999[\text{g/cm}^3] \times 6.022 \times 10^{23}[\text{mol}^{-1}]} \right]^{1/3} = 3.1\text{\AA} . \quad (8.1)$$

In figure 8.4, we see the clay system before we insert water. The total volume between the two parallel slabs are found, and applied when we insert water molecules. The inserted water is seen to the right in the figure. Showing that some of the voxels are empty to get the correct density.

Voxels to Fill

Our pore is finite, so inserting voxels of size 3.1 \AA does not always give you the correct density. Since the system is divided into an integer number of voxels, we have to make sure not to fill all of them. Because we do not want to low density we *ceil*, the number of voxels given by $n_x = L_x/l$, so filling all the voxels would yield to high density. First we find the number of voxels filling the system if they were of the size given by equation 8.1. Then we find the actual size of the voxels \tilde{l} . The total number of voxels N , is now given as

$$N = \frac{V}{V_{\text{vox}}} .$$

How many voxels we want to fill is given by

$$\tilde{N} = N \frac{\tilde{l}^3}{l^3} .$$

We fill the voxels by looping through all voxels N , and filling them if a random number between 0 and 1, is smaller than \tilde{N}/N . The oxygen atom is inserted at the center of the voxel, while one hydrogen atom is placed 1 \AA from the oxygen, and the last hydrogen placed to make sure of an H-O-H angle of 109.47° .

8.3 Initiating Flow

In a nanoporous systems continuum hydrodynamics is known to break down. Due to the small size of such systems surface forces influences the dynamics of the water more than for larger systems. We study these effects using molecular dynamics, the results can be used as input parameters for coarse grain models, such as dissipative particle dynamics and lattice Boltzmann methods. These models are capable of modeling porous flow at a larger length scale.

To study hydrodynamics in nanoporous rocks we need a method to initiate flow. In molecular dynamics this is not trivial, due to the discrete nature of atoms. Several methods are proposed and explained in chapter 5.3. Here we give you the method used in this theses and the input parameters used in a LAMMPS script.

The initial system is a 4 nm pore, with water inserted using the voxelation method. First we equilibrate the system at 300 K , under constant volume conditions. We want to obtain a constant energy and temperature term, before we initiate flow. The thermostat is now coupled in x, y and z-direction. The system is also periodic in all three directions. We do not update the positions of the atoms in the clay matrix, liquid/solid forces are calculated, but portlandite positions are not updated. By keeping atoms in the portlandite stationary, we

avoid tearing the portlandite apart with the flow. Other similar studies follow the same procedure [4].

When the system is equilibrated, we initiate a flow of 10 m/s. This corresponds to a velocity of

$$\begin{aligned} 10[\text{m/s}] &= 10 \cdot \frac{10^{10}}{10^{15}} [\text{\AA/fs}] \\ &= 10^{-4} [\text{\AA/fs}]. \end{aligned}$$

To not interfere and slow particles down, thermostatting is only performed in the y-direction, perpendicular to the flow.

```
compute myTemp spc temp/partial 0 1 0 # fix-id, group-id, name-fix x y z
```

The command for adding a constant force is given as

```
fix flow spc addforce 0.00075 0.0 0.0
```

This fix adds a constant force to atoms in the `spc` group.

8.4 Elasticity

In molecular dynamics there are two different ways to find the elasticity constants. We can obtain the analytical expression for the second derivative of the potential function, or we can perform a numerical simulation. In the former method Young's modulus, Poisson's ratio and shear strength is found at zero kelvins. In the latter, we apply strain in different directions and measure the corresponding stress, this can be done at a finite temperature. We applied strains and measured the stress for different sizes of bulk portlandite.

8.4.1 Applying Strain in Molecular Dynamics

As we saw in chapter 4.3.4, we get the stress in different directions directly from the LAMMPS software. The pressure tensor is calculated as the virial from statistical mechanics, and by applying a strain in one direction we can find the resulting stress in the same direction. Performing elongations and compressions is not trivial in molecular dynamics, because there are different properties of a system you want to keep constant, e.g. pressure, energy and volume. We strain the system by elongating the simulation box, to avoid high stresses on the

outermost atoms, we remap the position for all the atoms. By doing this we are stretching the whole system, and not only atoms at the edge. Then we let the atoms wiggle into place reaching energy equilibrium, before we strain the box again. Here is a stepwise outline of the method used in this thesis.

- The box is proper relaxed in the NPT-ensemble with no external pressure, $P = 0$. This is done to make sure there are no internal stresses in the system
- We strain the system by extending the simulation box in a wanted direction.
- The atoms are mapped to their new positions.

The problem in molecular dynamics is the short time-scales, the applied strain is in orders of a nuclear explosion. This is not realistic, and probably there is a strain rate dependence as well. The advantage is that this method can be used at any temperature, to calculate the stiffness parameters of the material. Our strain rate is 0.001 /ps, a lower value than [41] use for their stress-strain curves. A way to solve the strain rate problem is to strain and relax the system for a longer amount of time. But then we need more computational power, and still we might be far off a realistic strain rate. The method for straining a simulation box in LAMMPS is given as

```
fix    1 all npt temp 300.0 300.0 1000 x 0 0 1000.0 z 0 0 1000.0
fix    2 all deform 1 y erate 0.000001 remap x
```

Pressure P is now only controlled in the directions not subject to a strain. Atoms are remapped according to the given strain rate, by the remap command.

Creating a Fracture

Fracture simulations were carried out with a $10 \times 10 \times 1$ unit cell slab of portlandite. By using our implementation of the `region ellipse` command in LAMMPS, we marked atoms in the center of the slab. Setting $a = 3, b = 1$ and $c = 1$ in the equation of an ellipse. The radius of the ellipse is set to 4 Å. The crack is open in the z-direction of the slab, leaving us with a 2D-simulation, because of the periodicity. Deletion of atoms were done so that whole molecules were removed, so there was no need to reconstruct the surface. ReaxFF does not differ between atoms belong to which molecule. The fracture is therefore cut before the simulation starts; the system with the fracture is then read as an input script for the ReaxFF simulation.

8.4.2 Note on Elasticity Simulations

Fracturing of clays involve bond breaking. In the portlandite covalent bonds between calcium and oxygen are broken. ClayFF does not model this bond breaking, and some assumptions are made. One of these is that the charge of an atom remains constant during a strain. This is in general not true, since some atoms are more electronegative than others. As mentioned before, this is where the bond order potentials come in handy, since they weight the interaction depending on neighbor atoms. Examples of this are the Tersoff, Brenner and usc-potential, still though, these potentials does not consist of parameters for clays. We justify our ClayFF simulations, by the fact that there have been several successful studies of fracture in amorphous silica with simple two-body potentials [41]. Normal to the clay layers there are only weaker hydrogen bonds, so in this direction fracture modeling should work fine with ClayFF. Muraldiharan et al. [41] proposed a charge transfer function which accounted for the distance between interacting pairs to compute q . They found that the slope of the stress-strain curve remains the same. The peak stress only changes by a constant factor.

A possibility would be to use the `fix qeq` command in LAMMPS, to calculate charges on the fly. This is computational expensive and the parameter settings are not adjusted to clay. Setting `qeq` parameters is described as an *art* by LAMMPS developers. We suggest a direct comparison between ReaxFF and ClayFF using the assumptions discussed above, by doing this we will get a better knowledge of the limitations of non-reactive potentials.

Part III

Numerical Simulations and Results

Chapter 9

Structural Properties

We report structural properties, such as radial distribution function, density, angle orientation and tetrahedral order parameter. This is done for bulk water, portlandite amorphous silica, and a nanopore. Even simple structural properties as the density can give us an indication of the reliability of our numerical simulations. Hence we see this chapter as a valuable validation of our numerical simulations. In the following chapters we will first introduce and describe a method, followed by the results found from this method. Then on to a new method and its result.

9.1 Radial Distribution Function

The radial distribution function $g(r)$, comes in handy because it gives us a direct comparison between experiments and a model. In neutron and X-ray scattering we obtain precisely $g(r)$, and numerical it is not that hard to find either. Here we briefly explain the radial distribution function. We have implemented the radial pair distribution function, but for our analysis we used the built in $g(r)$, in the VMD software [27].

In a numerical simulation we find the number density $\rho_n(r)$, at a distance r , from the atom we are considering. Then we take the ratio between ρ_n and the ideal gas value. For an ideal gas there are no interactions, so we have $g(r) = 1$. A bump in $g(r)$ is then a deviation from unity, and indicates interactions between atoms. If $g(r) = 0$ it indicates a repulsive interaction, since there are no atoms within the distance r . Radial distribution functions also provides bond lengths in molecules, a peak is therefore often evidence of a bonded interaction. A bond length can then be read out of a $g(r)$ plot.

Table 9.1: Oxygen-Oxygen Pair Distribution Functions for All Water Models Using 1000 Molecules Simulations

water model	first max position		second max position		third max position	
	(Å)	g_{OO}	(Å)	g_{OO}	(Å)	g_{OO}
SPC	2.75	2.74	4.55	1.05	6.85	1.04
SPC/E	2.75	3.03	4.55	1.12	6.85	1.04
usc	3.05	2.11	4.75	0.91	6.35	1.19
exptl [55]	2.88	3.09	4.50	1.14	6.73	1.07

Table 9.2: Oxygen-Hydrogen Pair Distribution Functions for All Water Models Using 1000 Molecules Simulations

water model	first max position		second max position		third max position	
	(Å)	g_{OH}	(Å)	g_{OH}	(Å)	g_{OH}
SPC	0.95	13.16	1.8	1.31	3.25	1.52
SPC/E	0.95	13.32	1.75	1.56	3.25	1.56
usc	0.95	23.26	2.05	0.77	3.45	1.53
exptl [55]			1.85	1.38	3.30	1.60

9.1.1 Measurements of $g(r)$

Here we compare the radial pair distribution function $g(r)$, for the SPC, SPC/E and usc water models. We determine the H-O-H angle for the three-body usc-potential, and compare it with the constant angle potential ClayFF.

Figure 9.1, compares $g(r)$ for the three water models SPC, SPC/E and usc. As expected the SPC and SPC/E plots show similar behavior, since the two models only differ by the point charge strength. The dashed line is the $g(r)$ for our implementation of the usc-potential. We do not know of any published papers that plot these values, so we compare with the SPC-models. However, the usc-potential is a many-body potential so they are not direct comparable.

The H-H distribution for the usc-model in 9.1a lacks the dip at around 3 Å. For the O-H distribution 9.1b, we see that the second peak for the dashed line is a bit lower than the SPC models. Since the usc-model allows for dissociation of hydrogen from oxygen, a lower value for the O-H peak is reasonable. The oxygen-oxygen interaction 9.1c, illustrates how the SPC/E model interacts over a longer distance because of the increased charge setting. The dashed line for the usc-potential also shows a interaction for a longer distance than the two SPC-models. In figure 9.1d, the dashed lines are the angle distribution of the usc-potential, while the solid line is the angle distribution of the SPC-model. Here we clearly see how a many-body potential differs from a static potential. The angles are

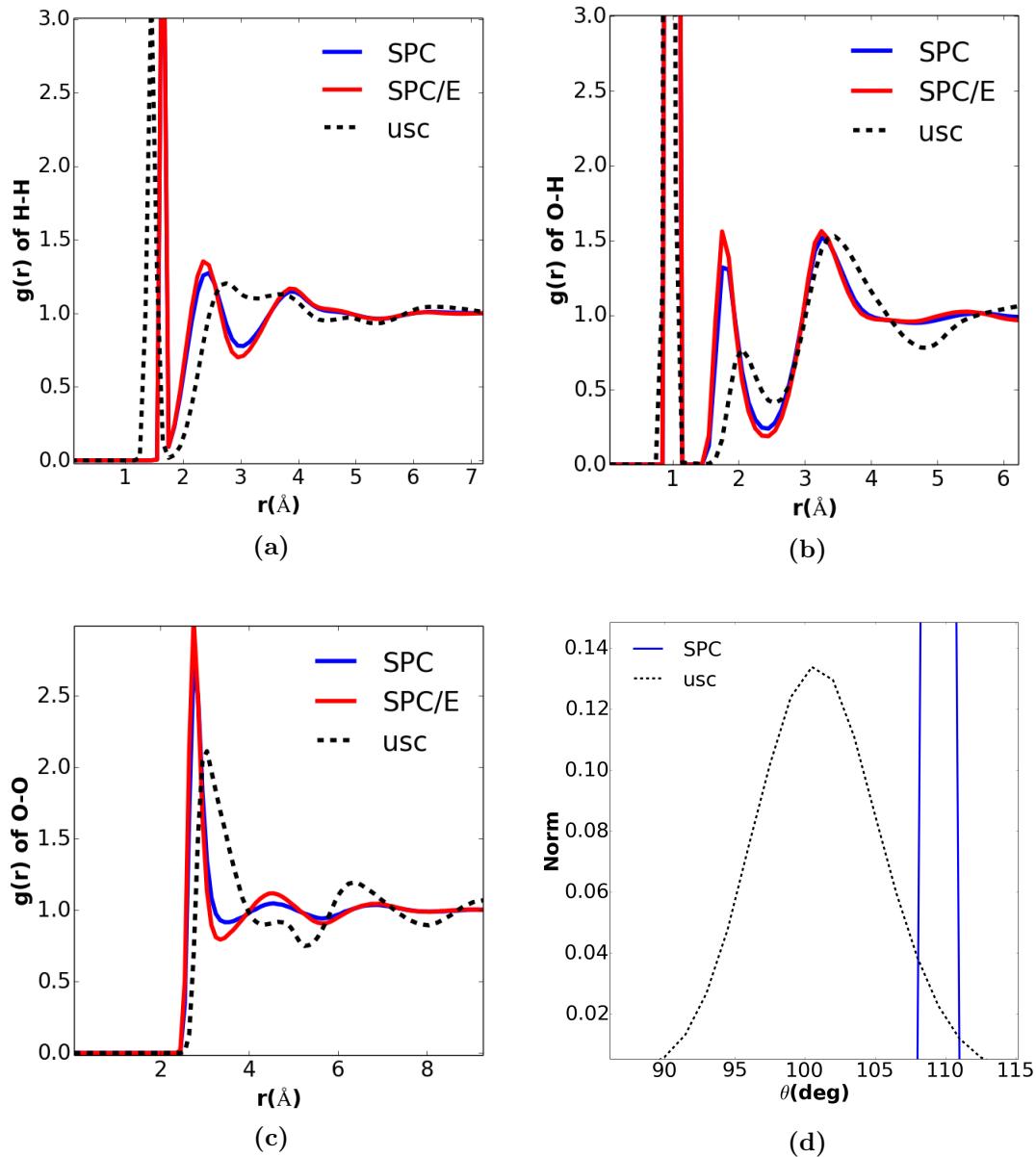


Figure 9.1: (a)-(c) Pair distribution function for HH, OH and OO from our MD simulation. Comparing SPC, SPC/E and the usc-water model. (d) Is the angle distribution for the usc-model and SPC-model.

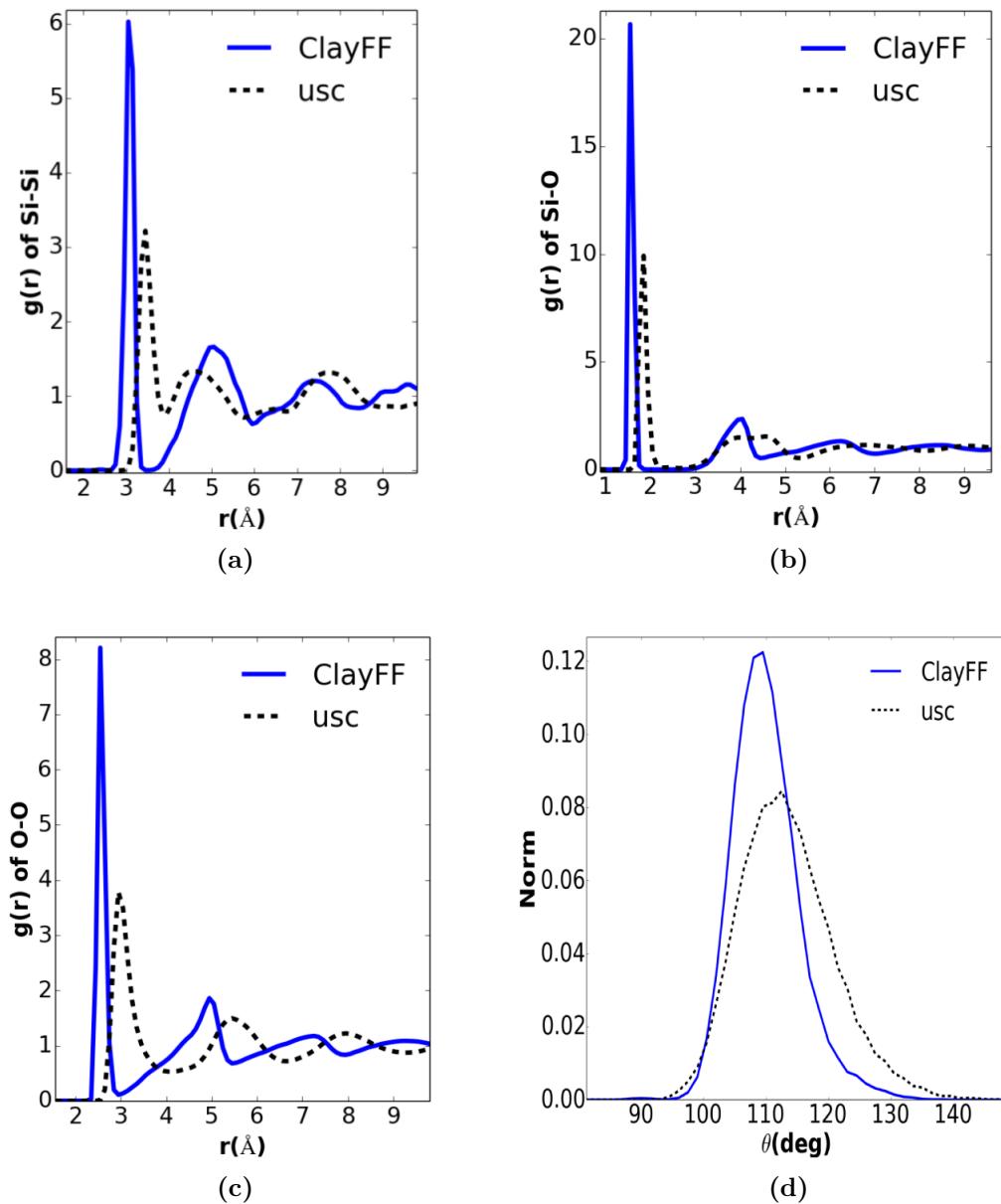


Figure 9.2: (a)-(c) Pair distribution function for Si-Si, Si-O and O-O from our MD simulation. Comparing the ClayFF and usc-model. (d) Angle distribution of O-Si-O angles for the ClayFF and usc-model.

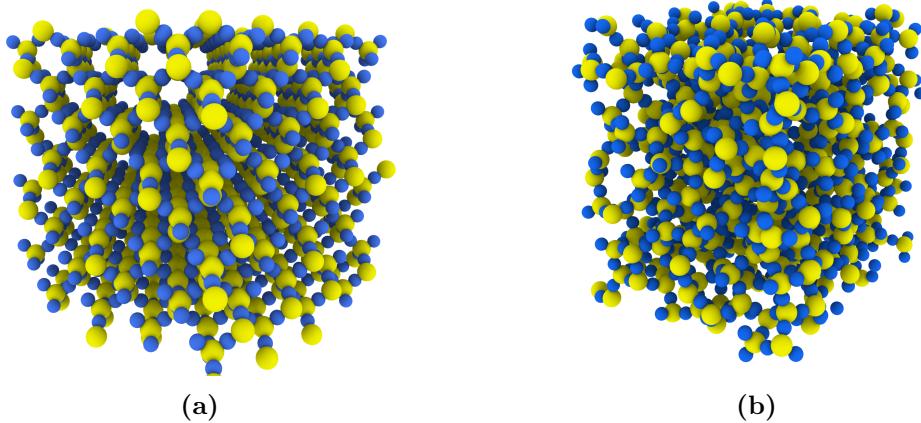


Figure 9.3: (a) b-cristobalite system. (b) a-silica system after the melt and cool procedure, here illustrated with the ClayFF potential.

much more distributed around θ_0 . Our results for $g(r)$, and angle distribution are in good correspondence with an unpublished article from the developers of the usc-potential.

In table 9.1, we have listed the O-O distribution for the three water models. The O-H bond distance is found in table 9.2, here we have also listed the second and third peak value of the radial pair distribution function. From the tables 9.2 and 9.1, we note a satisfactory result for the usc-model. The peak of $g(r)$, for O-H distance is a bit high, we think the high peak is a result of many-body potentials allowing for more than two hydrogen atoms bonded to an oxygen atom.

Originally the usc-potential was meant to model silica. We therefore wanted to test our implementation of the potential versus ClayFF with silica parameters. We measure the distribution function for bulk a-silica, for our two models and the angle distribution between O-Si-O of the two models. We find the mean of the angle distribution giving, $\theta_{\text{usc}} = 112.6^\circ$ and $\theta_{\text{ClayFF}} = 109.3^\circ$. ClayFF, being a two-body potential actually shows a better correlation with the values (109.6°) obtained by Vashistha, for the O-Si-O angle [70]. The radial distribution function for silica is given in 9.2. Again the main features are equal, but we have to note that we compare a two-body potential with a many-body potential.

From figure 9.3, we see the structure of a b-cristobalite system 9.3a, before we heated and cooled the system to get a-silica 9.3b. It is interesting to note how a two-body potential like ClayFF preserves the angle distribution among Si and O atoms.

9.2 Density

When simulating bulk water and water inside a nanopore, we want to start out with a density close to the density of bulk water, $\rho_w = 0.99 \text{ g/cm}^3$. The insertion of water in the system can be done in different ways. The best way is to perform a grand canonical Monte Carlo simulation. In a GCMC simulation we set the chemical potential of an outside reservoir, and fill the pore by performing Monte Carlo insertions of water molecules. In LAMMPS, the `fix gcmc` command is able to do this for simple interatomic potentials. But it is not constructed for many-body potentials and long range electrostatic forces. Our solution to this problem was stated in chapter 8.2, where we argued for the voxelation method.

The density was calculated for bulk systems of water, portlandite, a-silica and water confined in clay nanopores. For the latter we also find the water density as a function of distance from the clay matrix. Density calculations are straightforward, and computed in LAMMPS by setting the system size and mass of the different atom types.

9.2.1 Water and Clay Density

Here we report density of bulk water, portlandite and a-silica, the density of water confined in a nanopore is also found. The densities that are given, are averaged over several time steps and in an equilibrium state of the system.

Table 9.3: Initial and equilibrium density of the atomic systems.

System	Eq.density[g/cm ³]	Reported[g/cm ³]
SPC _{bulk} 1000	0.99	0.998 ^[36]
SPC/E _{bulk} 1000	0.986	0.998 ^[36]
usc _{bulk} 1000	0.994	
Portlandite	2.233	2.239 ± 0.032 ^[13]
a-Silica _{ClayFF}	2.143	2.15 ^[6]
a-Silica _{usc}	2.174	
3nm Pore	1.06 ± 0.08	
4nm Pore	0.987 ± 0.11	
6nm Pore	1.01 ± 0.11	

We notice that when we run a NPT-simulation the filling density can be quite low and still we reach a reasonable equilibrium density for bulk water. The system would need more time to approach the correct density, therefore it is desirable to initiate the system with a more correct density and run NVT-ensemble. For

the SPC/E and usc models we ran the whole simulation at a constant volume. The two solids, portlandite and a-silica where equilibrated in the NPT-ensemble before we measured the density. Table 9.3 reports the density for our systems and compare with known results from literature. We find the density of bulk water in the center of the nanopores to be a bit higher than for bulk water, indicating to high filling density in the pores for our simulations.

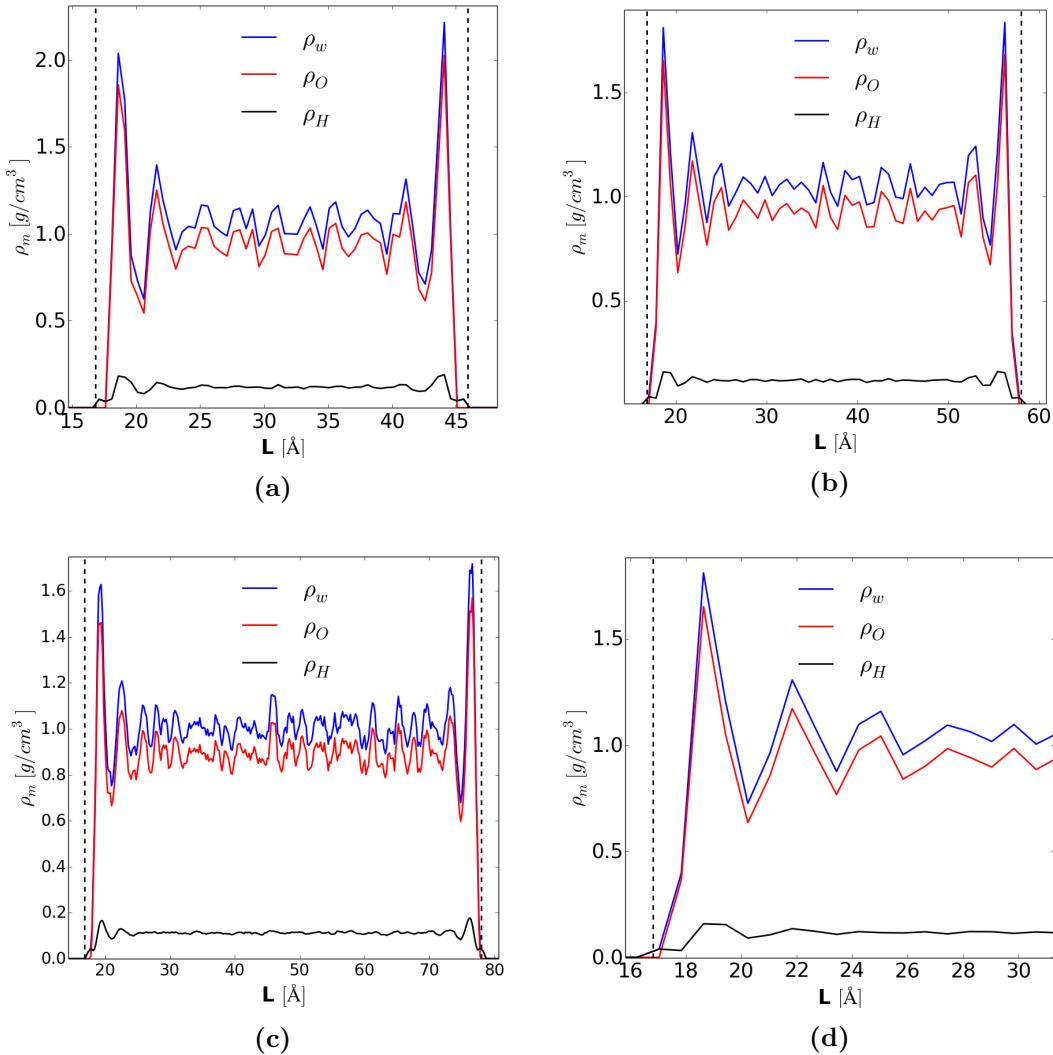


Figure 9.4: (a)-(c) The mass density across a 3-nm, 4-nm and 6-nm pore.
 (d) Zoomed in on the 4 nm pore.

We found the bulk density of water inside the 3 nm, 4 nm and 6 nm pore by averaging the values in the center of the pores figure 9.4. The values we find are within one standard deviation from the bulk water density of the SPC-model

reported in [36]. In figure 9.4d we have zoomed in on the density distribution for the 4 nm pore. The vertical dashed line is the position of the surface hydrogen atoms. The fluctuations decay after 8 Å, where we get an average bulk water density. Surface hydrogen atoms in the portlandite and oxygen in the water is attracted by each other, creating a higher density at the surface. We suggest that the portlandite surface is hydrophilic, because it is wetted by the water.

9.3 Tetrahedral Order Parameter

The tetrahedral order parameter measures the order of tetrahedrallity of water. The order of tetrahedrallity tells us how tetrahedral a set of water molecules are, see figure 9.5. For water it is known that for lower temperatures, the value of the parameter increases towards 1. This means that water with a low temperature is more structured than water with a high temperature. We don't only look into this, but we want to see if this parameter is affected by the clay surface, or if its keeps its bulk value. The tetrahedral order parameters is mathematical defined as

$$Q_k = 1 - \frac{3}{8} \sum_i^3 \sum_{j=i+1}^4 \left[\cos \psi_{ijk} + \frac{1}{3} \right]^2, \quad (9.1)$$

where ψ_{ijk} is the angle formed by the molecule k , and its nearest neighbors i and j . We measured the tetrahedral order parameter for bulk water and for water confined in nanopores.

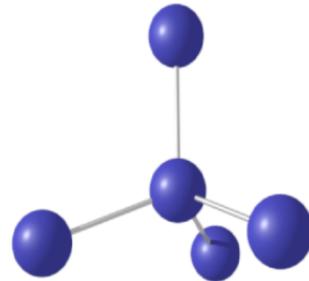


Figure 9.5: Perfect tetrahedral network formed by five atoms. The crystal structure acted as an unit test for our implementation of equation 9.1. The image was rendered using povray ray tracer

9.3.1 Bulk Water Order Parameter

Finding a bulk water value for the tetrahedral order parameter is done for validation against values found in the pore. We found the order parameter for three

different system sizes, and for three different water models.

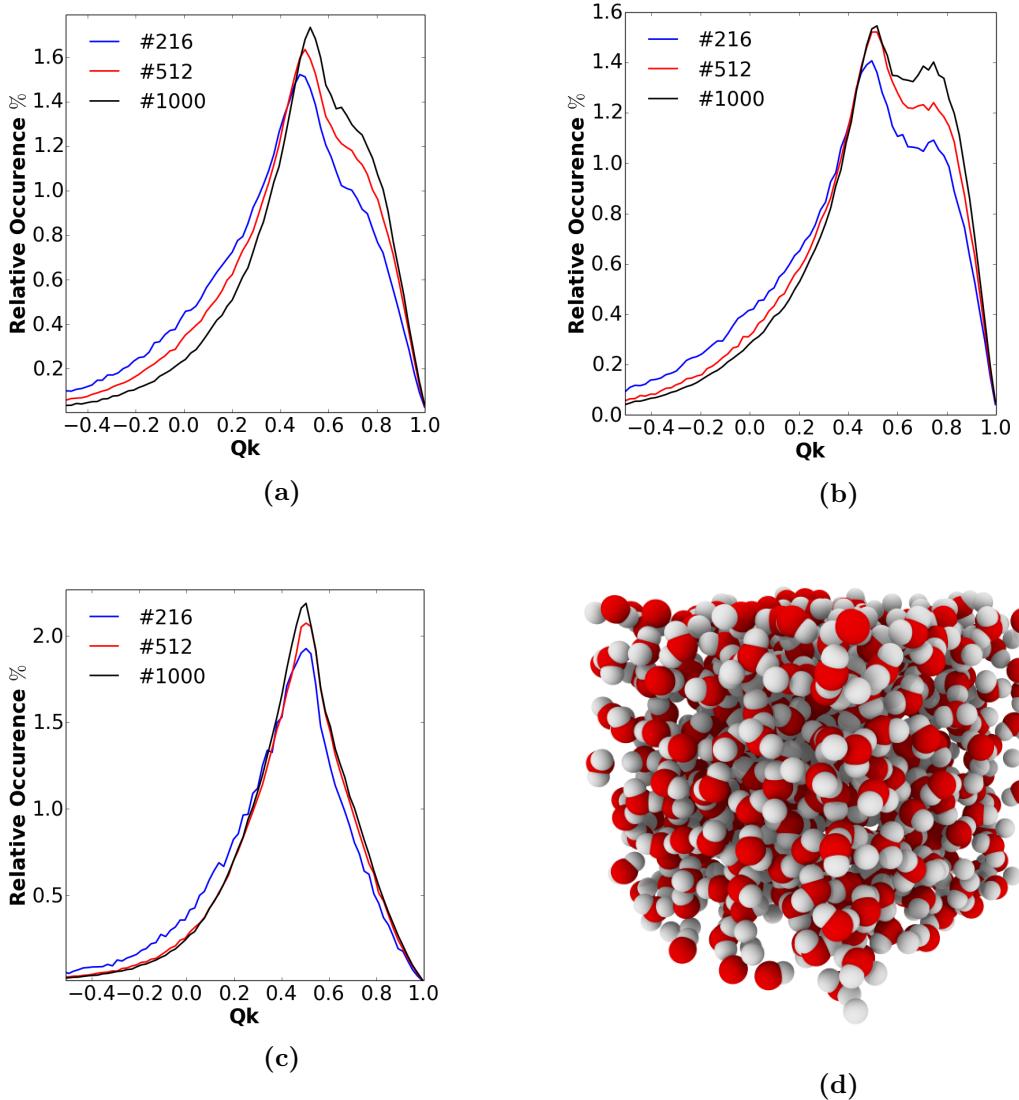


Figure 9.6: (a)-(c) Tetrahedral order parameter for three different water models, (a) SPC-model, (b) SPC/E-model and (c) usc-model. We clearly observe how the SPC/E model (b) is more tetrahedral than the other two modes, from the peak at ≈ 0.75 . What is surprising is the lack of this peak at all for the usc-water model which is supposed to favor tetrahedral ordering from its dipole-dipole interactions. (d) Bulk water system, SPC/E-model.

We report the average distribution and peak value, for the 1000 water molecules system. For the usc-model $Q_k = 0.33$, SPC-model $Q_k = 0.30$ and SPC/E-model $Q_k = 0.32$, is the average values. The peak values for the highest peak are $Q_{\text{usc}} = 0.51$, $Q_{\text{SPC}} = 0.55$ and $Q_{\text{SPC/E}} = 0.53$. In the SPC/E plot, the second

peak value is $Q_{\text{second peak}} = 0.75$. In [10] they report a bulk phase water tetrahedral order parameter average of $Q_k = 0.433$, for the SPC/E. Their values are however based on a perfect tetrahedral value of $5^{1/2}/3$, instead of unity as in equation 9.5. Kumar et al. [32] found the average order parameter for the TIP5P water potential at 300 K to be ≈ 0.4 . Their peak Q values corresponds with our values.

9.3.2 Pore Order Parameter

When inside a pore, we expect the order parameter to be dependent on the distance from the pore wall. This is due to the interactions between the water and the clay. We measure the tetrahedral order parameter at the surface, 4 Å away from the surface and in bulk, for the three different systems.

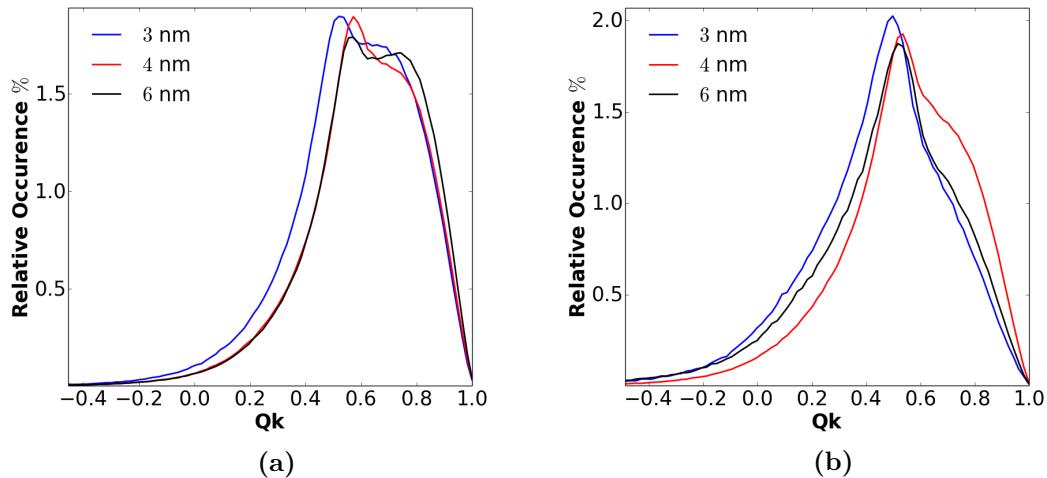


Figure 9.7: (a) Tetrahedral order parameter distribution for bulk in the three pores. (b) TOP at the surface of the same pores.

The bulk average values were as follows, 3 nm pore $Q_k = 0.39$, 4 nm pore $Q_k = 0.34$ and $Q_k = 0.35$ for the 6 nm pore. The peak values for the highest peak where, $Q = 0.52$, 0.57 and $Q = 0.57$, for the 3, 4 and 6 nm pores, respectively. For the 6 nm pore we observe a second peak at, $Q = 0.74$. While for the 3 nm pore, we see a plateau and no significant second peak. In the 4 nm pore the plateau is steeper and there is only a vague sign of a second peak.

We find the average value for the surface order parameter. For the 3 nm pore $Q_k = 0.36$, for the 4 nm pore $Q_k = 0.34$ and for the 6 nm pore $Q_k = 0.34$. The max values are found from the highest peak and the corresponding x-value gave, $Q = 0.5$, $Q = 0.54$ and $Q = 0.52$ for the 3, 4 and 6 nm pore, respectively.

Chapter 10

Diffusion and Bulk Viscosity

Diffusion is the theory behind seemingly random movement of particles at a small scale and explains the mixing of milk in your cup of coffee, or more important the diffusion of oxygen from our lungs into our blood, keeping us alive. In industry enhanced or for the sake of matter less diffusion is important, in mixing or separation procedures. There are a lot of applications and advantages of studying the process behind diffusion. In desalination, we are interested in a high flux of water through a membrane. Research on diffusion through nanoporous membranes can be conducted to optimize flux rates. Enhancing water flux through these membranes can reduce the cost of e.g. desalination processes.

A liquids resistance to external stress is called viscosity. If the liquid is syrup it has a high viscosity, while water on the other hand has a lower viscosity. Again, studying viscosity of a fluid through nanopores and how we can change this viscosity, has applications from oil and gas extraction to desalination processes.

In this chapter we study diffusion of bulk water and water confined in nanopores. Viscosity is calculated for bulk water, giving us a number to compare the later estimation of shear viscosity of water inside pores.

10.1 Diffusion

When we measure diffusion, we measure the self-diffusion. This is a measure of the diffusive behavior of a system, by averaging over a number of particles. The diffusion coefficient D , is an important dynamical property of a water model. In the macroscopic world diffusion is described by Fick's law,

$$j = -D\nabla_c$$

where D , is the proportionality constant also referred to as diffusion coefficient, j is the flux and ∇_c , is the gradient of the concentration. We want to find the self-diffusion, by self-diffusion we mean the diffusion of labeled species among identical solvent. So we measure the diffusion of one molecule in a reference

system from where that molecule was initially placed. Then we average the movement of all the other molecules to find the diffusion constant through the relation

$$\frac{\partial \langle r^2(t) \rangle}{\partial t} = 2dD, \quad (10.1)$$

which is the Einstein relation. Here d is the dimensionality and D is the diffusion coefficient. The left part of equation 10.1, is the mean squared displacement, and can easily be measured in molecular dynamics. For N particles/molecules we can measure the mean squared displacement from

$$\langle \Delta r(t)^2 \rangle = \frac{1}{N} \sum_{i=1}^N \Delta r_i(t)^2,$$

We can plot the mean squared displacement $\langle \Delta r(t)^2 \rangle$ versus the period we measured the msd. From the slope of this graph we find the diffusion coefficient. There are three important things to remember when measuring the diffusion

- The system size is finite, this affects the msd graph
- One should sample D from various origins, since where you start your sampling should have no influence on your result.
- There will be a ballistic regime in the msd of the molecule. This needs to be taken care of when you measure the slope.

Because D varies with the position of the diffusive particle we cannot use equation 10.1, without considering that the particle need to be inside a diffusive regime, and not to long so that the particle probes to areas with other D-values [6]. On this basis, we calculate our diffusion constant within a time-regime of 5–10 ps. This should be sufficiently long for the particle to be in the diffusive regime, and not to long for particles to move into areas with other D-values.

The diffusion was calculated for bulk water, 3, 4 and 6 nm pore. We find bulk water values for comparison with literature, and as a reference point for the nanopore diffusion. Bulk water diffusion for the SPC, SPC/E and usc-model is calculated for different system sizes L . In the porous systems we found D_z and D_{xy} , as a function of distance from the pore wall. Keep in mind that this alters the expression 10.1. Because of the dimensionality variable d .

10.1.1 Bulk Water Diffusion

The bulk water diffusion is calculated for the SPC, SPC/E and the usc-model. A value for D is computed from the mean square displacement graph, such as figure 10.1a. By computing several msd values we improve our statistics. Figure 10.1d is the extrapolated value of D , when $1/L \rightarrow 0$. For each mean square

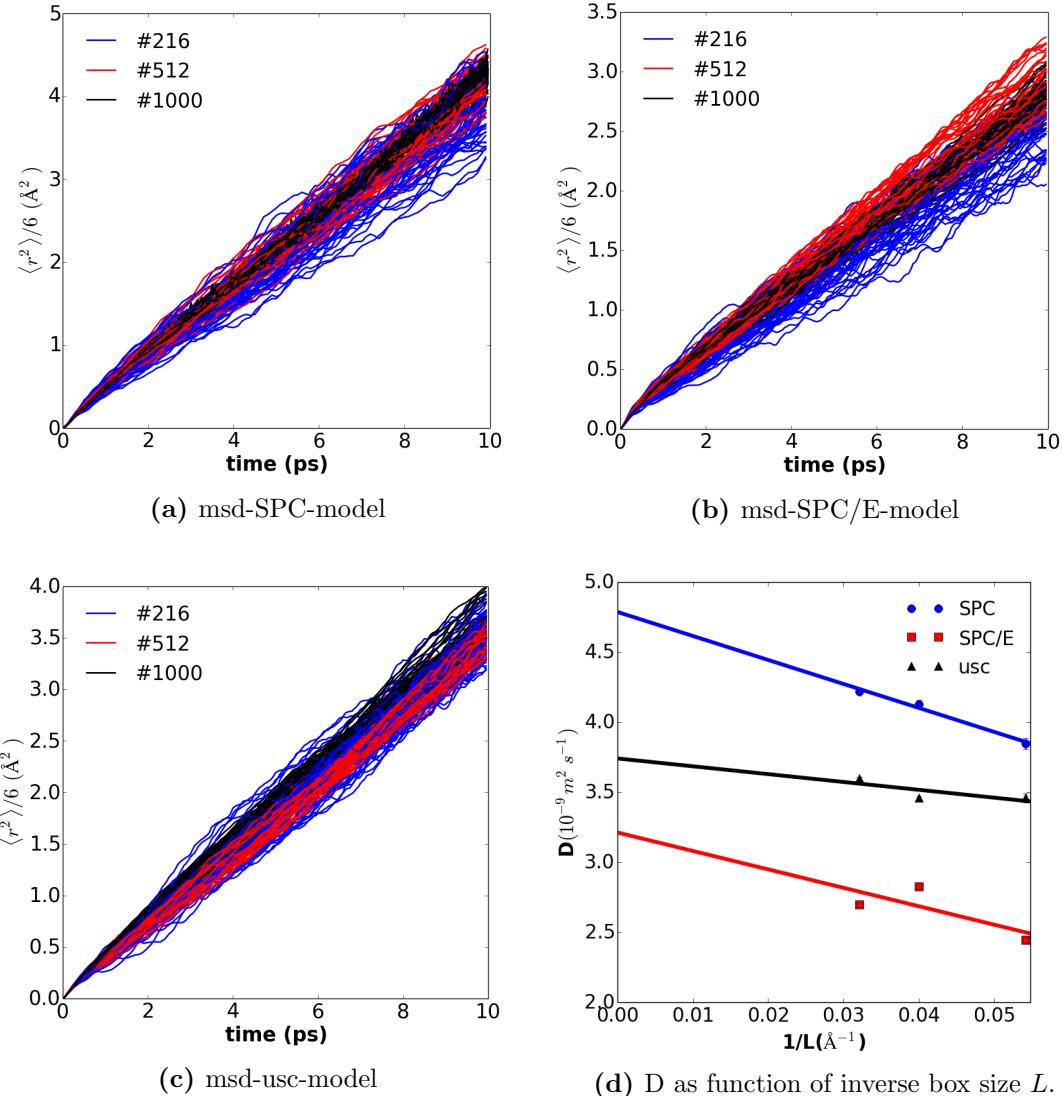


Figure 10.1: (a)-(c) Mean square displacement for three different system sizes, (a) SPC-model, (b) SPC/E-model and (c) usc-model. (d) The size independent D_0 is extrapolated from the graph and given in table 11.1.

Table 10.1: Bulk Water Diffusion Coeffiecient

water model	Diffusion Coefficient [$10^{-9}\text{m}^2/\text{s}$]			Others [36]	
	D#216	D#512	D#1000	D_0	
SPC	3.85 ± 0.04	4.13 ± 0.03	4.22 ± 0.02	4.79	4.22
SPC/E	2.45 ± 0.02	2.83 ± 0.02	2.70 ± 0.01	3.21	2.75
usc	3.46 ± 0.03	3.46 ± 0.01	3.60 ± 0.01	3.74	

displacement plot, we find an estimate of D . Our average values are listed in table 11.1. We see how the system size influences the D-values, the SPC/E-model is closest to experimental values. The D-value for usc is closer to an experimental value than the SPC-model.

10.1.2 Diffusion in Nanopores

We report the diffusion of water inside nanopores with $H = 3, 4$ and 6 nm . The diffusion coefficient is found as a function of wall distance z . Given that the surface is the position of the outermost hydrogen atoms. To uncover any directional dependence, the diffusion coefficient is calculated in the xyz, xy and z-direction, individually. Trajectories of the oxygen atom is sampled every 50 fs for 40 ps. To account for atoms moving in and out of bins, we sampled the mean square displacement for 5 ps for each origins. Using a total of 35 origins to estimate the diffusion coefficient, in 18 bins for the 3 nm pore, 25 bins for the 4 nm pore and 37 bins for the 6 nm pore.

We see that our values for diffusion are low close to the pore wall, but increases towards bulk value inside the pore. Our results are consistent with the findings of [6] and [4]. Where they found that $D(z)$ approaches a bulk value inside the pore, for the SPC/E-model. In figure 10.2a, diffusion in the xyz-direction is plotted. For the 3 and 6 nm pore, no values are found close to the surface. We note that bulk values for diffusion are reached in the 4 and 6 nm pore. Figure 10.2b, shows the in-plane diffusion. Bulk values are reached within $8 - 10\text{ nm}$. We observe signs of the enhanced diffusion in two-dimension, as expected from statistical mechanics. D_z is plotted in figure 10.2c, bulk water values are reached further inside the pore, and not reached at all for the two smallest pores. Surface effects are prominent in this plot. Note that in general the system is larger in xy-direction than z-direction, resulting in lower D-values, as we saw in the previous chapter. From figure 10.2d, we see how 2D-diffusion is larger than 1D and 3D-diffusion.

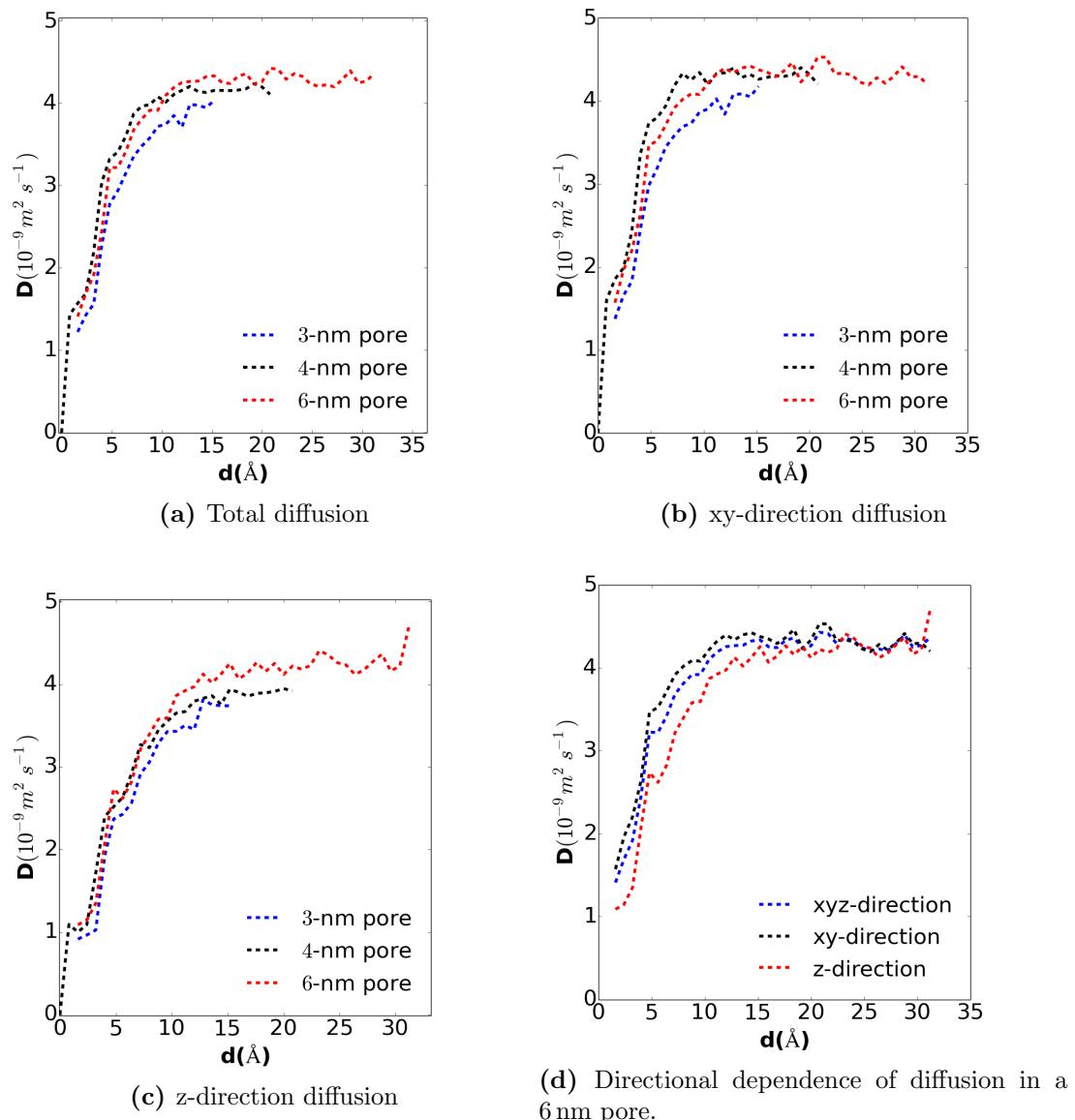


Figure 10.2: (a) Diffusion in the xyz-direction as a function of distance from the pore wall. (b) xy-direction diffusion as a function of the distance to the pore wall. (c) z-direction diffusion as a function of distance from the pore wall. (d) 6 nm pore directional dependence of D .

10.2 Equilibrium Computations of Bulk Water Viscosity

Viscosity is the resistance of a fluid to external forces; here we calculate the viscosity using two equilibrium approaches. Meaning that when the system is in equilibrium, there are no external forces on the system. From the plot of the diffusion coefficient versus the inverse of the box size, we find that D is dependent on system size and viscosity η [60]. The more well-known method to estimate viscosity is by Green-Kubo relations. The first method is described by

$$D_{PBC} = D_0 - \frac{2.837k_B T}{6\pi\eta L}, \quad (10.2)$$

here D_0 is the extrapolated diffusion coefficient from figure 10.1d, η is the shear viscosity, k_B is Boltzmanns constant, T is temperature and L is box length. The values of η are estimated from this equation by re-arranging it and using the mean value of D_{PBC} ,

$$\eta = \frac{2.837k_B T}{6\pi L(D_0 - D_{PBC})}. \quad (10.3)$$

A control of the units

$$\begin{aligned} \eta \left[\frac{\text{kg}}{\text{m s}} \right] &= \frac{2.837 \times k_B \times \left[\frac{\text{m}^2 \text{kg}}{\text{s}^2 \text{K}} \right] \times T[\text{K}]}{6\pi L[\text{m}] (D_0 - D_{PBC}) [\frac{\text{m}^2}{\text{s}}]} \\ &= C \frac{\left[\frac{\text{m}^2 \text{kg}}{\text{s}^2} \right]}{\left[\frac{\text{m}^3}{\text{s}} \right]} \\ &= \left[\frac{\text{kg}}{\text{ms}} \right]. \end{aligned}$$

Now we turn to a more accurate method for viscosity calculations.

10.2.1 Green-Kubo Computation of Viscosity

Shear viscosity is a measure of the fluids resistance to flow. Flow in molecular dynamics is an out of equilibrium system state. Green-Kubo relations allow us to do measurements on a system in equilibrium. The principle is based on pressure fluctuations and the correlation $C(t)$, in time off these fluctuations. This method consists of computing the off-diagonal components of the stress tensor e.g. P_{xy} or P_{xz} . From the relation

$$\eta_{GK} = \frac{V}{k_B T} \int_0^\infty \left\langle P_{\alpha,\beta}(t) P_{\alpha,\beta}(0) \right\rangle dt, \quad (10.4)$$

where we compute the auto-correlation function for the off-diagonal elements of the stress tensor. When the integral of the stress correlation function converges, it defines the viscosity from measurements on the equilibrium state of a system. The autocorrelation function is given as

$$C_{\alpha,\beta} = \left\langle P_{\alpha,\beta}(t)P_{\alpha,\beta}(0) \right\rangle, \quad (10.5)$$

where the angle brackets denotes the average value of the correlation function over a number of samples.

We sample the stress tensors $P_{xx}, P_{yy}, P_{zz}, P_{xy}, P_{xz}$ and P_{yz} every time step $\Delta t = 1 \text{ fs}$, for 2 ns . The off-diagonal components give each an independent value of η , the diagonal components give a value for η from $P_{xx} - P_{yy}$ and $P_{yy} - P_{zz}$. This gives us several estimates on η and better statistics. We compute the auto-correlation function for the whole time-array with the fast Fourier transformation function available in *NumPy*¹. The integral of the auto-correlation function is found from $t = 0$ to $t = 250 \text{ ps}$, and plotted as a function of time.

Note on Green-Kubo

Two methods were considered for sampling the auto-correlation function(acf). We ended up with the Fourier transform method on the whole time-array, this gives us only one estimate of the acf. Another possible solution is to find the correlation function for a number of origins, t_0 , and sample over a shorter window time t_w . By moving the origin 10 fs at a time, estimating acf and moving the origin again. Then we would have got a several averages for acf and probably better statistics. However, the method proved to be incredible slow because we did not use Fourier transformation, but matrix multiplications.

10.2.2 Equilibrium Viscosity Results

We find the bulk water viscosity from equation 10.3, and Green-Kubo relations (GK). The results are compared with other articles for the SPC and SPC/E model, for the usc-model there is, to the best of our knowledge, no estimates of the viscosity. To calculate the viscosity of bulk water we use equation 10.3. The D_{PBC} value was found from the largest system sizes L , D_0 was the extrapolated value from table 11.1. Results are listed in table 10.2.

This is the first estimate of the viscosity for the usc-water potential. If we compare the SPC/E results² with [60], we see that the results differ by 39% for the η_{PBC} value. For the GK computations with the SPC-model, we compare with the study [65], where they found the viscosity to be $0.399 \pm 0.003 \text{ cP}$.

¹<http://www.numpy.org/>

²The units are here in Poise $1 \text{ Pa} \cdot \text{s} = 10 \text{ P}$

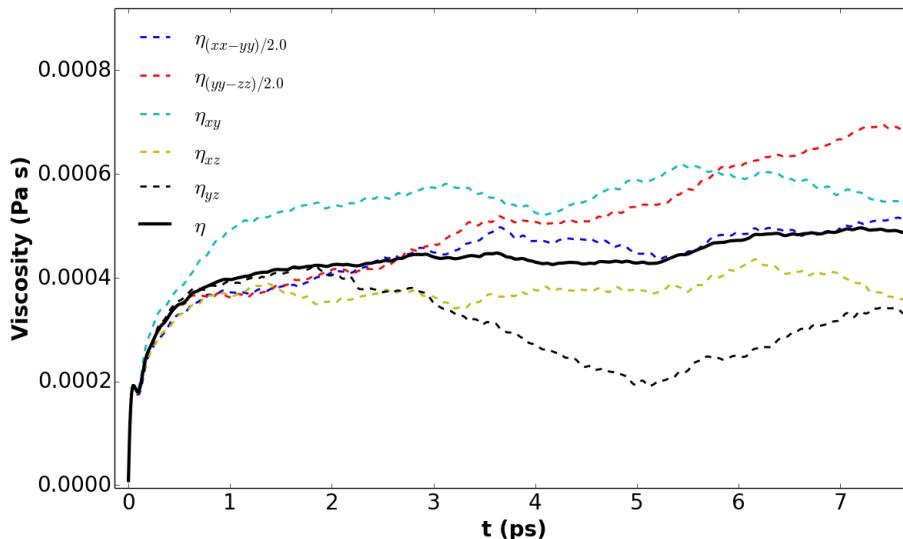


Figure 10.3: Green-Kubo calculation of equilibrium viscosity. Five individual estimates of η are calculated from six stress values.

Table 10.2: Bulk Water Viscosity

Water model (300 K)	$\eta_{\text{PBC}}(\text{cP})$	$\eta_{\text{GK}}(\text{cP})$
SPC	0.35	0.45 ± 0.022
SPC/E	0.39	
usc	1.48	
Exp.	0.889	

In figure 10.3 we have plotted the viscosity η as a function of time. We average over the six different components of the stress tensor $P_{\alpha,\beta}$, and get five estimates for the viscosity. Our computation of the viscosity for the SPC-model is found in table 10.2. The mean value is found by taking the average of the black line in figure 10.3 from 2 ps to 10 ps. Our value would have been closer to literature if we had taken the average between 1 – 2 ps.

Chapter 11

Non-Equilibrium Molecular Dynamics

Non-equilibrium molecular dynamics or NEMD is the theory of out of equilibrium molecular dynamics. When a system is out of equilibrium we are in this context considering a streaming velocity for all the particles. As mentioned in chapter 5, there are several issues to consider when imposing flow to a system of discrete particles. One issue is that the thermostat must be turned off in the direction of flow, because the thermostat rescales particle velocities and we would never get a steady velocity profile. In this chapter we simulate flow between two parallel plates, we calculate the shear viscosity, slip-length and flux. These results are compared to continuum results and we assess the relevance of a continuum description of nanopores.

Elastic properties of portlandite are investigated by straining the system. Again the system is out of equilibrium due to the stress which arises inside the system under strain. We compare elasticity constants found by a reactive versus those found by a non-reactive potential. Our values are then compared to literature, however molecular dynamics simulations of elastic properties of portlandite, has as far as we know, never been done before. Relevant literature is therefore limited, and we compare with results from other clay minerals.

Performing NEMD simulations are not trivial and much of the reason is the time-scale involved. To compute the velocity profile we sampled the velocities for 3 ns. Shearing the system also requires long simulation times. The results derived in this chapter suffer from time-limits and are presented as a guide for other students willing to dig in on one of the topics.

11.1 Hydrodynamics

Gibbs dividing surface is estimated to give a correction in surface position. We use Gibbs surface to compute the slip-length. The shear viscosity is estimated

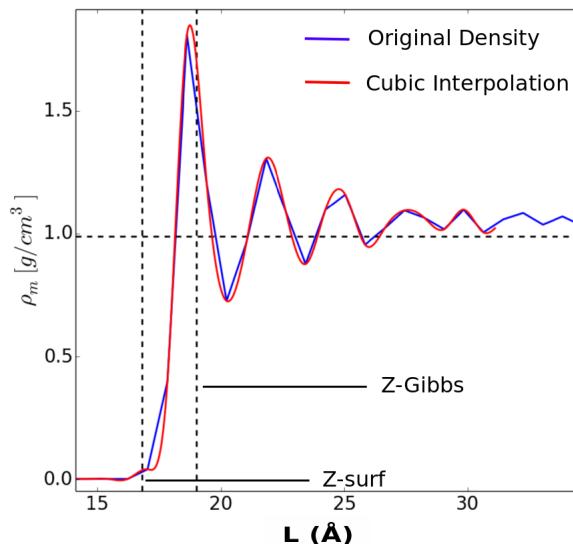


Figure 11.1: Gibbs dividing surface for the 4 nm pore. z-surf is the outermost atoms in the clay, while z-Gibbs is the computed surface.

for water confined in a pore. We then assume constant density and viscosity in the pore, this is a valid approach sufficiently far from the surface. We assess the relevance of a continuous description on a nanoscale by comparing results obtained with NS and molecular dynamics.

11.1.1 Gibbs Dividing Surface

Gibbs dividing surface was found for the 4 and 6 nm pore. We applied the method illustrated in figure 11.1. Density is originally sampled in a limited number of bins, this gives us a non-continuous derivative at some points. We interpolate the points by a cubic interpolation method in *NumPy*, by doing this we get more function values for the integration. Gibbs surface computations can be summarized with the following points

- For all ρ_w values we find the deviation h from bulk density ρ_{bulk} .
- For each value in this array we square to find the relative area
- For each value in h , we find the area to the left and to the right of the i^{th} value
- The difference between the left and right integral gives a minimum value
- This minimum value is the position where the areas to the left and right of the bin are approximately equal

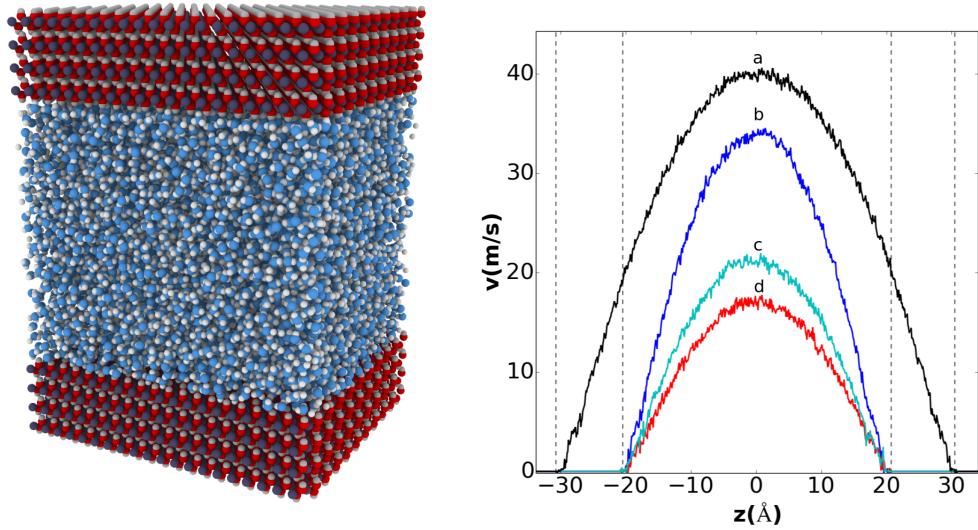


Figure 11.2: (Left), (a) 6 nm pore (Right) Velocity profiles, 6 nm, no SHAKE or special bonds (b) 4 nm Double force, no SHAKE or special bonds. (c) 4 nm SHAKE and special bonds on. (d) 4 nm no SHAKE no special bonds. The dashed lines are the position of the surface hydrogen atoms.

The resulting Gibbs dividing surface was 2.2 and 2.24 for the 4 and 6 nm pore, respectively. This is close to the values found by [4] (1.2 Å) and [37] (2.4 Å). We now have a new surface at which we want to compute the slip-length from.

11.1.2 Shear Viscosity

From the steady state Stokes equation we know that the velocity profile should have a parabolic form (Poiseuille). From figure 11.2, the velocity profile is indeed parabolic in the center of the pore. This curvature allows us to determine the viscosity from equation 5.2. By linear regression we find the cubic fit for the profiles in 11.2. They are on the form

$$v_x = az^2 + bz + c \quad (11.1)$$

By differentiating this equation and plugging the result into equation 5.2 we obtain the viscosity.

11.1.3 Estimates of the Shear Viscosity

The shear viscosity is computed for a 4 and 6 nm pore. We also compute the viscosity, for a system where the force was doubled, figure 11.2 profile (b). We

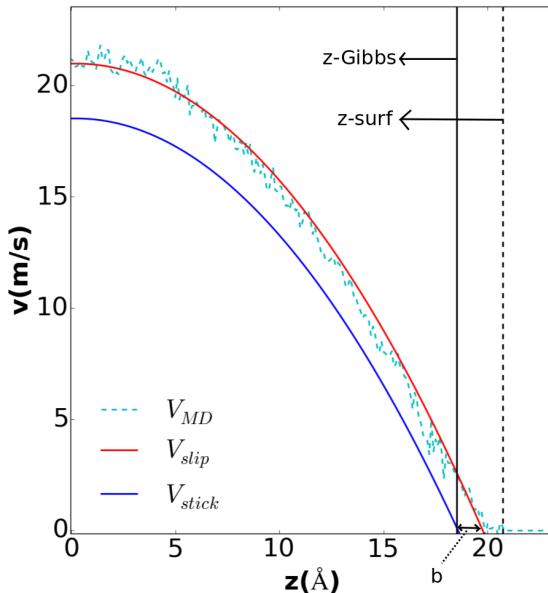


Figure 11.3: Velocity profiles in 4 nm pore. We plot v_{MD} from simulations and fit this with a parabolic v_{slip} profile. From the slip length and Gibbs dividing surface we found v_{stick} where we have vanishing velocity at the Gibbs surface.

computed the viscosity for the velocity profiles a-d. (a) $\eta = 0.56 \pm 0.03$, (b) $\eta = 0.61 \pm 0.04$, (c) $\eta = 0.48 \pm 0.04$ and (d) $\eta = 0.61 \pm 0.04$ cP. Because of a typo in our input script, no special bonds or SHAKE algorithm were turned on for the water molecules in (a), (b) and (d). This was corrected in (c), the viscosity we found here were close to our computations of bulk viscosity, $\eta_{bulk} = 0.45 \pm 0.02$ cP. Which is in reasonable accuracy to our equilibrium viscosity computations, $\eta = 0.48 \pm 0.04$ cP.

11.1.4 Slip Length

We compute the slip length for the systems (a) and (c) in figure 11.2. From the calculation of the viscosity we have the interpolation of the curves, which gives us the expression for the velocity v_x . By using equation 5.8, we can compute the slip length from the fitted velocity profile and Gibbs dividing surface z_{Gibbs} . Hence we obtain for system (a) $b = 0.85 \pm 0.01$ Å and for system (c) $b = 1.29 \pm 0.02$ Å. The discrepancy between the two values is believed to stem from no special bonds, and the SHAKE algorithm turned off for system (a). These values are a bit lower than Botan found in [4], $b = 2.1 \pm 0.05$ Å. But he was investigating the slip-length in montmorillonite clays, not portlandite.

The observed slip length seems small, to assess the contribution from slip vs

stick we calculate the overall flux from NS, using v_{slip} and v_{stick} . Ideally we would have computed the flux from MD, just tracked the particles as they passed the boundary $x = L_x$. This was however not done, but assuming from figure 11.3, we see that the fitted line follows the MD parabola quite well. From this we are able to calculate the difference in flux, from the v_{slip} and v_{stick} profile. For a 4 nm pore we find $j_{\text{stick}} = 0.28 \text{ \AA}^3/\text{fs}$ and $j_{\text{slip}} = 0.34 \text{ \AA}^3/\text{fs}$, so for the slip system the flux is 18 % larger than for a stick system. A drastic difference when we know that much of the fluid inside shale pores is under such confinements.

11.2 Material Properties

In the following we assume isotropy in the xy-plane and anisotropy in the z-direction normal to the clay layering. We also assume that there is no strain rate dependency on the elasticity constants. We compute Young's modulus, Poisson's ratio for five system sizes for the clay force field. For a $10 \times 10 \times 1$ system we compare these values with values we find with ReaxFF. From chapter 11.2, we know that stiffness parameters are directional dependent in a TI material. We find the in plane value and out of plane Young's modulus value according to equation

$$\epsilon_x = \frac{\sigma_x}{E} - \frac{\nu}{E} \sigma_y - \frac{\nu'}{E'} \sigma_z . \quad (11.2)$$

In the above equation the strain in the x-direction is dependent on the stress in y and z-direction. However, since the system is in vacuum we assume $\sigma_y = \sigma_z = 0$, when we strain the system in the x-direction. This allows us to calculate Young's modulus in x, y and z-direction from the linear relationship $E = \frac{\sigma_x}{\epsilon_x}$.

As mentioned there are several ways to find the stress-strain relationship in molecular dynamics. In this thesis we follow an engineering approach, much in the same direction as we would have done experimentally. We know that our material is a TI-material, giving five independent elastic constants. From definition, Poisson's ratio is given as

$$\nu_{ij} = \frac{-\epsilon_{jj}}{\epsilon_{ii}} . \quad (11.3)$$

Here ϵ_{ii} is the strain in the direction of the applied stress and ϵ_{jj} is the strain in the lateral direction. In general we have $\nu_{ij} \neq \nu_{ji}$.

11.2.1 Stress and Strain Relations

Young's modulus and Poisson's ratio are computed for system sizes $10 \times 10 \times 1$, $10 \times 10 \times 2$, $20 \times 20 \times 1$, $20 \times 20 \times 2$ and $5 \times 5 \times 10$. Systems simulated with ClayFF is relaxed for 100 ps then strained for 200 ps, with a strain rate of 0.001 /ps. ReaxFF systems were relaxed for 50 ps and strained for 100 ps, with a

Table 11.1: Young’s modulus and Poisson’s ratio for Portlandite

		ClayFF	ReaxFF	Others ¹
Young’s moduli/GPa	1	28.93	61.04	99.39
	2	33.56	73.06	99.39
	3	29.98	21.31	36.29
Poisson’s ratio	1	0.24	0.20	0.24
	2	0.28	0.21	

strain rate 0.002 /ps. Values for Young’s modulus and Poisson’s ratio are averaged for the various system sizes, since there is little to none system size effect, see figure 11.4b. By interpolating the linear regime, region-I in figure 11.4a, we find elasticity constants in all three directions. Poisson’s ratio is only presented in the isotropic plane.

11.2.2 Yield Fracture

We strain the portlandite parallel and normal to the clay layers and plot the stress-strain curve for different system sizes.

From figure 11.4, we observe that the linear regime of the stress-strain curve is to a good approximation independent of system size. In figure 11.4a, there are four regions with different behavior. Region-I is the elastic linear regime of the system. For a TI-material we expect equal response to stress in the plane of isotropy, xy-plane. In the z-direction, normal to the plane of isotropy we observe a shorter region-I. Parallel to the clay layers there are no covalent bonds to break, but weaker hydrogen bonds. We see from figure 11.4a that this has an effect on the stress-strain curve. Hantal et al. [23] obtained similar results for their stress-strain curves, i.e. peak stress values and value of strain at yield strength.

In the figures 11.6a and 11.6c, we see a portlandite slab pre and post-crack, respectively. Lack of pre-existing cracks make this a yield fracture. For the non-reactive force field ClayFF, this is in general not valid. Due to the breaking of chemical bonds. Still, it is interesting to observe how the fracture propagates through the portlandite and compare with the ReaxFF.

Because the stress-strain curves display the same behavior, as seen in figure 11.4b, we modeled a $10 \times 10 \times 1$ unit cell system with the ReaxFF. Again we plot the stress-strain relationship and compare with our result from the ClayFF. Stress-strain curves plotted for the ReaxFF reveals much larger fracture toughness 11.5a, by a factor of three. We also see a higher strain rate before the transition to region-III. Region-I is divided into two parts, split by a transition

¹DFT from [33]

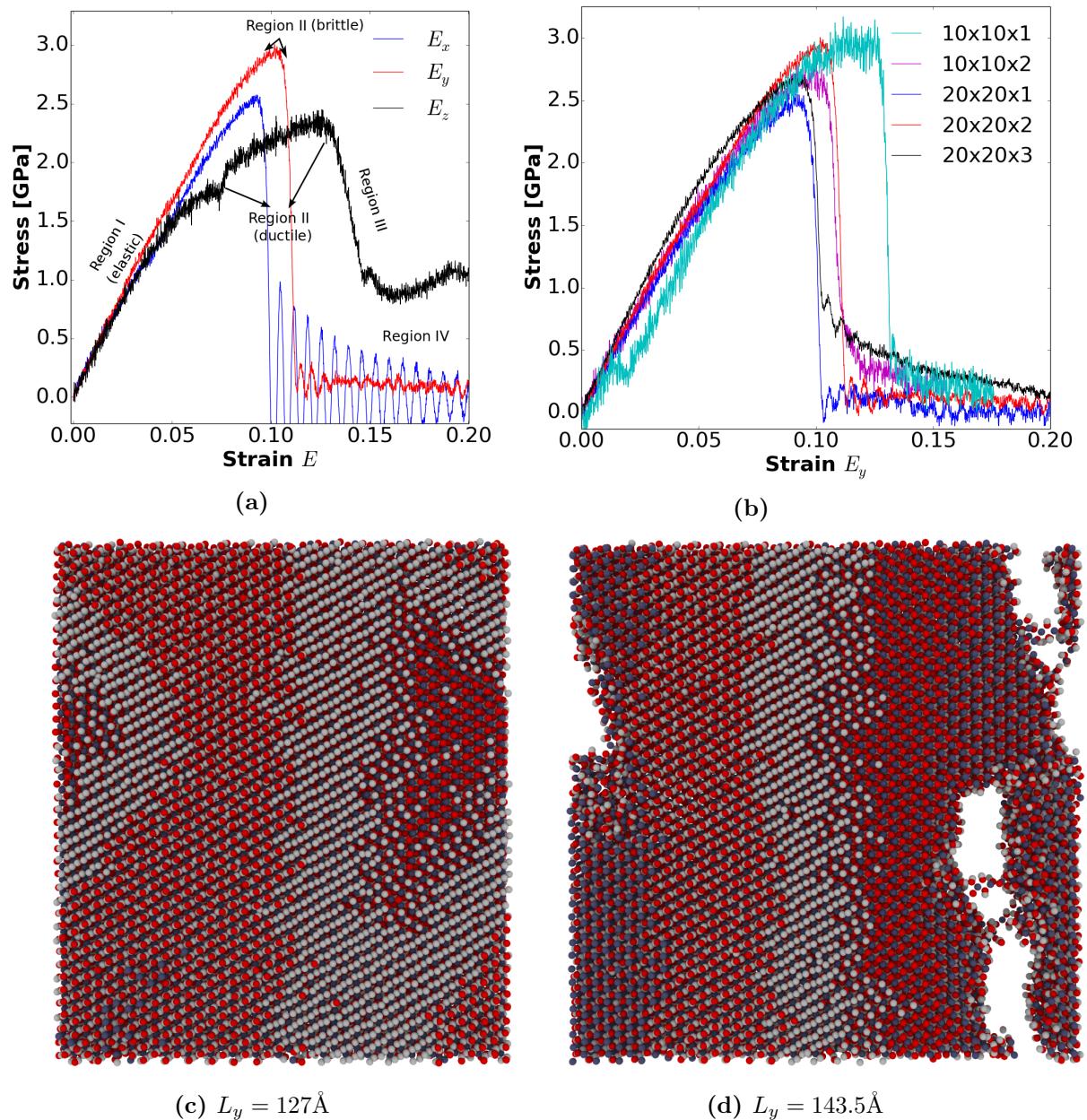


Figure 11.4: Stress vs strain. (a) Direction dependence of stress. (b) System effect of stress vs strain. (c) Pre-strain, system size $146 \times 127 \times 18.8\text{\AA}$ (d) post-strain system size $145.5 \times 143 \times 18.7\text{\AA}$

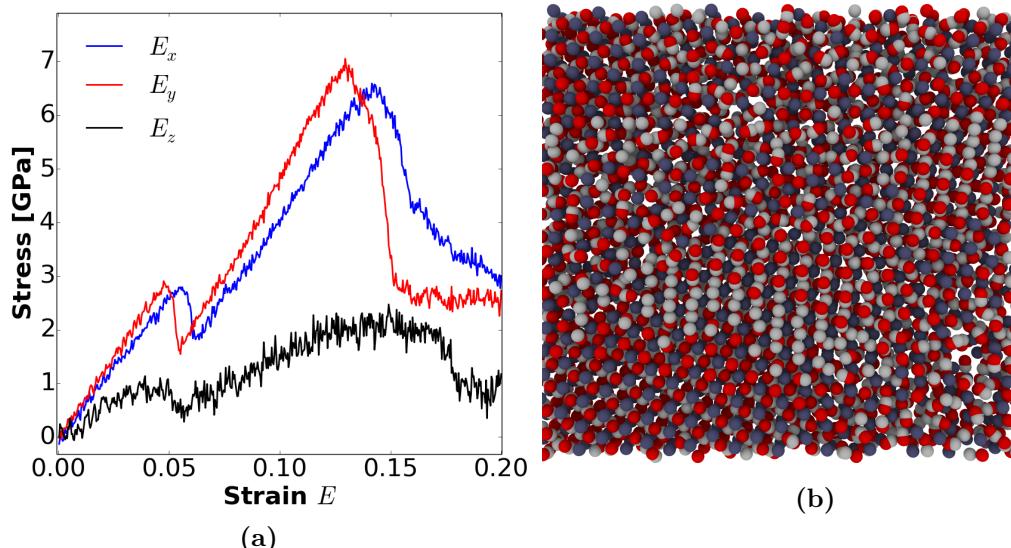


Figure 11.5: (a) Stress-strain relation for a $10 \times 10 \times 1$ unit cell portlandite.
(b) System after the strain, notice no evidence of an open fracture.

from region-I behavior to region-II and III behavior. Before the stress again raises linearly with the strain. This is seen for all three directions. We believe this is a result of the applicability of the ReaxFF to change atom configurations to maintain bonding. Resulting in a short dip in stress while the atoms re-configure before the bonds are equally strong, or stronger in the new atom configurations. To support our suggestion we refer to figure 11.5b, where there is no evident fracture after the large dip in the stress-strain curve.

Comparing E_z for ClayFF and ReaxFF in figure 11.4a and 11.5a, we note that the stress-strain curve displays similarities for the magnitude of the stress and strain value at the point of fracture. Even the little dip at ϵ_{zz} is visible. This indicates that fracture studies parallel to the clay layers are possible for a non-reactive force field.

11.2.3 Initial Crack Fracture

We study and compare a $10 \times 10 \times 1$ unit cell portlandite system, with the ClayFF and ReaxFF. By comparing a non-reactive force field with a reactive force field, we hope to quantize the difference and explain why we need a reactive force field to model fracture mechanics. For our stress computations we dumped the per atom strain $P_{\alpha,\beta}$ every 300 fs, these values were averaged over 10 ps. The system modeled with ClayFF is equilibrated for 200 ps before we strain it for 200 ps with a strain rate of 0.001/ps. ReaxFF is known to be a computational expensive force field, therefore the system was only equilibrated for 50 ps, with a time step

of 0.1 fs. Still though, temperature and energy rapidly settled for an equilibrium value. Our sampling run lasted for 100 ps, with a strain rate of 0.002/ps. Giving the same total strain of the system as for the ClayFF model, $\epsilon_{\text{total}} = 0.2$.

Ovito² has a built in bin and reduce function, which allows us to bin the per atom stress P_{yy} in a 35×30 grid. Plotting is then done with python, and we get the stress distribution in the direction of strain for the system(figures 11.6 and 11.7).

11.2.4 Determine the Stress Intensity Factor K

As we saw in chapter 11.2, the stress around a crack tip diverges when the strain increases. The critical stress intensity factor K , is found for a $10 \times 10 \times 1$ system with both the ClayFF and ReaxFF. From chapter 11.2, we saw that we had a breakdown of continuum when $r \rightarrow 0$, the stress would increase to infinity. It is shown in [2], that for a finite system with a periodic image of cracks, like our system, the Mode I stress intensity factor can be written as

$$K_I = \sigma \sqrt{\pi a} \left[\frac{W}{\pi a} \tan \left(\frac{\pi a}{W} \right) \right]^{1/2}. \quad (11.4)$$

Here a is half the fracture length, and W is the width of the system. We computed K_I by tracking the distance between the two outermost atoms in the fracture. The width of the system is given in the dump files from LAMMPS. This allows us to plot the stress intensity factor as a function of time 11.8. The peak values in this plot is called the critical stress intensity factor, K_{Ic} .

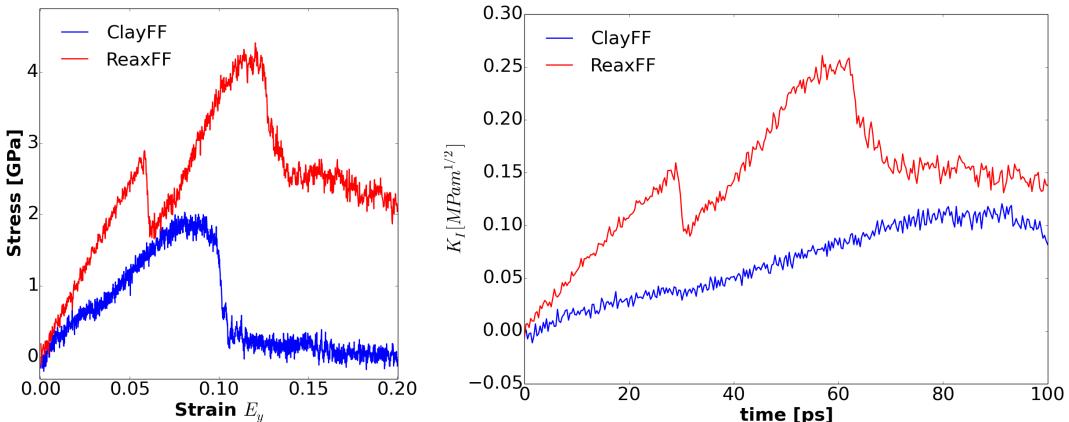


Figure 11.8: (Left) figure illustrates how the stress-strain curves for crack failure normal to the clay layers. (Right) Is the stress intensity factor K_I plotted as a function of time.

²<http://ovito.org/> [58]

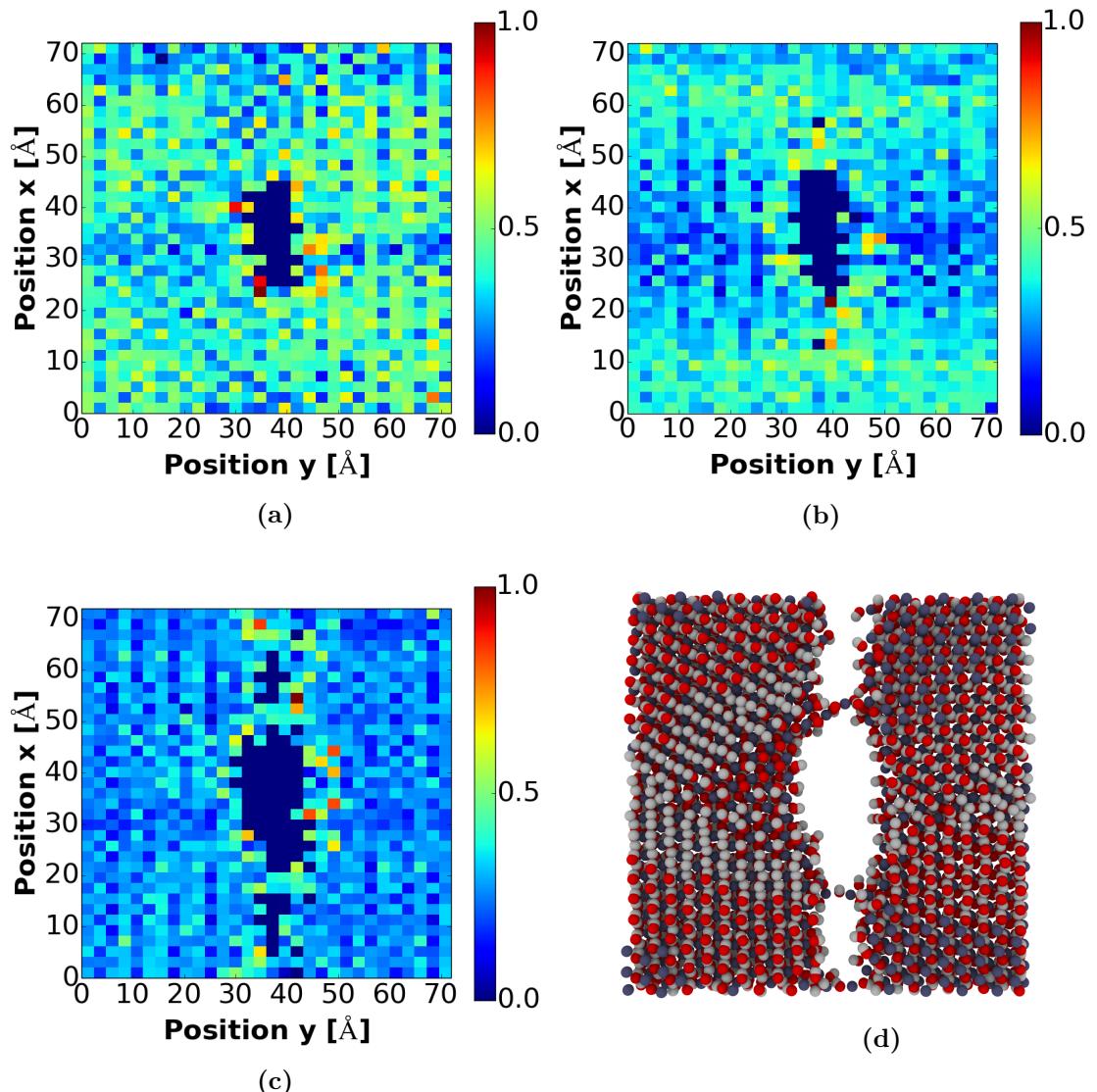


Figure 11.6: ClayFF: (a) Per atom stress before crack, (b) per atom stress initial phase of crack, (c) per atom stress during fracture, (d) Portlandite system $10 \times 10 \times 1$

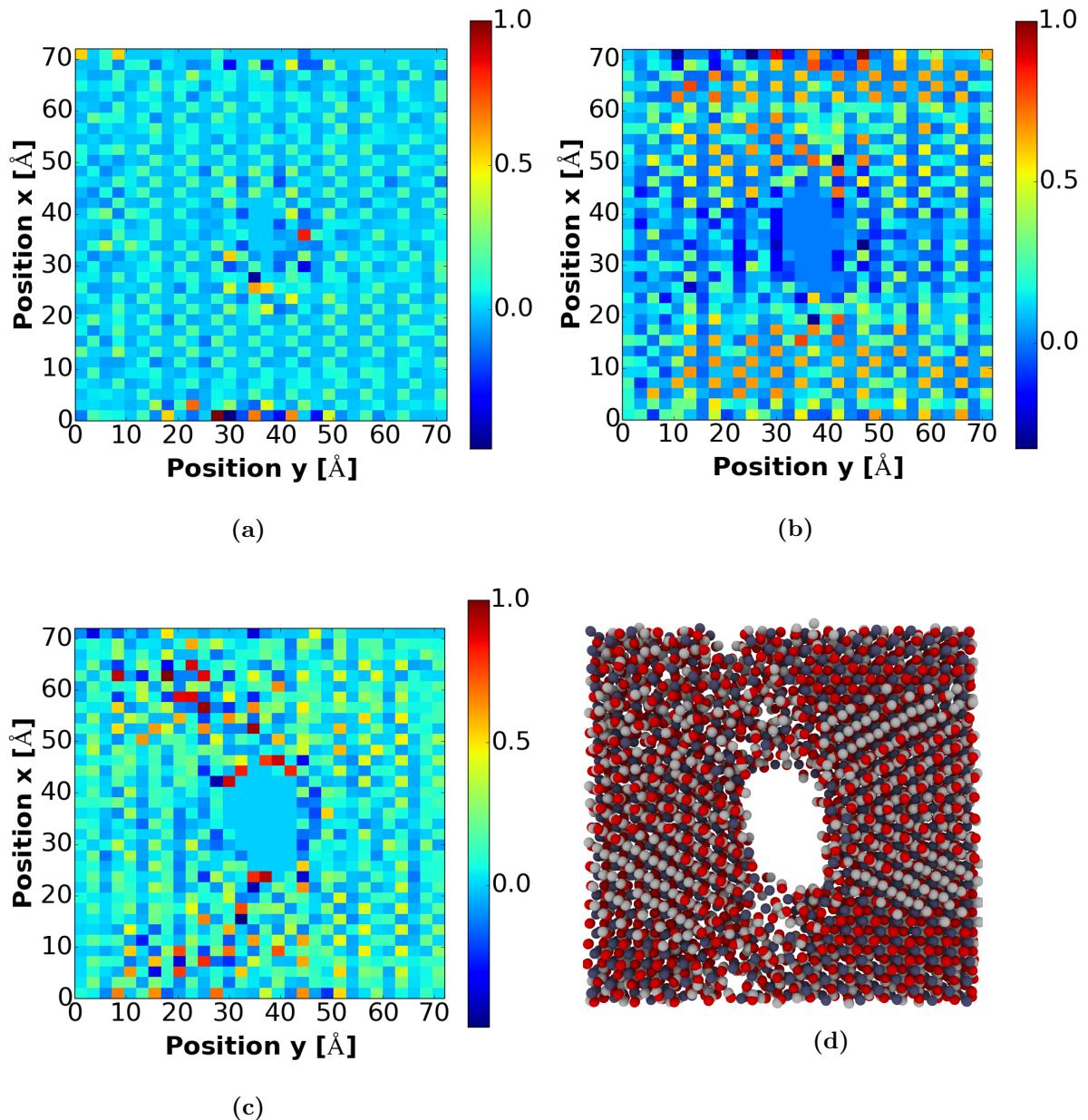


Figure 11.7: ReaxFF: (a) Per atom stress before crack, (b) per atom stress initial phase of crack, (c) per atom stress during fracture, (d) Portlandite system $10 \times 10 \times 1$

The maximum value reached for K_{Ic} is $0.26 \text{ MPam}^{1/2}$ and $0.12 \text{ MPam}^{1/2}$ for ReaxFF and ClayFF, respectively. We did not find any studies that computed K_{Ic} for portlandite, Hantal [23] found the fracture toughness for illite to be $0.12 \text{ MPam}^{1/2}$ with ClayFF. This value was however parallel to the clay layers, from our stress-strain curves 11.4a, we find that there is little direction dependence, so the K_{Ic} value would most likely be the same in the parallel direction as well. For ReaxFF Hantal, found $0.61 \text{ MPam}^{1/2}$, for a failure normal to the clay layers. This is higher than our value, but again, they used illite not portlandite.

Part IV

Discussion, Conclusion and Future Work

Chapter 12

Discussion, Conclusion and Future Work

12.1 Discussion and Conclusion

Here we discuss our results for structural properties, diffusion, viscosity, flow, elasticity and fracture. The different topics are divided into individual sections where we summarize and discuss our findings. A short conclusion is drawn, if we find this adequate.

12.1.1 Structural Properties

We found the radial distribution function of bulk water, and from this we got the distance between atoms. This allowed for a direct comparison between our implementation of the usc-potential, and two established water models; SPC and SPC/E. We saw that main features, such as O-H and O-O distance were approximately equal for usc-water and the two SPC-models. Angle distribution, showed that the H-O-H angle of usc-water were much more flexible than for SPC-water. This is simply because the usc-model is a many-body potential, without restrictions on the bond angle as for SPC. For some reason the developers of the usc-potential have set the equilibrium H-O-H angle to 104.47°, indeed we got a function distributed around this mean. This is not a correct H-O-H angle for water. The correct value is the one used in the SPC and SPC/E models, 109.47°. Our $g(r)$ results were close to the unpublished results found by the usc-developers.

We tested our implementation of the usc-potential on silica, by substituting water parameters for silica parameters. Again, correct values were hard to obtain, because usc-parameters are not open source. Annealing silica resulted in correct a-silica density, with both our usc-implementation and the ClayFF potential. The distribution function of silica for our usc-potential, fits that of silica

parametrization of ClayFF. Surprisingly, ClayFF gave a better estimate than the usc-potential of the O-Si-O angle distribution. This is unexpected since ClayFF is a two-body potential, while usc has a three-body term supposed to penalize non-tetrahedral angles. A possible explanation is the heat/cooling procedure, if the usc-silica was cooled too fast, oxygen and silica atoms might have been looked in unphysical angles.

Bulk water densities were found and acted as a validation of potential parameters. All bulk densities were close to values found in literature. Density of water inside a nanopore was found as a function of wall distance z . The density showed high fluctuations at the liquid/wall boundary, the possible hydrophilic surface of porlandite attracted water molecules, resulting in a higher density closer to the wall. This liquid/solid interaction probably causes water hydrogens to point toward the pore center, creating a hydrophilic film at the surface. The drop in density after the first peak is most likely a result of this rearranging. This again has implications on the next water layer, arranging its oxygen atoms closer to the wall, resulting in the next peak of the density plot. Our theory is supported by the width of the density peaks, which are about 3 Å wide. Pretty much the same size we estimated for a water molecule. This continues until the wall/liquid interactions are so weak that water behaves as in bulk again. Bulk behavior happens to occur at approximately the same distance as the length of our interatomic cutoff.

A way to characterize hydrogen bonding in water is through the tetrahedral order parameter (TOP). Strong hydrogen bonding allows for a tetrahedral structure of water molecules. We found the tetrahedral order parameter for bulk water and water confined in a nanopore. Bulk water estimates were done for all three water models, in the pore we used the SPC-model. The average values for the three water models were a bit low compared to literature. It is interesting to note the difference in the SPC and SPC/E-models for the TOP plots. For the SPC/E-model, a characteristic second peak is more prominent than in the SPC-model. The higher point charge value in the SPC/E-model is most likely responsible for creating a more tetrahedral structure, because of stronger hydrogen bonding. Our usc-model lacks this second peak, a dipole-dipole term is responsible for the hydrogen bonding in the usc-potential. A too weak parameter value for the dipole-dipole term is a possible answer to this question.

12.1.2 Diffusion and Viscosity

A bulk diffusion coefficient was computed for all three water models, using the mean square displacement and the Einstein relation. We saw how sampling msd-values for several time origins increased the accuracy of our estimate of D , our values were close to those found in literature, a small deviation from what others have found is expected. By sampling the diffusion coefficient for different system

sizes and extrapolating these values, we found D_0 for an infinite system size. We should have run for a larger system size to get better measurements close to $1/L \rightarrow 0$. A diffusion coefficient for our implementation of the usc-potential was also found, unfortunately there are no published results to compare these results with. It is known that many-body potentials, such as ReaxFF have problems obtaining the correct diffusion coefficient. So for a many-body potential the D-value we obtained was quite good, actually closer to experimental values for D than the SPC-model.

Values of D were found as a function of wall distance for pore sizes of $H = 3, 4$ and 6 nm . Near the portlandite, water proved more stationary, because of the confinement and suggested hydrophilic surface of portlandite. As for the density, D reaches a bulk value after 10 \AA . For the 3 nm pore, D-values are still lower than bulk after 10 \AA . Diffusion in the xy -plane displayed the same behavior as the total diffusion, but it reaches a higher bulk value. This enhanced diffusion in two-dimensions is known from statistical mechanics. In the z -direction the peak values for D were lower than bulk value for the 3 and 4 nm pores. The 6 nm pore reached a bulk value in the z -direction further inside the pore, than for diffusion in xyz and xy -direction. The curvature of the D_z plot was stretched further into the pore, reaching bulk values later. This can be explained by two things: 1) an artifact of diffusion and 2) by the physics behind the liquid/wall interaction. 1. As we saw for bulk water, a smaller system size resulted in a lower diffusion coefficient. Because the pore is smaller in the z -direction than the xy -direction, we should expect a lower D-value from system size effects. 2. The portlandite is most likely hydrophilic, restricting the movement of a water molecule by holding on to it with its potential. Bulk values were attained after 10 nm , the same distance as our potential cutoff value.

Viscosity was calculated using: 1) the extrapolated D_0 value from diffusion, and 2) Green-Kubo relations. The periodic boundary condition method proved poor at estimating the viscosity, a too low value for the SPC and SPC/E-models were found. The usc-value was high, but again we have no literature to compare with. Green-Kubo computations could have been used to find a better estimate of the viscosity for the usc-model. Green-Kubo estimates showed a large span in according to which of the stress components that were used. For the interval $1 - 2\text{ ps}$, all the calculations are within reasonable distance from each other, and our error was comparable to values found by others.

12.1.3 Hydrodynamics

Density fluctuations near the wall called for a new definition of the liquid/solid boundary. This was done by finding Gibbs dividing surface, which is the distance z_{Gibbs} where the area below and above bulk density is equal to the left and right. Our Gibbs surface values were close to published values for other water/clay

boundaries, considering that other studies used different water models and clay minerals. Our measurement of z_{Gibbs} were only used for the flow simulations, however it would have given us a more correct surface to measure z-dependence of diffusion.

Shear viscosity was found, by interpolating the parabolic velocity profile, giving us an equation for the velocity. The viscosity was found for four different profiles; three of these a, b and d were done without the fix SHAKE algorithm. This gave us an inaccurate answer for the shear viscosity compared to our bulk water estimates. Sample c) was done correct and its standard deviation was within the range of our bulk value from Green-Kubo computations. Flow was not studied in the 3 nm pore, because the bulk water region was too short.

Estimates of slip length were done for the 4 nm and 6 nm pore. We used the interpolation of the velocity profile from the viscosity calculations, Gibbs dividing surface was used to estimate the slip length b . Our values were lower than what Botan [4] found, but their simulation time was 40 ns, ours were restricted to 3 ns. Driving the flow over longer time would probably give a higher velocity at the boundary. We found that neglecting slip could dramatically alter the flux out of the pore. We had to use a large driving force to get a steady velocity profile within a reasonable amount of time. Maximum velocities were therefore around 20 m/s, for experimental slip calculations these velocities are around 2 cm/s. Molecular dynamics simulations are said to underestimate the slip length by a factor of 10. Compared to experimental values found in [67], this is reasonable. But we have to remember that experiments and molecular dynamics simulations are on a different time and length scale.

12.1.4 Material Properties

We found elastic properties of portlandite, such as Young's modulus and Poisson's ratio. We saw that a TI material had different elasticity values for the plane of isotropy compared to the anisotropic direction. Values of Young's modulus for ClayFF in the z-direction were close to density functional values found by Laugesen [33]. Our values for the isotropy plane were however, far off Laugesen's results. Not relaxing the system between the strains may have caused these low E-values. Poisson's ratio was found to be identical to DFT-values for one direction in the isotropy plane. ReaxFF came closer to DFT-values and discerned more between the isotropy plane and the anisotropic direction. More equal Poisson's ratios in the isotropy plane were also found using ReaxFF. Reasons for the low estimates on Young's modulus with ClayFF, stem from a simulation procedure without system relaxation between straining. We believe the yield strength decreases when the system is not allowed to relax, ReaxFF compensated by having higher yield failure strength, as seen later in the fracture simulations.

By continuing straining the portlandite after the linear elastic zone, we found

the yield fracture strength, as the peak stress before fracture propagation. Fractures normal to the clay layering are not well represented with the non-reactive ClayFF. Although the fracture seemed brittle, it underestimates the peak stress before fracturing. For a yield failure parallel to the 001 layer, we saw that ClayFF and ReaxFF showed similar behavior. The 001 direction was found to be the weakest direction of the clay, because of weak hydrogen bonding in this direction.

Several system sizes were tested to find if there was a size dependence of the stress-strain curve. We found that the peak stress value and max strain before failure changes, but not by much. The linear elastic region of the stress-strain curve is close to identical for all system sizes. This came in handy when we wanted to do stress-strain simulations with the more computational expensive ReaxFF. Allowing for a smaller system size and still preserving system characteristics.

We compared the stress field around an initial fracture in the 010 plane, normal to the clay layers. ClayFF seemed to distribute the stress around the whole system, while for ReaxFF, high stress mainly occurred in the vicinity of the fracture. Being a reactive force field, ReaxFF probably alters charge and bond strength close to the fracture instead of distributing the stress over the whole system. For the ClayFF system stress increases until the system fractures, then a sudden stress relaxation follows. Simulations with ReaxFF showed that atoms close to the fracture were subject to high stress. Stress was still high in the system after the first drop in the stress-strain curve, because bond strength changes according to the applied stress, preventing a total rupture.

The stress intensity factor were found for both the ClayFF and ReaxFF systems. A crude estimation of the fracture length a , by tracking the distance between the two outermost atoms allowed us to find K_I . In their study of illite, Hantal et al. compare their fracture toughness K_{Ic} with experimental values of muscovite mica from [56]. In this the experimental study, $K_{Ic} = 0.2 \text{ MPa } m^{1/2}$. Meaning that our values were closer than values found by Hantal et al.

12.2 My Contribution and Future Work

We restate the goals of this thesis

- (a) Implementing a reactive force field in LAMMPS and validating it against existing models.
- (b) Study properties fluids confined in a nanopore, such as structure and diffusion.
- (c) Study fluid flow, with non-equilibrium molecular dynamics to reveal how the slip length alters the flux through a nanopore

- (d) Crude estimations of elastic properties of clay, in which direction is the clay strongest?
- (e) Study fracture mechanics of clays, how does the choice of force field influence our results?

Our main aim for this thesis was to construct a reactive variant of the ClayFF, by keeping the original ClayFF parameters and implement the bond order scheme from the usc-potential. During our work, we found that doing the opposite would have been easier. That is implementing the usc-potential in LAMMPS, and parameterize this potential for clay systems. Our implementation worked for water and silica individually, water/silica systems were also simulated but with no success due to no bond order scheme. For the future we would like to implement the bond order scheme, and try to parameterize this potential for clay minerals. By doing this we hoped to create a reactive force field that was less computational expensive than ReaxFF. So a) was partly fulfilled. b) Was done for water confined in portlandite, because of its simplistic structure, portlandite would have been a good starting point for the parametrization of our reactive potential. No other literature on water migration in portlandite, on molecular level were found, so our findings are the first in its kind for portlandite. c) The slip length and fluid flux were found. Using portlandite was not adequate for fluid flow, but we note that we could easily have substituted it for a different clay mineral. Points d) and e); we found elastic properties of portlandite. The elasticity and fracture results were also first of its kind for portlandite. Although only crude estimates were done, the results showed interesting features of portlandite, and the way we model fractures in molecular dynamics.

12.2.1 Outstanding Questions

Future work consists of parametrization of the usc-potential, so that we can simulate clay minerals. The bond order scheme also needs to be implemented and tested. When this is done we can compare our new potential with results found with the ClayFF for diffusion and structural properties of water and clay. We saw that acceleration driven flow was the simplest method to conduct fluid flow. New methods are however implemented in LAMMPS, so that true pressure driven flow can be performed. Still, this will not be a trivial simulation, because you would have to create a thermostat for the different insertion/deletion regions in the pore. Better and more accurate elasticity and fracture simulations by relaxing and straining the system over a longer amount of time, would have been preferable. How elastic properties are altered when water is present in the fracture, would also be interesting to investigate in the future. We have a starting point for a reactive potential, which is faster than ReaxFF and more accurate than ClayFF. Hopefully, our results can be used as a framework for testing the parametrization of this new potential.

12.3 Molecular Dynamics as a Method for Modeling Shales

We have seen that molecular dynamics simulations are helpful in understanding processes at nanoscale. Shales are porous, with most of the porosity in the range of nanometers, resulting in a high surface area. Modeling fracture propagation and fluid migration at such a small scale, can give insight in processes happening at the large scale, as we saw for the fluid flux, and the directional dependent strength of portlandite. This can be used to develop methods for enhanced fracturing of shales, or structuring the grains in portlandite to increase its strength. There were several concerns regarding shale gas extraction. One concern was the vertical migration of fracking fluid and solutes towards groundwater aquifers. By creating a reactive molecular dynamics potential or using ReaxFF, we get a starting point in addressing these questions about fluid migration and fracture propagation in shales.

Part V

Appendices

Appendix A

SHAKE Algorithm

In this chapter we explain and derive the SHAKE algorithm for constrained movement in molecular dynamics. We follow closely the derivation from Frenkel and Smit's *Understanding molecular simulation: from algorithms to applications* [18].

We consider the case when we apply a bond between two atoms and want their distance d_{ij} to be fixed. The constraints of a particle can then be written

$$\sigma_1(r_i, r_j) = (r_i - r_j)^2 - d_{ij}^2 = 0 \quad (\text{A.1})$$

Where $r_i - r_j$ gives the distance between the particles r_{ij} . If the system is imposed to l constraints we can write a new Lagrangian (summarization of the dynamics of the system) \mathcal{L}' which constraints all the constraints

$$\mathcal{L}' = \mathcal{L} - \sum_l \lambda_l \sigma_l(r^N),$$

where λ is the Lagrangian multipliers. This is the new potential form of our system, and the equations of motion can then be written as

$$\frac{\partial}{\partial t} \frac{\partial \mathcal{L}'}{\partial \dot{x}} - \frac{\partial \mathcal{L}'}{\partial x} = 0 \quad (\text{A.2})$$

Then we have

$$\frac{\partial \mathcal{L}'}{\partial x} = -\frac{\partial U}{\partial x}$$

and

$$\frac{\partial \mathcal{L}'}{\partial \dot{x}} = \frac{\partial}{\partial \dot{x}} \left(\frac{1}{2} m_i \dot{x} \right) = m_i \ddot{x}.$$

The time derivative of the Lagrangian is then

$$\frac{\partial}{\partial t} \frac{\partial \mathcal{L}'}{\partial \dot{x}} = m_i \ddot{x}$$

and we get for the whole system

$$m_i \ddot{x}_i = -\frac{\partial U}{\partial x_i} - \sum_l \lambda_l \frac{\partial \sigma_l}{\partial x_i} \quad (\text{A.3})$$

$$= F_i + \sum_l G_i(l). \quad (\text{A.4})$$

In the last line we find the constraint force G_i this is the part we want to solve for in order to keep the bond distance d_{ij} constant. To solve for the Lagrangian multipliers λ we want to find the minimum or maximum of the above function. This is where the second derivatives of the constraints vanish:

$$\frac{\partial \dot{\sigma}_1}{\partial t} = \frac{\partial \dot{x} \nabla \sigma_l}{\partial t} \quad (\text{A.5})$$

$$= \ddot{x} \nabla \sigma_1 + \dot{x} \dot{x} : \nabla \nabla \sigma_l \quad (\text{A.6})$$

$$= 0. \quad (\text{A.7})$$

We can rewrite this equation using A.3

$$\frac{\partial \dot{\sigma}_1}{\partial t} = \sum_i \frac{1}{m_i} \left[F_i + \sum_l G_i(l) \right] \nabla_i \sigma_1 + \sum_{i,j} \dot{x}_i \dot{x}_j \nabla_i \nabla_j \sigma_1 \quad (\text{A.8})$$

$$= \sum_i \frac{1}{m_i} F_i \nabla_i \sigma_1 - \sum_i \frac{1}{m_i} \sum_l \lambda_l \nabla_i \sigma_l \nabla_i \sigma_l + \sum_{i,j} \dot{x}_i \dot{x}_j \nabla_i \nabla_j \sigma_1 \quad (\text{A.9})$$

This is a set of matrices which can be solved with matrix inversion. However this is time consuming and does not guarantee that the constraints will be satisfied [18]. Let us now consider the Verlet algorithm in the presence of constraints.

$$r_i^{\text{constrained}}(t + \Delta t) = r_i^{\text{unconstrained}} - \frac{\Delta t^2}{m_i} \sum_{k=1}^l \lambda_k \nabla_i \sigma_k(t). \quad (\text{A.10})$$

If we the system follows the constraints at a time $t + \Delta t$ it will satisfy $\sigma_k(t + \Delta t) = 0$. But if not the constraints will not be satisfied. We Taylor expand the constraints:

$$\sigma_k^c(t + \Delta t) = \sigma_k^u(t + \Delta t) + \sum_i^N \left(\frac{\partial \sigma_k}{\partial r_i} \right) \cdot [r_i^c(t + \Delta t) - r_i^u(t + \Delta t)] + O(\Delta t^4). \quad (\text{A.11})$$

Now we insert equation A.10 for $r_i^u - r_i^c$ in equation A.11, we get

$$\sigma_k^u(t + \Delta t) = \sum_{i=1}^N \frac{\Delta t^2}{m_i} \sum_{k'=1}^l \nabla_i \sigma_k(t + \Delta t) \nabla_i \sigma_{k'}(t) \lambda_{k'}. \quad (\text{A.12})$$

Again we note that this has a structure of a matrix equation. And by inverting this equation at every iteration we can solve the system. However this is not very computational efficient and this is where the SHAKE algorithm is presented to solve the equations. Instead of applying the iterative scheme above to all constraints at the same time, we iterate through each constraint in succession. In the Taylor expansion A.11 we use the approximation of $r_i^c - r_i^u$ as

$$r_i^c(t + \Delta t) - r_i^u(t) \approx -\frac{\Delta t^2 \lambda_k}{m_i} \nabla_i \sigma_k(t). \quad (\text{A.13})$$

Inserting equation A.13 into equation A.11 we get

$$\sigma_k^u(t + \Delta t) = \Delta t^2 \lambda_k \sum_{i=1}^N \frac{1}{m_i} \nabla_i \sigma_k(t + \Delta t) \nabla_i \sigma_k(t), \quad (\text{A.14})$$

and we get an estimate for the Lagrangian multipliers λ_k

$$\lambda_k \Delta t^2 = \frac{\sigma_k^u(t + \Delta t)}{\sum_{i=1}^N \frac{1}{m_i} \nabla_i \sigma_k(t + \Delta t) \nabla_i \sigma_k(t)}. \quad (\text{A.15})$$

In the simulation all constraints are treated in succession during one cycle of iteration, the process is repeated until we reach convergence for all the constraints.

Appendix B

Building LAMMPS on a HPC-cluster

In order to execute LAMMPS on a high parallel computing cluster as Abel, there are some differences from building it on your laptop. First you download a tarball from <http://lammps.sandia.gov/download.html>. Then we secure copy the tarball to our home area at LAMMPS.

See the following command :

```
$ scp lammps-version.tar username@abel.uio.no:~/
```

To extract the tarball we type

```
$ tar xvf lammps-version.tar
```

Then *cd* your way into the */src* folder and type the following:

```
/lammps-version/src$ module load .....
```

where the stapled lines are replaced by the following.

- module load intel
- module load intelmpi.intel
- make yes-user-intel
- make package-status (to install the packages you need for your simulations)
- make intel_cpu

The LAMMPS executable is then named lmp_intel_cpu.

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] T L Anderson. *Fracture Mechanics: Fundamentals and Applications*. Vol. 2. 1995, p. 688. ISBN: 0849316561. URL: <http://scholar.google.com/scholar?hl=en%5C&btnG=Search%5C&q=intitle:Fracture+Mechanics%5C#6>.
- [3] Dezső Boda and Douglas Henderson. *The effects of deviations from Lorentz–Berthelot rules on the properties of a simple mixture*. 2008. DOI: [10.1080/00268970802471137](https://doi.org/10.1080/00268970802471137).
- [4] Alexandru Boțan et al. *Hydrodynamics in clay nanopores*. 2011. DOI: [10.1021/jp204772c](https://doi.org/10.1021/jp204772c).
- [5] A. Boțan et al. “How electrostatics influences hydrodynamic boundary conditions: Poiseuille and electro-osmotic flows in clay nanopores”. In: *Journal of Physical Chemistry C* 117.2 (2013), pp. 978–985. ISSN: 19327447. DOI: [10.1021/jp3092336](https://doi.org/10.1021/jp3092336).
- [6] Ian C Bourg and Carl I Steefel. “Molecular Dynamics Simulations of Water Structure and Diffusion in Silica Nanopores”. In: (2012), pp. 2–10.
- [7] Donald W Brenner et al. *A second-generation reactive empirical bond order (REBO) potential energy expression for hydrocarbons*. 2002. DOI: [10.1088/0953-8984/14/4/312](https://doi.org/10.1088/0953-8984/14/4/312).
- [8] John Broderick et al. *Shale gas : an updated assessment of environmental and climate change impacts Shale gas : an updated assessment of environmental and climate change impacts*. Tech. rep. 2011, p. 133. URL: <http://www.324.cat/multimedia/pdf/1/0/1359364607701.pdf>.
- [9] F Célarié et al. “Glass breaks like metal, but at the nanometer scale.” In: *Physical review letters* 90.7 (2003), p. 075504. ISSN: 0031-9007. DOI: [10.1103/PhysRevLett.90.075504](https://doi.org/10.1103/PhysRevLett.90.075504).
- [10] P.-L. CHAU and A. J. HARDWICK. *A new order parameter for tetrahedral configurations*. 1998. DOI: [10.1080/002689798169195](https://doi.org/10.1080/002689798169195).

- [11] Li Chen et al. “Nanoscale simulation of shale transport properties using the lattice Boltzmann method: permeability and diffusivity”. In: *Sci. Rep* 5 (). URL: <http://www.nature.com/srep/2015/150128/srep08089/full/srep08089.html>.
- [12] Louise J Criscenti and Randall T Cygan. “Molecular simulations of carbon dioxide and water: cation solvation.” In: *Environmental science & technology* 47.1 (Jan. 2013), pp. 87–94. ISSN: 1520-5851. DOI: [10.1021/es301608c](https://doi.org/10.1021/es301608c). URL: <http://www.ncbi.nlm.nih.gov/pubmed/22779448>.
- [13] 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>.
- [14] 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>.
- [15] Adri Van Duin and Chenoweth Kimberly. “ReaxFF Potential Functions”. In: *Journal of Physical Chemistry A* 2 (2001), pp. 1–10. ISSN: 1520-5215. DOI: [10.1021/jp709896w](https://doi.org/10.1021/jp709896w).
- [16] P. P. Ewald. “Die Berechnung optischer und elektrostatischer Gitterpotentiale”. In: *Ann. Phys* (1921), pp. 253–287. DOI: [10.1002/andp.19213690304](https://doi.org/10.1002/andp.19213690304).
- [17] M. Fisher and Norman Warpinski. *Hydraulic-Fracture-Height Growth: Real Data*. 2012. DOI: [10.2118/145949-PA](https://doi.org/10.2118/145949-PA).
- [18] D Frenkel and B Smit. “Understanding molecular simulation: from algorithms to applications”. In: *Academic Press* New York, (2002).
- [19] GCS. *Scientists Achieve World’s Largest Molecular Dynamics Simulation*. URL: http://www.gauss-centre.eu/gauss-centre/EN/Home/home_node.html (visited on 05/04/2015).
- [20] Jacob Goldsmith and Craig C. Martens. “Molecular dynamics simulation of salt rejection in model surface-modified nanopores”. In: *Journal of Physical Chemistry Letters* 1.2 (2010), pp. 528–535. ISSN: 19487185. DOI: [10.1021/jz900173w](https://doi.org/10.1021/jz900173w).
- [21] S Granick. “Motions and relaxations of confined liquids.” In: *Science (New York, N.Y.)* 253.5026 (1991), pp. 1374–1379. ISSN: 0036-8075. DOI: [10.1126/science.253.5026.1374](https://doi.org/10.1126/science.253.5026.1374).
- [22] A. A. Griffith. *The Phenomena of Rupture and Flow in Solids*. 1921. DOI: [10.1098/rsta.1921.0006](https://doi.org/10.1098/rsta.1921.0006).

- [23] György Hantal et al. “Atomic-scale modelling of elastic and failure properties of clays”. In: *Molecular Physics* 112.9-10 (Mar. 2014), pp. 1294–1305. ISSN: 0026-8976. DOI: [10.1080/00268976.2014.897393](https://doi.org/10.1080/00268976.2014.897393). URL: <http://www.tandfonline.com/doi/abs/10.1080/00268976.2014.897393>.
- [24] Chih-Ming Ho. “Microflows and Nanoflows Fundamentals and Simulation”. In: *Physical Review E*. Vol. 67. 4. 2004, pp. 255–309. ISBN: 9780387221977. DOI: [10.1103/PhysRevE.67.041106](https://doi.org/10.1103/PhysRevE.67.041106). URL: <http://link.aps.org/doi/10.1103/PhysRevE.67.041106>.
- [25] J. Hövelmann et al. “Direct nanoscale observations of CO₂ sequestration during brucite [Mg(OH)₂] dissolution”. In: *Environmental Science and Technology* 46.9 (2012), pp. 5253–5260. ISSN: 0013936X. DOI: [10.1021/es300403n](https://doi.org/10.1021/es300403n).
- [26] Robert W. Howarth, Renee Santoro, and Anthony Ingraffea. *Methane and the greenhouse-gas footprint of natural gas from shale formations*. 2011. DOI: [10.1007/s10584-011-0061-5](https://doi.org/10.1007/s10584-011-0061-5).
- [27] W. Humphrey, A. Dalke, and K. Schulten. “Visual Molecular Dynamics”. In: *Journal of Molecular Graphics* 14 (1996), pp. 33–38. ISSN: 0263-7855. DOI: [10.1016/0263-7855\(96\)00018-5](https://doi.org/10.1016/0263-7855(96)00018-5). URL: <http://www.ks.uiuc.edu/Research/vmd/>.
- [28] Indiana University of Pennsylvania. *No Title*. URL: <http://www.iup.edu/page.aspx?id=25053>.
- [29] Jacob N. Israelachvili and Gayle E. Adams. *Measurement of forces between two mica surfaces in aqueous electrolyte solutions in the range 0?100 nm*. 1978. DOI: [10.1039/f19787400975](https://doi.org/10.1039/f19787400975).
- [30] J. E. Jones. “On the Determination of Molecular Fields. II. From the Equation of State of a Gas”. In: *Proceedings A* 106.738 (1924), p. 463. DOI: [10.1098/rspa.1924.0082](https://doi.org/10.1098/rspa.1924.0082). URL: <http://rspa.royalsocietypublishing.org/content/106/738/463>.
- [31] William L Jorgensen. “Revised TIPS for simulations of liquid water and aqueous solutions”. In: *The Journal of Chemical Physics* 77.8 (1982), p. 4156. ISSN: 00219606. DOI: [10.1063/1.444325](https://doi.org/10.1063/1.444325). URL: <http://link.aip.org/link/?JCP/77/4156/1%5C&Agg=doi>.
- [32] Pradeep Kumar, Sergey V Buldyrev, and H Eugene Stanley. “A tetrahedral entropy for water.” In: *Proceedings of the National Academy of Sciences of the United States of America* 106.52 (2009), pp. 22130–22134. ISSN: 0027-8424. DOI: [10.1073/pnas.0911094106](https://doi.org/10.1073/pnas.0911094106).
- [33] J. L. Laugesen. “Density functional calculations of elastic properties of portlandite, Ca(OH)₂”. In: *Cement and Concrete Research* 35.2 (2005), pp. 199–202. ISSN: 00088846. DOI: [10.1016/j.cemconres.2004.07.036](https://doi.org/10.1016/j.cemconres.2004.07.036).

- [34] Ju Li, Dongyi Liao, and Sidney Yip. *Coupling continuum to molecular-dynamics simulation: Reflecting particle method and the field estimator*. 1998. DOI: [10.1103/PhysRevE.57.7259](https://doi.org/10.1103/PhysRevE.57.7259).
- [35] Michael W. Mahoney and William L. Jorgensen. “A five-site model for liquid water and the reproduction of the density anomaly by rigid, nonpolarizable potential functions”. In: *The Journal of Chemical Physics* 112.20 (2000), p. 8910. ISSN: 00219606. DOI: [10.1063/1.481505](https://doi.org/10.1063/1.481505). URL: <http://scitation.aip.org/content/aip/journal/jcp/112/20/10.1063/1.481505>.
- [36] 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* 105.43 (2001), pp. 9954–9960. ISSN: 10895639. DOI: [10.1021/jp003020w](https://doi.org/10.1021/jp003020w).
- [37] V. MARRY et al. *Equilibrium and electrokinetic phenomena in charged porous media from microscopic and mesoscopic models: electro-osmosis in montmorillonite*. 2003. DOI: [10.1080/00268970310001626432](https://doi.org/10.1080/00268970310001626432).
- [38] Glenn J. Martyna, Douglas J. Tobias, and Michael L. Klein. “Constant pressure molecular dynamics algorithms”. In: *The Journal of Chemical Physics* 101.5 (1994), p. 4177. ISSN: 00219606. DOI: [10.1063/1.467468](https://doi.org/10.1063/1.467468). URL: <http://scitation.aip.org/content/aip/journal/jcp/101/5/10.1063/1.467468>.
- [39] Sean McNamara, Eirik Flekkøy, and Knut Måløy. *Grains and gas flow: Molecular dynamics with hydrodynamic interactions*. 2000. DOI: [10.1103/PhysRevE.61.4054](https://doi.org/10.1103/PhysRevE.61.4054).
- [40] Valeria Molinero and Emily B. Moore. “Water modeled as an intermediate element between carbon and silicon”. In: *Journal of Physical Chemistry B* 113.13 (2009), pp. 4008–4016. ISSN: 15206106. DOI: [10.1021/jp805227c](https://doi.org/10.1021/jp805227c). arXiv: [0809.2811](https://arxiv.org/abs/0809.2811).
- [41] K Muralidharan et al. “Molecular dynamics studies of brittle fracture in vitreous silica: Review and recent progress”. In: *Journal of Non-Crystalline Solids* 351.18 (2005), pp. 1532–1542. ISSN: 00223093. DOI: [10.1016/j.jnoncrysol.2005.03.026](https://doi.org/10.1016/j.jnoncrysol.2005.03.026). URL: <http://linkinghub.elsevier.com/retrieve/pii/S0022309305002140>.
- [42] Tom Myers. “Potential Contaminant Pathways from Hydraulically Fractured Shale to Aquifers”. In: *Ground Water* 50.6 (2012), pp. 872–882. ISSN: 0017467X. DOI: [10.1111/j.1745-6584.2012.00933.x](https://doi.org/10.1111/j.1745-6584.2012.00933.x).
- [43] Evgeniy M. Myshakin et al. “Molecular Dynamics Simulations of Carbon Dioxide Intercalation in Hydrated Na-Montmorillonite”. In: *The Journal of Physical Chemistry C* 117.21 (May 2013), pp. 11028–11039. ISSN: 1932-7447. DOI: [10.1021/jp312589s](https://doi.org/10.1021/jp312589s). URL: <http://pubs.acs.org/doi/abs/10.1021/jp312589s>.

- [44] Ken-ichi Nomura et al. *Interaction and coalescence of nanovoids and dynamic fracture in silica glass: multimillion-to-billion atom molecular dynamics simulations*. 2009. doi: [10.1088/0022-3727/42/21/214011](https://doi.org/10.1088/0022-3727/42/21/214011).
- [45] Dr. Neal R. O'Brien. *December Newsletter*. 2011. URL: http://www.clarkson.edu/camp/reports%5C_publications/dec11/page7.html.
- [46] R Pit, H Hervet, and L Leger. “Friction and slip of a simple liquid at a solid surface”. In: *Tribology Letters* 7.2-3 (1999), pp. 147–152. ISSN: 00431648. doi: [10.1023/a:1019161101812](https://doi.org/10.1023/a:1019161101812).
- [47] Michael C. Pitman and Adri C T Van Duin. “Dynamics of confined reactive water in smectite clay-zeolite composites”. In: *Journal of the American Chemical Society* 134.6 (2012), pp. 3042–3053. ISSN: 00027863. doi: [10.1021/ja208894m](https://doi.org/10.1021/ja208894m).
- [48] Steve Plimpton. “Fast Parallel Algorithms for Short-Range Molecular Dynamics”. In: *Journal of Computational Physics* 117.1 (1995), pp. 1–19. ISSN: 00219991. doi: [10.1006/jcph.1995.1039](https://doi.org/10.1006/jcph.1995.1039). URL: <http://www.sciencedirect.com/science/article/pii/S002199918571039X>.
- [49] J P Ryckaert, G Cicotti, and H J C Berendsen. “Numerical-Integration of Cartesian Equations of Motion of a System with Constraints - Molecular-Dynamics of N-Alkanes”. In: *Journal of Computational Physics* 23 (1977), pp. 327–341. ISSN: 00219991. doi: [10.1016/0021-9991\(77\)90098-5](https://doi.org/10.1016/0021-9991(77)90098-5). URL: [http://dx.doi.org/10.1016/0021-9991\(77\)90098-5](http://dx.doi.org/10.1016/0021-9991(77)90098-5).
- [50] David E Shaw et al. “Millisecond-scale molecular dynamics simulations on Anton”. In: *Proceedings of the Conference on High Performance Computing Networking Storage and Analysis SC 09* c (2009), p. 1. doi: [10.1145/1654059.1654099](https://doi.org/10.1145/1654059.1654099). URL: <http://portal.acm.org/citation.cfm?doid=1654059.1654099>.
- [51] Adarsh Shekhar et al. “Nanobubble collapse on a silica surface in water: Billion-atom reactive molecular dynamics simulations”. In: *Physical Review Letters* 111.18 (2013). ISSN: 00319007. doi: [10.1103/PhysRevLett.111.184503](https://doi.org/10.1103/PhysRevLett.111.184503).
- [52] Shell. *Unlocking tight and shale gas*. URL: <http://www.shell.com/global/future-energy/natural-gas/gas.html>.
- [53] Simon Less Simon Moore. *Gas Works? Shale gas and its policy implications*. Tech. rep. Policy Exchange, 2012. URL: <http://www.policyexchange.org.uk/images/publications/gas%20works%20-%20feb%2012.pdf>.
- [54] C H Sondergeld et al. “SPE 131771: Micro-structural studies of gas shales”. In: *SPE Annual Technical Conference and Exhibition*. 2010, p. 17. ISBN: 9781555632878. doi: [10.2118/131771-MS](https://doi.org/10.2118/131771-MS).

- [55] A.K. Soper and M.G. Phillips. *A new determination of the structure of water at 25C.* 1986. DOI: [10.1016/0301-0104\(86\)85058-3](https://doi.org/10.1016/0301-0104(86)85058-3).
- [56] John G. Spray. *Frictional Melting Processes in Planetary Materials: From Hypervelocity Impact to Earthquakes.* 2010. DOI: [10.1146/annurev.earth.031208.100045](https://doi.org/10.1146/annurev.earth.031208.100045).
- [57] Paul Stevens. “The ‘Shale Gas Revolution’ : Developments and Changes”. In: *Chatham House Briefing Paper* August (2012), p. 12.
- [58] Alexander Stukowski. “Visualization and analysis of atomistic simulation data with OVITO - the Open Visualization Tool”. In: *Modelling and Simulation in Materials Science and Engineering* 18.1 (2010), p. 015012. ISSN: 0965-0393. DOI: [10.1088/0965-0393/18/1/015012](https://doi.org/10.1088/0965-0393/18/1/015012).
- [59] Filip Sund. “Water confined in nanoporous silica”. PhD thesis. Universitetet i Oslo, 2014. URL: <https://www.duo.uio.no/handle/10852/41122>.
- [60] Sami Tazi et al. *Diffusion coefficient and shear viscosity of rigid water models.* 2012. DOI: [10.1088/0953-8984/24/28/284117](https://doi.org/10.1088/0953-8984/24/28/284117). arXiv: [arXiv: 1204.2501v1](https://arxiv.org/abs/1204.2501v1).
- [61] C.M. Tenney and R.T Cygan. “Molecular Simulation of Carbon Dioxide, Brine, and Clay Mineral Interactions and Determination of Contact Angles”. In: *Environ. Sci. Technol.* 3.48 (2014), pages. URL: <http://pubs.acs.org/doi/abs/10.1021/es404075k>.
- [62] J. Tersoff. “New empirical approach for the structure and energy of covalent systems”. In: *Physical Review B* 37.12 (1988), pp. 6991–7000. ISSN: 01631829. DOI: [10.1103/PhysRevB.37.6991](https://doi.org/10.1103/PhysRevB.37.6991).
- [63] John A. Thomas and Alan J H McGaughey. “Water flow in carbon nanotubes: Transition to subcontinuum transport”. In: *Physical Review Letters* 102.18 (2009). ISSN: 00319007. DOI: [10.1103/PhysRevLett.102.184502](https://doi.org/10.1103/PhysRevLett.102.184502).
- [64] S Timoshenko and J N Goodier. *Theory of Elasticity.* Vol. 49. 1986, pp. 427–143. ISBN: 0070858055. DOI: [10.1007/BF00046464](https://doi.org/10.1007/BF00046464). URL: <http://www.amazon.com/Elasticity-McGraw-Hill-Classic-Textbook-Reissue/dp/0070858055>.
- [65] Berend Smit Ting Chen and Alexis T. Bell. “Are pressure fluctuation-based equilibrium methods really worse than nonequilibrium methods for calculating viscosities?” In: *J. Chem. Phys* 131.24 (2009). URL: <http://scitation.aip.org/content/aip/journal/jcp/131/24/10.1063/1.3274802>.
- [66] Quy Dong To et al. “Molecular dynamics simulations of pressure-driven flows and comparison with acceleration-driven flows”. In: *Advances in Mechanical Engineering* 2012 (2012). ISSN: 16878132. DOI: [10.1155/2012/580763](https://doi.org/10.1155/2012/580763).

- [67] C Tropea, A L Yarin, and J F Foss. *Springer Handbook of Experimental Fluid Mechanics*. Vol. 46. 2007, pp. 2653–2655. ISBN: 9783540251415. DOI: [10.2514/1.38773](https://doi.org/10.2514/1.38773).
- [68] U.S. et al. *Technically Recoverable Shale Oil and Shale Gas Resources: An Assessment of 137 Shale Formations in 41 Countries Outside the United States*. Tech. rep. U.S. Department of Energy, 2013. URL: <http://www.eia.gov/analysis/studies/worldshalegas/pdf/fullreport.pdf>.
- [69] 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).
- [70] P. Vashishta et al. “Interaction potential for SiO₂: A molecular-dynamics study of structural correlations”. In: *Physical Review B* 41.17 (1990), pp. 12197–12209. ISSN: 01631829. DOI: [10.1103/PhysRevB.41.12197](https://doi.org/10.1103/PhysRevB.41.12197).
- [71] Jianwei Wang, Andrey G Kalinichev, and R James Kirkpatrick. “Molecular modeling of water structure in nano-pores between brucite (001) surfaces1”. In: *Geochimica et Cosmochimica Acta* 68.16 (2004), pp. 3351–3365. ISSN: 00167037. DOI: [10.1016/j.gca.2004.02.016](https://doi.org/10.1016/j.gca.2004.02.016). URL: <http://linkinghub.elsevier.com/retrieve/pii/S0016703704001668>.
- [72] Mineral Web. *Silica structure*. URL: <http://www.minweb.co.uk/quartz/beridata.html>.