

# Efficient workflows in molecular dynamics simulations and applications

by

Anders Hafreager

**THESIS**

*for the degree of*

***Philosophiae Doctor***



*Prepared under supervision from the*  
***Faculty of Mathematics and Natural Sciences***  
*University of Oslo*

*March 6, 2019*

© **Anders Hafreager, 2019**

*Series of dissertations submitted to the  
Faculty of Mathematics and Natural Sciences, University of Oslo  
No. 2158*

ISSN 1501-7710

All rights reserved. No part of this publication may be  
reproduced or transmitted, in any form or by any means, without permission.

Cover: Hanne Baadsgaard Utigard.  
Print production: Representralen, University of Oslo.

# Contents

<b>Contents</b>	<b>iii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Structure of the thesis . . . . .	3
<b>2 Atomic simulations</b>	<b>5</b>
2.1 The world at the atomic scale . . . . .	6
2.2 Molecular dynamics . . . . .	8
2.3 The Vashishta potential . . . . .	9
2.4 Tools and workflow . . . . .	10
2.4.1 Stage I: Initial configuration . . . . .	11
2.4.2 Stage II: Simulation . . . . .	12
2.4.3 Stage III: Post-processing and analysis . . . . .	12
2.4.4 What's missing? . . . . .	13
<b>3 Software development</b>	<b>15</b>
3.1 LAMMPS . . . . .	17
3.1.1 Acceleration techniques . . . . .	17
3.1.2 GPU package . . . . .	18
3.1.3 KOKKOS package . . . . .	19
3.2 My open source contributions . . . . .	19
<b>4 Theory of crystals</b>	<b>23</b>
4.1 Terminology . . . . .	25
4.2 Wulff construction . . . . .	28
4.3 Facet formation . . . . .	31
4.4 Surface diffusion . . . . .	32
4.5 Energy barriers . . . . .	32
<b>5 Summary of publications</b>	<b>35</b>
<b>6 Outlook</b>	<b>39</b>
<b>Bibliography</b>	<b>41</b>

<b>Publications</b>	<b>49</b>
1 Effective workflow in molecular dynamics simulations using Atomify — a real-time LAMMPS visualizer	49
2 Game-Engine-Assisted Research platform for Scientific computing (GEARS) in Virtual Reality	67
3 Direct atomic simulations of facet formation and equilibrium shapes of SiC nanoparticles	75
4 Structural replication of nanoporous media using procedural noise	95

# Acknowledgements

The past four years have been truly amazing, and I have zero regrets on pursuing a PhD. The work in this thesis was carried out at Physics of Geological Processes (PGP) at the Department of Physics at the University of Oslo from 2014 through 2018. Being around people who are so passionate about physics is incredible. Thanks to Statoil who funded this work through the Tight Rocks project.

My first encounter with someone passionate about physics was in high school when I had the pleasure of having Knut Løvseth as teacher in both mathematics and physics. He clearly loves physics and teaching, and was an excellent teacher. I remember when I by accident evaluated  $\ln(-1)$ , and  $i\pi$  showed up on my calculator with no warning at all. I was stunned, because in my mind, there was no reason why the numbers  $\pi$  and  $e$  should be related. I asked him about it, and he brought an old book for the next class. He then showed me complex numbers and how these are related to  $\pi$  through the unit circle. He has been a true inspiration to me throughout my many years as a teaching assistant, supervisor and researcher. Thank you Knut, this thesis would not have happened without you.

My career path actually started many years before high school. In 1998, at the age of 11, I won a Packard Bell computer in Donald Duck & Co. Having my own computer from early on has been tremendously useful. The year after, my dad showed me how to write Visual Basic scripts in Excel and I have loved programming since. I'm grateful to you, dad, that you introduced me to what now is my favorite thing to do.

And to my mom and my stepdad, who always supported me in my choices: thank you so much for always being there for me :) You truly understood my passion for computers, and helped me whenever I needed help, although you were a bit worried that I spent too much time in front of the computer. You weren't wrong, but it turned out ok. I'm sorry about that one time when we<sup>1</sup> got a 10000 NOK phone bill because I had played Planetarion online during night time:D

My intense interest for computers and games was shared with my two best friends from childhood Pål Einar Storsveen and Ole-Kristian Øien. We have been to so many LAN's and played endless amounts of computer games. Thank you guys, it has been fantastic.

Later, during my bachelor, I met Andreas Nakkerud. Wow, what a journey it has been. We have played a ridiculous amount of computer games over the years - especially Minecraft, OpenTTD and Counter-Strike. But there is so much more than that. We've had so many interesting, deep discussions about physics and taught several courses together. And let's not forget about all the hammock trips to the woods and cabins.

I shared an apartment with Andreas, Richard Rørmark, Sindre Aarsaether, Bedeho Mender and Andreas Våvang Solbrå. We all shared the joy (and the decent amount of pain) of programming, and those years were truly amazing. In fact, Sindre, Andreas and I started a company named TapCat in 2011 to develop apps for the iPhone. That we failed is an understatement, but I have zero regrets. I learned so much, and had a great time living the life as a startup without money in Silicon Valley for 2 months. Sindre and Andreas: it was great and I love that we did this together.

After the failed startup, I went back to university to start a masters at computational physics with Anders Malthe-Sørenssen as supervisor. That was one of the best

---

<sup>1</sup>By we, I mean my parents

choices I have ever made in my life. He has been a fantastic supervisor who has believed in me and given me the freedom to follow my true passion. Anders has also been my main supervisor on my PhD, together with Dag Kristian Dysthe. Thank you so much for all help and inspiring discussions!

Anders introduced me to the group Collaboratory for Advanced Computing and Simulations (CACs) group at University of Southern California (USC) when I was doing my masters. At USC, I met Rajiv Kalia, Priya Vashishta and Aiichiro Nakano, whom I have had the pleasure of working with on multiple research projects over the past two years. Spending three months there in 2017 has really shaped me as a researcher, and I have a much deeper insight in statistical mechanics thanks to numerous discussions and hard questions.

During my master's degree, I also met and worked with Sverre Arne Dragly, whom I've continued working with throughout my PhD. We both share a deep interest for programming and visualizations. We love discussing how programming can be done great (I have a long way to go). I really hope to be able to continue working with you in the future.

I share an office with Kjetil Thøgersen<sup>2</sup> and Henrik Sveinsson. Thank you so much for these years. Numerous hours of *rubber ducking* and coffee breaks have really been great. Henrik, I want to add an extra thanks to you. I will miss and remember our trips to Los Angeles for the rest of my life. And you know, everything else :)

Teaching has been one of my greatest pleasures during my time at the university. I want to thank all the student's I have been so lucky to meet and talk to. Fysikkforeningen and Lillefy, you have a big place in my heart:)

Alexandra, you have no idea how awesome you are. We have so much fun and you spread joy and happiness every day, and we will soon make an app called *Dumskalleprika* together. Thanks for being you!

Lastly, to my love Sunniva: the passion we share for physics, education and technology is *unbelievable*, and makes look forward to tomorrow every single day. Our mornings with coffee (with complete silence the first 10 minutes, of course) are precious. The past weeks of my PhD I can't wait to share the rest of my life with you and Alexandra♥

Sunniva and Alexandra: I love you both to the moon and back<sup>3</sup>.

*Anders Hafreager, Oslo, June 30, 2018*

---

<sup>2</sup>Important quote from the summer we first met: *Jeg har faatt en forespoersel fra Grete om en Thøgersen som sies aa jobbe paa datalabben. [...] naa skal han visst ha betaling for hele semesteret. Hvem er det og hva gjelder det?*

<sup>3</sup>Which is more than you when you handed in yours! The moon is today approximately 406 061 km away from the earth, but when you handed in yours it was only about 365 353 km away :D

# Introduction

## 1.1 Motivation

If our civilization would die out, all our knowledge and information would be lost. Chances are, however, that future life would evolve and maybe start figuring out science all over again. If only one sentence should be passed on to the next generation of life, what would it be? Richard Feynman wrote about this in the introductory chapter to his famous book series *The Feynman lectures on physics*. Here, he argues, that the sentence should be something along the lines of[25]

All things are made of atoms - little particles that move around in perpetual motion, attracting each other when they are a little distance apart, but repelling upon being squeezed into one another. - Richard Feynman

This is a powerful summary of physics that has not been known by humanity for much longer than a century - about the age of the oldest living humans on earth. Pick any macroscopic object, and this is a true statement for all gases, liquids and solids. Although it is a simple idea, the physics of atoms is very complex. An atom has a charged nucleus that results in electrons liking to be nearby. These electrons must follow strict rules about what places they can be in<sup>1</sup>. Once the electrons position themselves this way, the interaction between two or more atoms becomes very complicated.

There is an incredible amount of effects on the atomic scale that have macroscopic consequences. The melting temperature of ice is around 273.15 K, whereas silicon carbide melts at 3100 K[35]. What determines this? And why do diamonds, quartz and table salt have that macroscopic shape with clear facets - flat faces on the surface? Today we do know the answer to these questions, and they all happen to depend on details on the atomic scale.

The dynamics of atoms follow the Schrödinger equation - a complicated equation that only has a handful known exact solutions. Approximate methods like Hartree-Fock[34] and *Density Functional Theory* (DFT)[47] can be used to find solutions for more advanced systems using a computer<sup>2</sup>. However, even with today's computers,

---

<sup>1</sup>The electron *places* are the allowed quantum numbers  $n, l, m_l$ .

<sup>2</sup>The Hartree-Fock methods were actually originally used to calculate approximate solutions *without* computers in the 1920s.

these methods only work efficiently for small systems up to a few thousand atoms[61], so other methods have to be used for larger systems.

In this thesis I focus on *molecular dynamics* - a model where atoms are assumed to follow the laws of classical mechanics, given some interatomic potential. These interatomic potentials are usually found using quantum mechanical methods like DFT, and are fitted so that they reproduce macroscopic properties. Molecular dynamics is a numerical model that we use to run *numeriments*<sup>3</sup> on a computer. The model produces atomic trajectories, on which we apply statistical mechanics to explain thermodynamic quantities and macroscopic properties.

In traditional experiments, we cannot follow the trajectories of every single atom, but with atomic simulations like molecular dynamics, we can understand phenomena on a very detailed level. The rapid increase of computing power and new algorithms opens up the possibility to solve new, and more complicated problems each year. Supercomputers have been used to run molecular dynamics simulations with several billion atoms[86], and will eventually catch up with some experiments. Most of today's simulation software and analysis tools have been built around the idea that the system preparation and running the simulation are separated processes, but this idea does not reflect the actual workflow.

The scope of this thesis is improving the workflow in molecular dynamics simulations. Molecular dynamics simulations are often divided into a three-step process where we first prepare an initial configuration from which the simulation starts. Secondly, the simulation is performed, before we analyze the data as the third step. Preparing initial geometries for simulations can be tricky and is very important since this may greatly affect the outcome. Molecular dynamics simulations are often used to understand phenomena from experiments.

One example that I have worked on in this thesis is the geometry of nanoporous materials. A lot of the world's energy production comes from hydrocarbons trapped inside tight rocks, and understanding how fluid flows inside such materials has been a popular field of study[19, 39, 40]. Studying these materials in simulations requires us to be able to represent the geometry on a computer. Three-dimensional representations can for instance be captured using expensive *focused ion beam scanning electron microscopy* (FIB-SEM). This method destroys the material[16, 18]. Other methods include generating geometries from statistical and geometrical characterizations.

In this thesis, I have developed a method using procedural noise functions[50, 69, 70] that can be trained to reproduce geometries that are statistical similar to a reference geometry. The geometries used in this work were produced by molecular dynamics simulations, but it could just as well have been a real sample from FIB-SEM. Once the optimal parameters for the noise model are found, generating statistical similar geometries is quickly done, and can be used for further studies.

When I started learning molecular dynamics, the learning curve was steeper than what is necessary because several different applications are needed to run and analyze full simulations. Some physical phenomena are sensitive to initial conditions, and running several simulations with trial and error is essential to find the optimal parameters. Since the human brain is excellent at interpreting visual representations of data, immediate visual feedback would be very useful, but usually isn't available in existing software. Laptops and desktop computers are today powerful enough to perform simulations that were state of the art only 20 years ago. Therefore, I have developed

---

<sup>3</sup>A numeriment is a numerical experiment - an experiment performed on a computer[79].



*Atomify* - an application that enables real-time visualization of running molecular dynamics simulations.

With the game industry pushing forward on the development of *virtual reality* (VR), this technology has been improved tremendously lately. VR opens for new possibilities also in scientific research. The perception of being inside a simulation surrounded by atoms filling the complete field of view may be useful to understand physical and chemical processes. Interacting with a simulation while it runs is possible, and new branches of problems can now be studied. In this thesis I have also developed an environment for running simulations in real-time using VR using game engines<sup>4</sup> like Unity and Unreal Engine.

Of course, such tools are only here to help answering scientific questions. In this thesis I have also performed molecular dynamics simulations to study facet formation in silicon carbide nanocrystals. Faceting is a type of crystal growth that happens due to reorganizations of surface atoms through surface diffusion[12, 22]. Surface diffusion is strongly temperature dependent: at low temperatures, surface diffusion is too slow for faceting to happen on molecular dynamics time scales, and at high temperatures, faceting may not be present due to surface melting, or the system being above the roughening transition temperature[109]. That me and my co-authors were able to study this phenomenon turned out to come as a direct consequence of immediate feedback from Atomify while searching for the optimal temperature.

## 1.2 Structure of the thesis

The thesis is organized as follows. Chapter 2 gives an overview of atomic simulations with a brief discussion on the world on the atomic scale in section 2.1. Section 2.2 introduces molecular dynamics in more detail with the mathematical formulation and a discussion on interatomic potentials. The Vashishta potential is discussed in more detail in section 2.3, since this has been used extensively in this thesis.

Chapter 3 is about software development and how this is an integral part of modern scientific research. This chapter addresses how changes in modern hardware affects software development, with extra focus on LAMMPS - the molecular dynamics software package I have used in this thesis. How to obtain faster molecular dynamics simulations is discussed. In addition, I go through my contributions to the source code of LAMMPS.

Chapter 4 is devoted to the physics of crystal growth. I go through some background and terminology, before the Wulff construction is defined and explained. Then, how facets form in nanocrystals through surface diffusion is discussed. Chapter 5 is a summary of my publications in this thesis, and chapter 6 presents some thoughts and ideas for future research.

All four publications are included at the end of the thesis. Publication 1, 2 and 4 have been submitted for review, whereas publication 3 is being prepared for submission.

---

<sup>4</sup>Game engines are programming environments used to produce games. See [https://en.wikipedia.org/wiki/Game\\_engine](https://en.wikipedia.org/wiki/Game_engine) for more details.



# CHAPTER 2

## Atomic simulations

Computer science inverts the normal. In normal science, you're given a world, and your job is to find out the rules. In computer science, you give the computer the rules, and it creates the world.

---

*Alan Kay*

Nature is a very complex system with complicated rules that determine how and where mass, energy and momentum is transferred. In a sense, this is the only thing that happens in the universe. Quantum mechanics define the rules, and the macroscopic properties we observe are explained by statistical mechanics. Many observable macroscopic effects result from details on the subatomic scale. Such macroscopic effects are everything from melting of ice, to something so great and complex as life itself. Both are possible due to quantum mechanical rules that regulate what makes an atomic nucleus stable, and how electrons interact with the nucleus. The quantum mechanical framework allows us to explain and predict how the distribution of single electrons make some molecules stable, while others are not. Hydrogen and oxygen happily form water molecules due to valence bonding, filling their valence shells.

Water has several unusual properties. For instance, that ice floats in water is explained by the fact that the hydrogen bonds form perfect tetrahedra in the solid phase. This tetrahedral structure has lower density than liquid water at 0°C[52]. This is remarkable because most solids are denser than their corresponding liquids and gases. Despite being one of the most studied substances, water is still poorly understood[68].

In principle, we do have the mathematical toolbox to explain nearly all phenomena that happens on Earth<sup>1</sup>, but solving the Schrödinger equation for many particles<sup>2</sup> simultaneously is a challenging task[83]. Although it is a difficult problem, understanding the world from a bottom-up perspective is essential to be able to model physical systems on the atomic scale.

---

<sup>1</sup>Although we can only explain 4% of the matter in the universe, so there is quite a job remaining until I'm satisfied.

<sup>2</sup>Many here actually means more than one.

The work in this thesis is about molecular dynamics. In molecular dynamics, the motion of a set of atoms is calculated using Newton's second law in a classical force field. The full phase space coordinates are known at all times, and statistical properties can be calculated from averages, assuming ergodicity[54]. Molecular dynamics has been proved successful in reproducing and explaining many phenomena, such as mechanical properties [14, 42, 63], chemical reactions [7, 101] and properties of many liquids[4].

High frequency vibrations of light atoms like hydrogen defines the the timescale of atomic motion, and is of order femtoseconds. While some processes like chemical reactions may occur within a few picoseconds or less, some biological processes occur over timescales of more than a millisecond[84]. Simulating such systems with femtosecond temporal resolution thus needs  $10^{12}$  timesteps. To beat the millisecond barrier, specialized molecular dynamics hardware has been built - a so-called *Application Specific Integrated Circuit* (ASIC)<sup>3</sup> machine called Anton[84]. This machine has been used to study protein structures in water[46, 85], which requires very long simulation times.

Much has happened in computing since the first molecular dynamics simulation of hard spheres in the 1950s[2, 3], and more realistic systems using the Lennard Jones potential[43] in the 1960s[75]. Since then, the computing power has seen a tremendous increase. From the CDC 3600<sup>4</sup> used in [75], which performed less than 1 million instructions per second, to modern 3 GHz processors in typical laptops that can calculate about 100 billion operations per second. With this rapid increase of computing power, the length and time scales achievable in numeriments are finally catching up with experiments.

Desktop machines and laptops are now more powerful than many supercomputers only 20 years ago, but the workflow has not changed much. In this chapter, I discuss briefly the bottom-up view of our world from the smallest building blocks governed by the rules of quantum mechanics, to larger systems where Newton's laws can be applied. Molecular dynamics is introduced with some extra effort on potentials - especially the Vashishta potential[102], which has been used in multiple projects throughout this thesis. Lastly, I discuss the workflow of a typical molecular dynamics research project is executed and how this can be improved.

## 2.1 The world at the atomic scale

The smallest building blocks in the universe are, as far as we know, the particles in the standard model (and probably some others explaining the largely unknown dark matter). It consists of six quarks, six leptons, the force carrying bosons and the Higgs boson. Interaction between them arises from simple<sup>5</sup> symmetries. The up and down quarks combine to create protons and neutrons, which make up the atomic nucleus surrounded by electrons.

These atoms may come in contact with other atoms, and potentially form molecules and larger crystals. Atoms are generally quantum mechanical objects, but many properties can correctly be described classically given the right interatomic potential. In classical mechanics, the state of a system is defined by a point in phase space that changes over time through an equation of motion. The phase space point is the set of the positions  $\mathbf{r}$  and momenta  $\mathbf{p}$  for all  $N$  atoms, making up the canonical coordinates

---

<sup>3</sup>ASIC machines are built to be good at a specific task and cannot do general computing.

<sup>4</sup>CDC 3600 was one of several computers built by Control Data Corporation in the mid-1960s.

<sup>5</sup>Only a madman would call the mathematics simple, but the symmetries are at least simple.

$\mathbf{q}$

$$\mathbf{q} = (\mathbf{r}, \mathbf{p}) = (\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N, \mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_N) \quad (2.1)$$

where  $\mathbf{r}_i$  is the position of atom  $i$  and  $\mathbf{p}_i$  is its momentum. The energy is found through the Hamiltonian

$$\mathcal{H}(\mathbf{r}, \mathbf{p}) = K(\mathbf{p}) + V(\mathbf{r}) = \sum_{i=1}^N \frac{|\mathbf{p}_i|^2}{2m_i} + V(\mathbf{r}), \quad (2.2)$$

where  $K(\mathbf{p})$  is the kinetic energy,  $m_i$  is the mass of atom  $i$  and  $V(\mathbf{r})$  is the potential energy in the system. This Hamiltonian defines the time evolution of the system through Hamilton's equations

$$\frac{d\mathbf{p}}{dt} = -\frac{\partial \mathcal{H}}{\partial \mathbf{r}}, \quad \frac{d\mathbf{r}}{dt} = \frac{\partial \mathcal{H}}{\partial \mathbf{p}}. \quad (2.3)$$

In real life, knowing the positions and momenta for every single atom in a system is of course not feasible, so we use statistical mechanics to help us describe rather *ensembles* - a collection of states with given probabilities. It links the ensemble of states to thermodynamic quantities and is very useful when doing atomic simulations.

For instance, in the canonical ensemble<sup>6</sup>, each state  $(\mathbf{r}, \mathbf{p})$  has a probability determined by its energy

$$\rho(\mathbf{r}, \mathbf{p}) = \frac{1}{Z} \exp\left(-\frac{\mathcal{H}(\mathbf{r}, \mathbf{p})}{kT}\right), \quad (2.4)$$

where  $k$  is Boltzmann's constant and  $T$  is the temperature. The normalization  $1/Z$  comes from integrating over all states

$$Z = \int_{\mathbf{r}} \int_{\mathbf{p}} d\mathbf{r} d\mathbf{p} \rho(\mathbf{r}, \mathbf{p}), \quad (2.5)$$

but is usually not practical to find. A thermodynamic quantity  $A$  is obtained through an ensemble average, i.e. integrating over all possible states and evaluating the quantity  $A$  multiplied by the probability of finding the system in that state

$$\langle A \rangle_{\text{Ensemble}} = \int_{\mathbf{r}} \int_{\mathbf{p}} d\mathbf{r} d\mathbf{p} A(\mathbf{r}, \mathbf{p}) \rho(\mathbf{r}, \mathbf{p}). \quad (2.6)$$

The ergodic hypothesis states that *the time spent in a region of phase space around  $(\mathbf{r}, \mathbf{p})$  is proportional to  $\rho(\mathbf{r}, \mathbf{p})$* . This is useful since we usually do not know the probability density  $\rho(\mathbf{r}, \mathbf{p})$ . We do, however, know how a system will evolve using equation (2.3). The ensemble average is assumed to, for long times, be equal to the time average so that

$$\langle A \rangle_{\text{Ensemble}} = \langle A \rangle_{\text{Time}} = \frac{1}{N} \sum_{i=1}^N A(\mathbf{r}^{(i)}, \mathbf{p}^{(i)}), \quad (2.7)$$

---

<sup>6</sup>The canonical ensemble is also known as the  $NVT$  ensemble. Here the system has a constant number of particle  $N$ , volume  $V$  and temperature  $T$

where  $N$  is the number of independent<sup>7</sup> measurements, and  $A(\mathbf{r}^{(i)}, \mathbf{p}^{(i)})$  is the value of observable  $A$  at measurement  $i$ .

Numerous macroscopic quantities can be found using this technique and I refer to [4] and [26], both of which are excellent books on this topic. Having these mathematical tools available, we can set up systems and create trajectories in the phase space using molecular dynamics which is discussed next.

## 2.2 Molecular dynamics

The idea of molecular dynamics is so simple that anyone accepting that everything is made up of atoms and knowing Newton's second law will understand the basics. Although quantum mechanics describes the rules at the atomic scale, we assume in molecular dynamics that the motion of atoms is described by classical mechanics using Hamilton's equations. We start with some initial state  $\mathbf{q} = (\mathbf{r}, \mathbf{p})$ .

A typical interatomic potential (or force field) is usually decomposed into a sum of two-particle contributions, three-particle contributions and so on[4]

$$V(\mathbf{r}) = \sum_{i,j} V^{(2)}(\mathbf{r}_i, \mathbf{r}_j) + \sum_{i,j,k} V^{(3)}(\mathbf{r}_i, \mathbf{r}_j, \mathbf{r}_k) + \dots \quad (2.8)$$

From Hamilton's equations, we find the forces in the system since the kinetic energy term only depends on  $\mathbf{p}$

$$\frac{d\mathbf{p}}{dt} = -\frac{\partial \mathcal{H}}{\partial \mathbf{r}} = -\frac{dV}{d\mathbf{r}}, \quad (2.9)$$

which then becomes

$$\mathbf{F}(\mathbf{r}) = -\nabla V(\mathbf{r}) \quad (2.10)$$

$$= \sum_{i,j} -\nabla V^{(2)}(\mathbf{r}_i, \mathbf{r}_j) + \sum_{i,j,k} -\nabla V^{(3)}(\mathbf{r}_i, \mathbf{r}_j, \mathbf{r}_k) \quad (2.11)$$

$$= \sum_{i,j} \mathbf{F}^{(2)}(\mathbf{r}_i, \mathbf{r}_j) + \sum_{i,j,k} \mathbf{F}^{(3)}(\mathbf{r}_i, \mathbf{r}_j, \mathbf{r}_k) \quad (2.12)$$

for a three-particle potential. Without any external forces, the sum of all forces is zero, so we need to look at the forces  $\mathbf{f}_i$  acting on each atom  $i$ .  $\mathbf{f}_i$  is found as

$$\mathbf{f}_i(\mathbf{r}_i) = -\partial_{\mathbf{r}_i} V(\mathbf{R}), \quad (2.13)$$

where  $\partial_{\mathbf{r}_i} = (\partial_{x_i}, \partial_{y_i}, \partial_{z_i})$  and  $\partial_x \equiv \frac{\partial}{\partial x}$  is shorthand for partial derivatives.

Most potentials consist of general terms reflecting known physical effects. For instance, take the Lennard Jones potential[43]. It is only a pairwise potential of the interatomic distance  $r$  between two atoms

$$V(r) = 4\epsilon \left[ \left( \frac{\sigma}{r} \right)^{12} - \left( \frac{\sigma}{r} \right)^6 \right]. \quad (2.14)$$

Here, the  $r^6$  and the  $r^{12}$  terms represent the van der Waals attraction force and steric repulsion respectively. The 12th power in the repulsion term has no physical meaning,

---

<sup>7</sup>Independent means that the measurements should be performed with some time delay in between.

but is numerically is cheap to compute once the van der Waals  $r^6$  term is calculated. The parameters  $\sigma$  and  $\epsilon$  can be fitted to reproduce many properties of noble gases[53], but is also used in several other systems[4].

In a system with ions, the effect of Coulomb interaction is often important. This term goes like  $1/r$ , and is considered a long-term interaction since it decays very slowly, compared to other terms, for large  $r$ . In periodic systems, a particle at position  $\mathbf{r}_i$  will interact with an infinite array of images of the simulation box, including itself, multiple times. To efficiently compute forces with long-range interactions, one often needs to use special algorithms of the type *long-range solvers*. This group of algorithms contains the Ewald summation[26] method and PPPM (particle-particle particle-mesh)[37], both of which exploit properties of the Fourier transform in periodic systems. Charges can either be fixed throughout the simulation, or be adjusted during the simulation using charge equilibration as described in [76] and [59].

Developing interatomic potentials is a field of study of its own. One of the most advanced force fields that exists today is called ReaxFF, and was originally developed to study chemical reactions[101]. It includes more physical effects than Lennard Jones, and its original form is given as

$$E_{\text{system}} = E_{\text{bond}} + E_{\text{over}} + E_{\text{under}} + E_{\text{val}} + E_{\text{pen}} + E_{\text{tors}} + E_{\text{conj}} + E_{\text{vdWaals}} + E_{\text{Coulomb}}, \quad (2.15)$$

and has a total of 93 parameters that need to be fitted. The conventional approach to fit these parameters is using an *ab initio* method like DFT[47, 101]. Then, the force field is often adjusted to reproduce macroscopic properties like the first sharp diffraction peak of the structure factor  $S(\mathbf{q})$ [103] and other relevant properties depending on the research interest.

All in all, a potential is a mathematical function designed to capture relevant physical concepts. In this thesis, I have used the Vashishta potential in publication 3 and 4. This potential is discussed next.

## 2.3 The Vashishta potential

Vashishta et. al. published around 1990 a potential to study amorphous GeSe<sub>2</sub>[102] and SiO<sub>2</sub>[103] in molecular dynamics. In both studies, the potential reproduced the structure factor  $S(\mathbf{q})$  to an impressive degree. The potential has been very successful in describing the mechanical properties of both crystalline and amorphous silica. The same functional form can be used for different elements as well, so the group has published a series of parameters for other oxides[105], ceramics[104] and aluminum nitride[106]. While being similar to the Stillinger-Weber potential[91], it includes several important differences in both the two-body term and the three-body term.

The most general functional form of the potential looks like<sup>8</sup>

$$V(\mathbf{r}) = \sum_{i,j} V_{ij}^{(2)}(r_{ij}) + \sum_{i,j,k} V_{ijk}^{(3)}(r_{ij}, r_{ik}, \theta_{ijk}), \quad (2.16)$$

where the two-body contribution is given as

$$V_{ij}^{(2)}(r) = \frac{H_{ij}}{r^{\eta_{ij}}} + \frac{Z_i Z_j}{r} e^{-r/\lambda_{1,ij}} - \frac{D_{ij}}{r^4} e^{-r/\lambda_{4,ij}} - \frac{W_{ij}}{r^6}, \quad r \leq r_{c,ij}. \quad (2.17)$$

---

<sup>8</sup>The original version did not contain the exponential screening of the Coulomb term or the  $C_{ijk}$  term in the three-body energy. For proper values of  $\lambda$  and  $C_{ijk}$  it reduces to the original form.

Each parameter has a subscript which comes from the fact that different pairs and triplets of elements have different parameter values. Here,  $H_{ij}$  is the strength of the steric repulsion (representing the same physical effect as the  $r^{12}$  term in Lennard Jones). The second term is the Coulomb interaction with charges  $Z_i$  and  $Z_j$ . This term contains an exponential screening factor with parameter  $\lambda_{1,ij}$ . We also have a charge-induced dipole interaction with strength  $D_{ij}$ , which also is exponentially screened with another parameter  $\lambda_{4,ij}$ . Then, lastly, the van der Waals interaction with interaction strength  $W_{ij}$ .  $r_c$  is the cutoff distance, the distance at which the force and the energy is zero.

Having a cutoff introduces an error. This error is compensated with a shifted term with linear force interpolation towards  $r_c$

$$V_{ij}^{(2 \text{ shifted})}(r) = \begin{cases} V_{ij}^{(2)}(r) - V_{ij}^{(2)}(r_c) - (r - r_c) \left[ \frac{dV_{ij}^{(2)}}{dr} \right]_{r=r_c} & r \leq r_c \\ 0 & r > r_c. \end{cases} \quad (2.18)$$

The number of atomic neighbors scales as the volume of the surrounding sphere of radius  $r_c$ , so the neighbor count scales as  $r_c^3$ . The screening factors in the potential then increases the computational efficiency by an order of magnitude[105], because it allows a much smaller cutoff distance  $r_c$ . It has been shown that these screening factors do not change the structural or dynamic properties of the material[60].

The three-body term is evaluated for all triplets  $i, j, k$  within the three-body cutoff, and is given as

$$V^{(3)}(r_{ij}, r_{ik}, \theta_{ijk}) = B_{ijk} \frac{\left[ \cos \theta_{ijk} - \cos \theta_{ijk}^0 \right]^2}{1 + C_{ijk} \left[ \cos \theta_{ijk} - \cos \theta_{ijk}^0 \right]^2} \times \exp\left(\frac{\gamma_{ij}}{r_{ij} - r_{0,ij}}\right) \exp\left(\frac{\gamma_{ik}}{r_{ik} - r_{0,ik}}\right), \quad r_{ij} \leq r_{0,ij}, \quad r_{ik} \leq r_{0,ik} \quad (2.19)$$

$$= P^{(3)}(\theta_{ijk}) \times R^{(3)}(r_{ij}, r_{ik}), \quad (2.20)$$

where I have factored out the spatial part  $R^{(3)}(r_{ij}, r_{ik})$  and the angular part  $P^{(3)}(\theta_{ijk})$ . The spatial part describes the bond stretching and the angular part represents bond bending.  $B_{ijk}$  is the strength of the interaction,  $\theta_{ijk}$  is the angle between vectors  $\mathbf{r}_{ij}$  and  $\mathbf{r}_{ik}$  and  $\theta_{ijk}^0$  is the equilibrium angle discussed above. Compared to the Stillinger-Weber potential, the two-body term of this potential includes more physical terms. The three-body term is nearly identical except the  $C_{ijk}$  term in the denominator, which is used to control the stiffness of the angular term.

In this thesis, the Vashishta potential has been used to generate the nanoporous structures in publication 4, and to study faceting of silicon carbide (SiC) nanoparticles in publication 3.

With this short introduction to molecular dynamics, I will in the next section discuss a typical workflow while doing molecular dynamics simulations. This workflow includes working with several different applications and tools.

## 2.4 Tools and workflow

A typical molecular dynamics simulation can be thought of as a process divided into three different stages. First, an initial state from which the system will start from has



to be created. This state contains the topology of the system (atom positions, and possibly bonds between them), initial velocities, charges and boundary conditions. It can be a tricky step and the time spent on preparing the system is often underestimated in advanced simulations.

Secondly, the simulation itself is performed, often with some professionally written software. Some analysis may be performed on the fly during the simulation<sup>9</sup>, but a third stage, the analysis of atomic trajectories and thermodynamic data is quite common after the simulation is finished.

These three stages define a typical workflow of atomic simulations, and has not changed much over the past 20 years despite the remarkable technological advances. With powerful desktop machines and laptops, visualizations and analysis can be performed locally, but the three stages are still decoupled. In this section I go into detail about each step with some examples from my own research and discuss how there is room for improvement in modern research.

### 2.4.1 Stage I: Initial configuration

Many simulations are simple to set up. Measuring bulk properties of a solid can be quickly prepared if the solid is a crystal with known lattice structure. The most common lattices, simple cubic (SC), face centered cubic (FCC), body centered cubic (BCC), hexagonal close-packed (HCP) and diamond, are already built-in features in most molecular software packages. These lattices describes most monoatomic crystals at room temperature in the periodic table[45], and bulk systems can be setup within minutes.

However, crystals consisting of multiple elements often exist in more complicated structures. Having initial configurations matching those found in nature is essential to be able to compare numeriments to experiments, since deviations can to a large degree affect the results. For instance, the melting temperature of perfect crystals in molecular dynamics simulations are expected to be higher than experimental bulk values since samples in nature are never perfect crystals, nor do they obey periodic boundary conditions[104]. Melting is often initiated at grain boundaries or crystal defects which act as melting seeds[71].

Setting up advanced systems with complicated topology often requires special software like Moltemplate[41]. Moltemplate is a scripting based molecule builder that create larger systems from building blocks. Complex bond topology and advanced interfaces are among the examples in the software package.

Some advanced systems are those resulting from complicated physical processes. One such example which I have studied in this thesis is nanoporous media. These systems consist of complicated pore networks found in nature. The physics of fluids in tight rocks is important since most of the hydrocarbons used for energy production are found in such systems. Continuum hydrodynamics breaks down when the channels are very narrow[44], so particle models like molecular dynamics are often used to study flow in nanopores. Even without flow, water confined in nanopores have shown to appear in new phases using molecular dynamics[87]. Preparation of such systems is complicated, and is the main focus of publication 4 where we have used procedural noise methods to create statistically similar geometries to those generated by computing intensive molecular dynamics methods.

---

<sup>9</sup>In very large simulations, this may be essential since the amount of data can be enormous.

Other systems may not be geometrically as hard to set up, but the behaviour of some systems are sensitive to initial conditions such as temperature. This was the case in publication 3 where several initial conditions were tested to see whether or not we could capture faceting of nanocrystals in molecular dynamics. Faceting is the process where surface atoms reorganize resulting in possibly a polyhedron (this is discussed further in chapter 4 and in publication 3). This process is highly temperature sensitive, because at too high temperatures, the system may melt the nanoparticle. A too low temperature may prevent faceting to happen on time scales available in simulations. Then, once the initial configuration is done, the simulation can be started.

## 2.4.2 Stage II: Simulation

Once the initial configuration is properly set up, the simulation itself may not be hard to perform. Existing advanced software packages exists and the one I have used in this thesis is discussed more in detail in section 3.1. An initial configuration is integrated through time using a numerical integration scheme like *velocity Verlet*[96], and trajectories can be saved to disk with any desired frequency. Aggregate scalar quantities such as temperature, stress or the mean square displacement can be calculated on the fly and stored to files for further analysis after the simulation is finished.

Many physical properties can be measured in equilibrium molecular dynamics - where the states (hopefully<sup>10</sup>) are sampled in a certain ensemble like  $NVE$ ,  $NVT$  or  $NPT$ . In these simulations, the system is first equilibrated before we start to perform sampling of states. Finding the diffusion coefficient, viscosity or thermal conductivity in a system can be done in equilibrium molecular dynamics using Green-Kubo relations[4, 26], but many properties require the simulation to be performed at non-equilibrium conditions such as when studying flow or irreversible processes.

In fluid flow simulations, the fluid is moving caused by an external pressure difference[44]. Several techniques exist to achieve a pressure difference. The simplest method is used in the case study in publication 1, where a constant force is applied on the atoms. Other more realistic methods may have reservoirs on both ends of the transport channels where fixed pressure boundary conditions are obtained by adding or removing atoms in the reservoirs[95]. Setting up and performing such simulations often require a lot of trial and error, and visual feedback is essential to work efficiently.

Atomify (publication 1) is a tool developed to reduce the gap between preparing the initial state and the simulation itself. It enables real-time visualization of the simulation, where adjustments to the simulation script can quickly be tested with immediate visual feedback.

## 2.4.3 Stage III: Post-processing and analysis

Once the simulation is done, the analysis stage begins. This stage is another big portion of the total time spent on a molecular dynamics research project. It is not always clear from the beginning what type of analysis one should perform and rerunning simulations to produce data not originally stored is common. As an example, in publication 3, we did not have enough temporal resolution to measure the diffusion coefficients from the saved trajectories, so we had to rerun the simulation from several timesteps.

---

<sup>10</sup>Ensuring that the system actually *does* follow trajectories in the desired ensemble can be tricky, and several techniques exists for doing so, see [54, 65, 82] for details.

Also here, visual feedback is crucial, and is often performed locally on an office computer or laptop after the data has been transferred from a supercomputer. In the beginning, molecular dynamics simulations were performed without visualizations, and only atomic positions and velocities were analyzed to extract i.e. diffusion coefficients and correlation functions[33, 75]. This is only natural since computers at the time did not necessarily have a display.

Later, once computers began to have screens, numerous visualization tools were developed to visualize molecular systems. Here I will only discuss the most popular ones used today, but I refer to [55] for a review of different tools.

In 1996, *Visual Molecular Dynamics* (VMD) was released as a tool to visualize and analyze atomic simulations with main focus on proteins and nucleic acids[38]. It supports displaying trajectories of atoms and molecules, but also interacting real-time with a running instance of molecular dynamics softwares such as *Nanoscale Molecular Dynamics* (NAMD)[62]. VMD can be controlled with mouse clicks in the *Graphical User Interface* (GUI), but also by writing scripts in the *Tcl* programming language. High quality images can be rendered to produce publication-ready figures, while also supporting high performance real-time rendering using *Open Graphics Library* (OpenGL). However, the GUI is outdated, and the *Tcl* language is used less today. This opens up for new, more modern tools to enter the market.

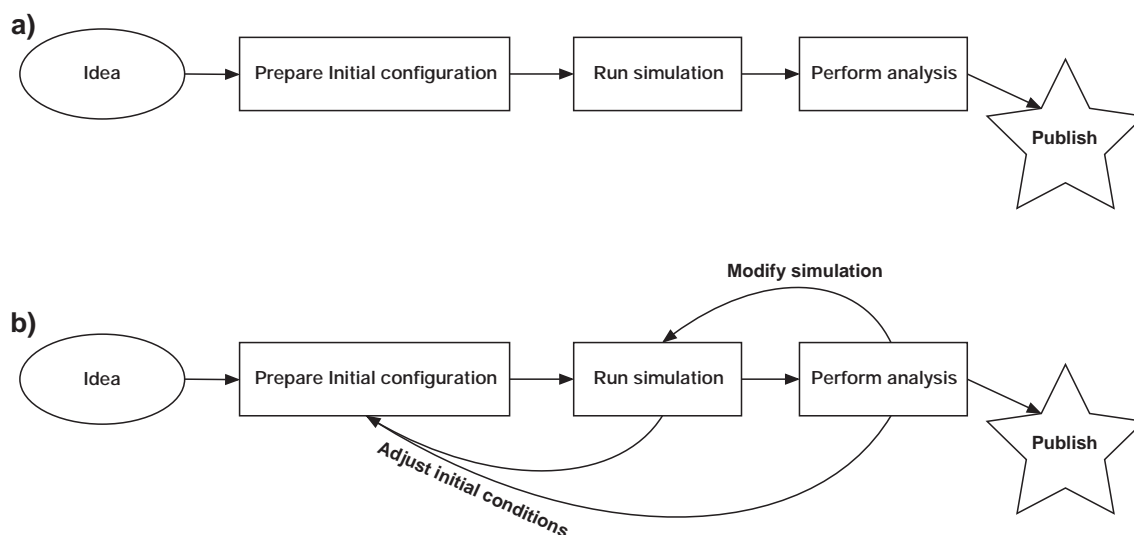
In 2009, another visualization tool called Ovito was released[92]. Ovito is a modern visualization tool that supports many file formats[94] and advanced analysis through its powerful modifier pipeline. A series of *modifiers* can be applied. Such modifiers can be system replication, slicing, displacement analysis (which can be used to color atoms) and more advanced techniques like coordination analysis and *Common Neighbor Analysis* (CNA) analysis[23, 93] to identify crystal structures. See [94] for a full list of features. Most of these features are also available in a flexible Python *Application Programming Interface* (API), which enables efficient post-processing analysis workflows. If Ovito lacks a specific analysis method, a user can write custom Python modifiers as a part of the pipeline.

#### 2.4.4 What's missing?

The workflow defined as these three stages are quite powerful and has been used in many published papers. However, in my experience, the three stages are too separated compared to how the creative mind works throughout a simulation. By separated, I mean that they often involve using different applications and switching back and forth between these many times. In figure 21 the idealized workflow is shown, compared to how a research project often actually is performed. If the initial configuration is created by one application, and another is used to visualize the result, the context switch can reduce efficiency and productivity.

Compare this to doing programming. It is quite common to use an *integrated development environment* (IDE) - a software application that is more than just a text editor. The IDE knows the programming language syntax, and can both compile and run your program within the same environment. Similarly in video editing, sound and video are tightly bound, and all professional video editing softwares support editing both.

In molecular dynamics, the simulation conditions are usually defined in a script before the simulation starts, but is not a part of the file containing the initial geometry. Boundary conditions and possibly integration rules in different regions are commands



**Figure 21:** The idealized workflow shown in a). An idea leads to preparation of an initial configuration from which a simulation starts from. Then a simulation is performed and results are analyzed before the paper is published. Usually, the workflow is rather like shown in b). An idea leads to preparation of an initial configuration. Then, a simulation is performed and you realize that the initial configuration somehow is *wrong*, and you need to change it. This can happen many times even before a full simulation is finished. Then, once the results are produced, you might get new insight so that you once again have to change the initial configuration or simulation parameters. Repeat this process multiple times and you *might* end up writing a paper.

to the simulator software which usually does not support real-time visualization in an efficient manner. For beginners, this is especially hard because the context switches may increase the learning barrier.

This is why I created Atomify. Atomify is a suggested solution to reduce the gap between the three stages, but is not intended to replace existing post-processing softwares like VMD or OVITO. I don't mean that they are not necessary - especially since Atomify only runs on a local computer and cannot be used for very large simulations. It can, however, be used to glue together the separated initial conditions and preparing the initial geometry.

Currently, it does not yet support virtual reality. VR has by 2018 really reached a new milestone with several high quality, relatively cheap headsets available for everyone. Although the initial release of VMD supported stereoscopic rendering, which could be used to visualize 3D images using glasses of some kind[38], tilting and rotating your head did not change the camera position, and is then not necessarily considered VR. Today, with the Oculus Rift and HTC Vive, we can get full VR perception since the sensors track not rotation and tilting of your head, but also walking and crouching. For systems like the nanoporous materials in publication 4, being able to be *inside* the systems and observe the pore networks is very useful. Other researchers have also started developing similar tools[17], but these are so far limited in simulation size.

In publication 2, we have used existing game engines and coupled them to simulation softwares to visualize simulations in VR environments. With hand controllers, we can interact with the system in new ways. I expect a lot of new tools using VR to be developed over the next few years.

## Software development

I have always wished for my computer to be as easy to use as my telephone; my wish has come true because I can no longer figure out how to use my telephone.

---

*Bjarne Stroustrup*

Most modern research involve some software development. It is common to write small scripts to automate tasks, and create figures for publications. Running thousands of simulations with different input parameters can be done with simple scripts. However, developing a molecular dynamics code that can compete with existing codes is very difficult. The number of different algorithms and methods is large. Some of the most popular codes have been developed for more than 20 years[62, 73] and contain several hundred thousand lines of code each.

Several open source molecular dynamics softwares exist, such as LAMMPS[73], NAMD[62], GROMACS[74] and OpenMM[20], and have hundreds of contributors[29–31] that help improving the codes each year. Molecular dynamics simulations are often performed on supercomputers to obtain large systems and perform long simulations[86]. They should therefore show excellent scaling across thousands of processors[64, 74], which may not be a trivial programming task.

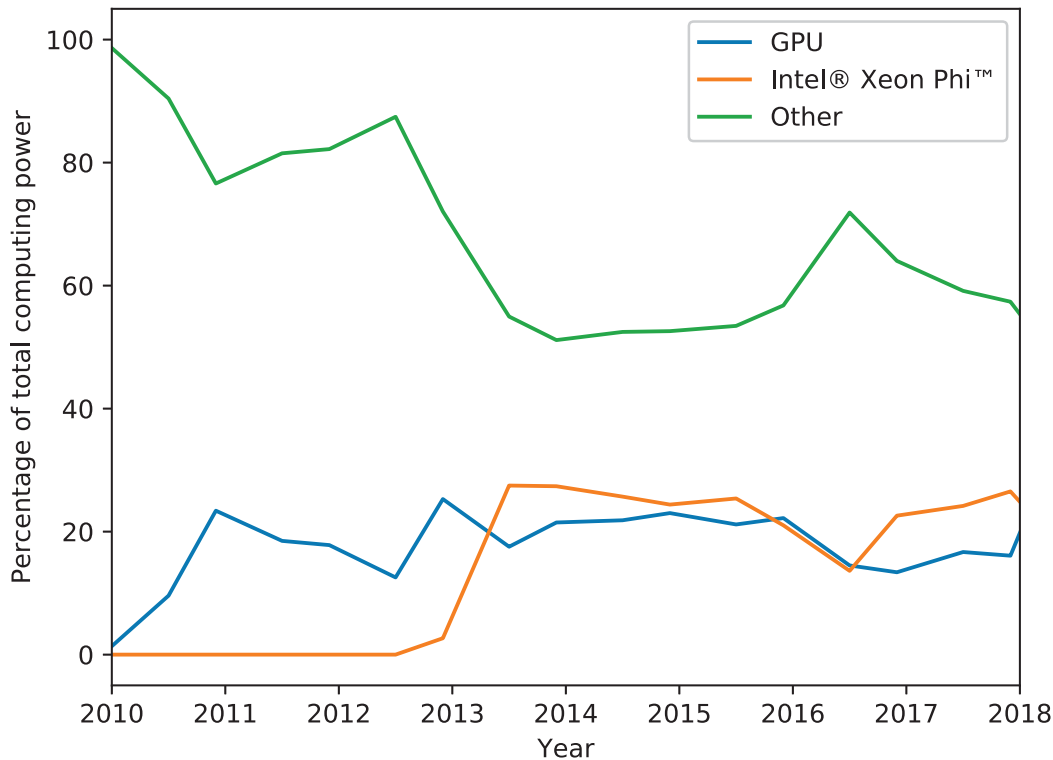
Development of simulation software often has two different focus areas: new features and performance optimizations. Among new features we find new potentials[99], and methods like spin coupling to molecular dynamics[67]. The latter was only recently implemented in LAMMPS[100] with a new parallel implementation in order to be efficient in large simulations<sup>1</sup>. Adding support of modern hardware to existing codes also requires a lot of development.

Herb Sutter, one of our most famous C++ experts, wrote in 2005 in his article *The Free Lunch is Over!!!: “Concurrency is the next major revolution in how we write software.”*

What he is referring to is that while the transistor count has steadily followed Moore’s law, the clock speed of processors haven’t increased much since the 3 GHz

---

<sup>1</sup>I used this as an example since it was implemented into LAMMPS this year.



**Figure 31:** The percentage of total computing power being powered by GPU’s, Intel®Xeon Phi™’s or regular processors on the top 50 supercomputers since 2010. In 2010, there were only a few supercomputers powered by GPU’s and the first Intel®Xeon Phi™supercomputer came in the end of 2012. Writing software that utilizes this special hardware requires extra knowledge. Source: [top500.org](http://top500.org).

mark in 2005. The current state of the art 5 nm transistors are getting close to the physical limit, and must eventually stop shrinking. Since 2005, new processors have gotten faster by adding more cores and better vectorization<sup>2</sup>. Most of the increased computing power since 2005 is only available through exploiting this new programming model.

*Central Processing Unit* (CPU)’s are so-called general-purpose computing units that are reasonably fast at general tasks, but this also means it does not yield maximum performance at specialized tasks. When the computer game industry grew in the 1990s, new vendors pushed forward a new type of computing devices which were specialized in running the exact same code simultaneously to render many pixels on the screen. These devices are called GPU’s and have the past 10 years found their way into supercomputers. This is shown in figure 31 where the total computing power on the top 50 supercomputers is divided into three categories: GPU’s, Intel®Xeon Phi™’s and regular processors. Intel®Xeon Phi™is a new type of computing card produced by Intel®to compete with GPU vendors in the computing market.

<sup>2</sup>Vectorization, or *Single Instruction Multiple Data* (SIMD), means that the processor executes the same instructions (multiplication, square root, etc.) on multiple values on a single core at the same time.



I start this chapter by discussing why I have chosen *Large-scale Atomic/Molecular Massively Parallel Simulator* (LAMMPS) as molecular dynamics software and its basics. Different acceleration techniques are then discussed with extra attention on how GPU's can be used, since these may be much faster than regular processors. Then, I go through my contributions to the source code of LAMMPS - all of which have been motivated by my own scientific research and development.

## 3.1 LAMMPS

In this thesis all simulations have been performed with LAMMPS - one of the most popular molecular dynamics codes today. It was originally developed by Steve Plimpton at Sandia National Labs, but has today many contributors and maintainers. LAMMPS is open source, written in C++ and is easy to extend with new classes due to its elegant build system and class factory design. The code runs with excellent scaling on thousands of processor cores[80] and supports acceleration on GPU's and Intel®Xeon Phi™'s through accelerator packages.

LAMMPS was originally written in FORTRAN 77 in the 90's, with its first version released in 1995 [81]. It was rewritten to F90 in 2001, before Sandia in 2004 decided to switch to C++ and rewrote 50000 lines from F90 to C++. This is the codebase of which today's version is based on.

The software is very flexible with many supported features. A simulation is defined using an input script that is executed line by line. The scripting language is probably Turing complete<sup>3</sup>, and I have implemented a molecular dynamics code *within* the scripting language using only variables and loops[32]. *Computes* can be used to measure properties like temperature, coordination number, radius of gyration and mean square displacement (many more exist). *Fixes* are applied each timestep and are used to i.e. obtain certain ensembles as *NVE*, *NVT* or *NPT*, or perform advanced simulations measuring properties such as thermal conductivity, chemical reactions using *nudge elastic band* or running shock simulations.

A good molecular dynamics code should run efficiently and yield high performance throughput utilizing the available hardware. Many different techniques exist and some will be discussed next.

### 3.1.1 Acceleration techniques

Most of the computation time in a molecular dynamics simulation is spent on calculating forces, which typically are given as (eq. (2.12)):

$$\begin{aligned} \mathbf{F}(\mathbf{r}) &= \sum_{i,j} \mathbf{F}^{(2)}(\mathbf{r}_i, \mathbf{r}_j) + \sum_{i,j,k} \mathbf{F}^{(3)}(\mathbf{r}_i, \mathbf{r}_j, \mathbf{r}_k) \\ &= \sum_{i,j} \left( \mathbf{F}^{(2)}(\mathbf{r}_i, \mathbf{r}_j) + \sum_{k>j} \mathbf{F}^{(3)}(\mathbf{r}_i, \mathbf{r}_j, \mathbf{r}_k) \right). \end{aligned} \quad (3.1)$$

Distributing and optimizing this calculation is central for nearly all optimization techniques. I will here present some of the relevant techniques in modern molecular dynamics codes.

---

<sup>3</sup>I haven't proved this, but it seems plausible.

**MPI:** LAMMPS is programmed using *Message Passing Interface* (MPI) where the software is started as multiple processes, possibly on different computers, each having a unique identifier that is used so each process can work on a smaller part of the full problem. This is one of the few efficient techniques that allows a software to run on distributed memory systems (i.e. multiple computers) in supercomputer centers. When a molecular dynamics simulation is run on  $N$  processes, the full system volume is divided into  $N$  subvolumes, one per process. Each process owns, and does computations on atoms within its subvolume.

**Threads and OpenMP:** Within the same process, the program can run multiple *threads*, each executing code on different processor cores. Several threads can run sequences of instructions sharing the same memory. Equation (3.1) is programmed as a loop over  $i$  atoms. By dividing the  $i$  loop into multiple ranges, each running on different threads, we can use several cores within the same process. For instance, if we have 1000 atoms and 4 threads, thread 1 can calculate the forces for atoms 1-250 while thread 2 works with atoms 251-500 etc. In C++, access to threads can be obtained through `std::thread`, or using `OpenMP`.

**SIMD:** For every pair  $i$  and  $j$ , the exact same code will be executed, except perhaps whether or not the pair is within the cutoff or not. This is what SIMD is made for - performing Single Instruction on Multiple Data. Using the SIMD registers, we can obtain a substantial speedup since several neighbors can be computed simultaneously. The `USER-INTEL` package in LAMMPS is written by Michael Brown. This package contains specialized versions of several of the classes in LAMMPS written in a way exploiting SIMD and multiple cores.

**Force tables:** The Vashishta potential discussed in section 2.3 contains multiple trigonometric functions such as  $\exp$  and  $\cos$ , in addition to evaluating powers of  $r$ . Calls to these are expensive compared to regular floating point operations such as addition and multiplication. These function calls can be avoided by storing precomputed values of the forces and energies into an array with only atomic distance as input parameter. In the force loop, the squared distance is calculated for each atom pair. This distance is used to find the two nearest precomputed values and different interpolation schemes can be used to evaluate the force and energy to any desired accuracy.

**GPU's:** Apart from the techniques mentioned above, all of which affects regular implementations on CPU's, other computing models also exist. The usage of additional computing cards such as GPU's has become quite popular as can be seen in figure 31. These cards add an additional computing power that can be used in parallel with regular processors. The computing and memory model is often different, so the original code doesn't necessarily automatically run on these cards. A GPU can have thousands of cores[108], and code is run in massive parallel executing the same instructions on multiple data inputs (SIMD). While a high-end processor today can perform about 1 TFLOP, modern GPU's can perform more than 15 TFLOPS on single precision floating point numbers[66]. I implemented one version of the Vashishta potential in the `GPU` package, and one in the `KOKKOS` package. These packages, hence the implementations, are quite different and deserve their own subsections.

### 3.1.2 GPU package

The GPU package was initially written by Michael Brown[8–10] in 2010 as an accelerator package for LAMMPS. It was built as a way to accelerate the two heaviest part of a timestep: neighbor list building and force computations. All of the data is owned by



the CPU, which is also where time integration happens. Atom positions are transferred to the GPU each timestep to build neighbor list and calculate the forces, which then are transferred back to the CPU for time integration.

Each potential has to be written as CUDA kernels since this was the only supported alternative by NVIDIA in 2010 when the package was written. OpenCL is supported through a CUDA→OpenCL transpiler, so this package can also be used on cards produced by other vendors than NVIDIA. Since the GPU is only used for neighbor list building and force calculations, this package also works in Atomify.

### 3.1.3 KOKKOS package

When time goes by, new hardware or new programming language features comes, and new rewrites of a software become inevitable. This is time consuming and prevents the development of new scientific ideas or features. As a solution to this problem, Sandia started developing a new library called KOKKOS[21]. The title of their paper is *Kokkos: Enabling manycore performance portability through polymorphic memory access patterns*. This paper addresses the manycore revolution and how different platforms obey different memory models, which may require different code per platform.

KOKKOS solves this problem with a template-based C++ library where the programmer does not need to worry about the memory access pattern<sup>4</sup>. As developers, we only need to think about the memory as a regular, contiguous array, but this is just an abstraction over the underlying memory model which depends on the target platform. The same C++ code is used for GPU's, Intel®Xeon Phi™'s or the more standard `OpenMP` and `pthread`. When new hardware or new platforms appear years from now, the software written using KOKKOS (such as LAMMPS) does not need to be updated since it can be supported by changing the KOKKOS library only. This reduces the effort needed to keep the software relevant many years from now.

In the GPU package, the processor owns the data, and atomic positions are only transferred to the GPU to build neighbor lists and calculate forces. Transferring data is not free and adds an additional overhead which should be avoided if possible. If KOKKOS is used on GPU's, the data is owned by the device and is only transferred to the CPU if non-KOKKOS features are being used. In a simple *NVT* simulation, the data lives on the device and has zero overhead in data transfer throughout the simulation. KOKKOS is currently written using static state variables, and does not support multiple instances of the KOKKOS object, nor reallocation during the life time of the process.

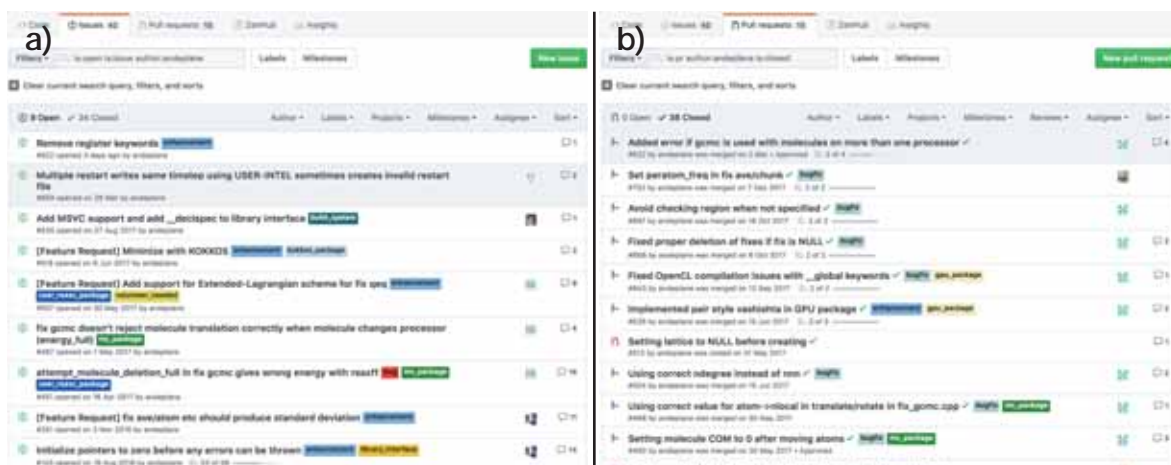
I have worked with both these packages and contributed to the LAMMPS source code which is discussed next.

## 3.2 My open source contributions

I have several contributions to the LAMMPS source code over the past years. As of May 22nd, 43 bug reports and 38 pull requests have been submitted as is shown in figure 32. Open source contributions often come from problems a user experiences, and this is also the case here.

---

<sup>4</sup>For instance, GPU's is faster using texture memory for random memory access, compared to the constant memory for contiguous access.



**Figure 32:** List of parts of my contributions to the LAMMPS software on GitHub. (a) A total of 43 issues has been submitted and (b) 38 pull requests have been submitted with multiple bugfixes, new features such as GPU implementations of the Vashishta potential and improvements on the GCMC code.

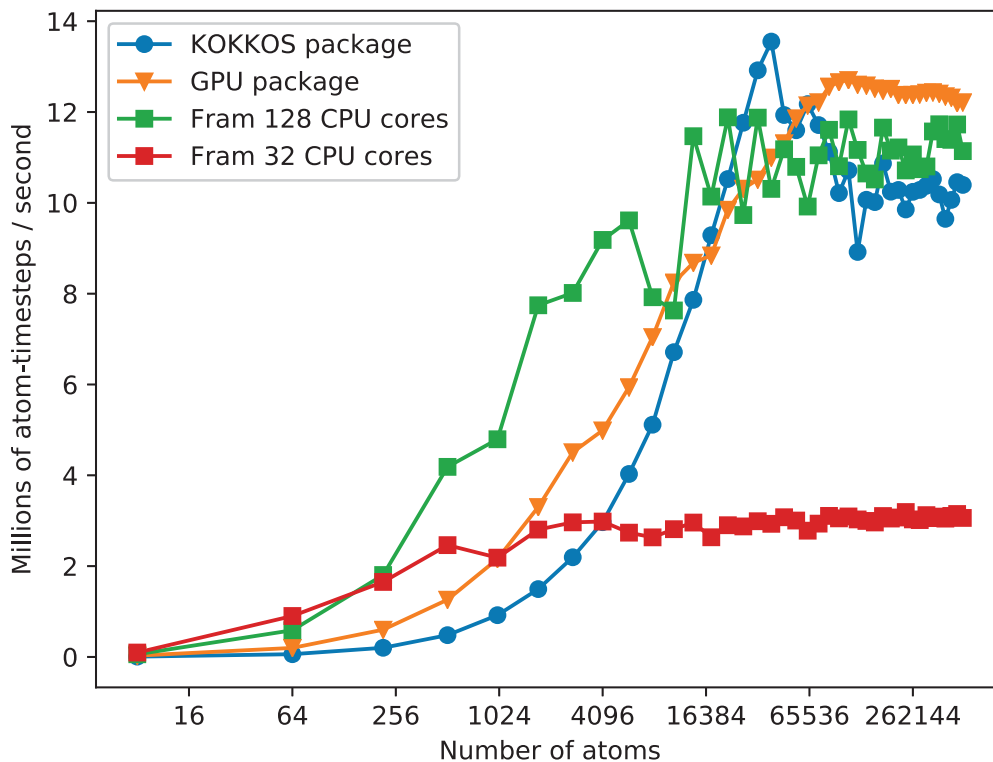
**Tabulation and cached neighbor lists:** The first contributions were optimizations on the Vashishta potential with tabulated two-body forces to prevent expensive function calls, but also cached neighbor lists for the three-body force loop. The tabulation is done using simple linear interpolation between the tabulated values, and gives a 60% speedup when using 50000 tabulation points compared to the analytical implementation.

The silicon carbide parameters for the Vashishta potential has a 7.36 Å cutoff for the two-body forces, but only 2.9 Å for the three-body forces[104]. Similar values are also found for the other materials[103, 105, 106]. The neighbor list is using the maximum cutoff value and will in the silicon carbide case consist of at least all atoms within 7.35 Å. A naïve implementation would loop through all these neighbors when calculating the three-body forces although only those within 2.9 Å are contributing.

To improve this, I implemented a cached neighbor list that contains only those atoms closer than three-body cutoff. This list is built each timestep while looping through the two-body forces. It is almost free and gives a great speedup (2x-4x). Later on, Axel Kohlmeyer implemented the same techniques in other **MANYBODY** potentials such as Stillinger-Weber[91] and Tersoff[97, 98]. We have now also implemented the cached neighbor lists in both the GPU package and the KOKKOS package.

**C++ exceptions:** LAMMPS was originally written so that the process exited if any errors had occurred. Such errors could be a syntax error in the user script, simulations losing atoms or some other invalid state. This was problematic for Atomify since if LAMMPS exits the process, this would also exit Atomify. What we want to happen is to print the error message to the user so she can quickly fix the problem and try again. Richard Berger, the main developer of the **Python** package, had similar problems when running LAMMPS in a Jupyter Notebook session.

As a solution to this, I suggested using `std::exception`. Exceptions are used to transfer control from where a problem occurred, to further up in the call stack where such an exception can be handled. This is a standard feature in many programming languages, C++ being one of them. LAMMPS now supports this and Atomify catches exceptions and presents error messages to users instead of exiting.



**Figure 33:** Benchmark on the two GPU implementations of the Vashishta potential (bulk 3C-SiC) using one NVIDIA Tesla P100 GPU, compared with 128 and 32 cores on Fram. Larger system size was achieved by replicating the system in all directions. The measure is atom-timesteps per second, a quantity that ideally should be constant since simulation time is  $\mathcal{O}(N)$  of number of atoms. We see that we need a rather large system to reach ideal performance which is expected since GPU's have many cores and require a substantial workload to perform ideally. Alternatively, multiple jobs can be run simultaneously to achieve good performance also on smaller systems.

**Variable initialization:** Once exceptions were supported, another related problem occurred. That the process will continue after a LAMMPS problem was detected was originally not in the developers minds. This was evident when a user of Atomify had an error in the input script. Most of the C++ classes in LAMMPS did not properly initialize variables. If an exception was thrown during object construction, pointers may have random values which resulted in segmentation fault during deallocation. Initializing pointers to `NULL`, and array lengths to zero before any script syntax validation was performed fixed these problems when running Atomify or LAMMPS in Jupyter Notebook. The standard classes and 33 out of 58 packages are now fixed and ready to use in Atomify.

**GPU implementations:** My scientific interests made me want to run the Vashishta potential on GPU's. Benchmarks showed that Stillinger-Weber had a great speedup on modern GPU's[80], and since the Vashishta potential is very similar to the Stillinger-Weber potential, it was reasonable to expect similar speedups also with this potential. My implementation was merged into the code in November 2016. As mentioned in sub-

section 3.1.3, the `KOKKOS` package does not support running LAMMPS as a library due to its global state. This is problematic for Atomify since it allocates and deallocates the LAMMPS object many times during one session. KOKKOS also only supports double precision, which typically gives a factor 2 performance decrease compared to single precision. The `GPU` package does support both single precision and library usage, so this package is ideal to use to do real-time simulations in Atomify. I then implemented the potential also in the `GPU` package. The implementation followed the style of existing similar potentials such as Stillinger-Weber. In figure 33, I show benchmarks on how my GPU implementations perform compared to 32 and 128 cores on our supercomputing cluster Fram. One single NVIDIA P100 card performs approximately the same as 128 cores on Fram using four Intel E5-2683v4 processors[89].

**Other bugfixes:** Several other bugfixes in different classes and packages has also been submitted. See <https://github.com/lammps/lammps/pulls?q=author%3Aandepiane> for a full list of pull requests.

# CHAPTER 4

## Theory of crystals

The beauty of crystals lies in  
the planeness of their faces.

---

*Alfred Edwin Howard Tutton*

We have used the methods developed in this thesis to address several physical phenomena in detail. The most extensive application is to facet formation in nanocrystals - the mechanism of transforming the shape of a nanoparticle into one with clear facets like shown in figure 41. Crystals are fascinating objects with an incredible amount of interesting physics in it. Depending on the number of electrons in the valence shell, atoms may organize in complicated structures that obey certain symmetries. This has several macroscopic consequences.

One of the text books describing crystal growth is *Physics of crystal growth* by Pimpinelli and Villain[72]. In figure 42, I have created a word cloud from the text in this book to extract the main concepts, and important terminology in a visual manner. The term *surface* appears three times more than any other technical<sup>1</sup> word which reveals that this is the main focus of the book.

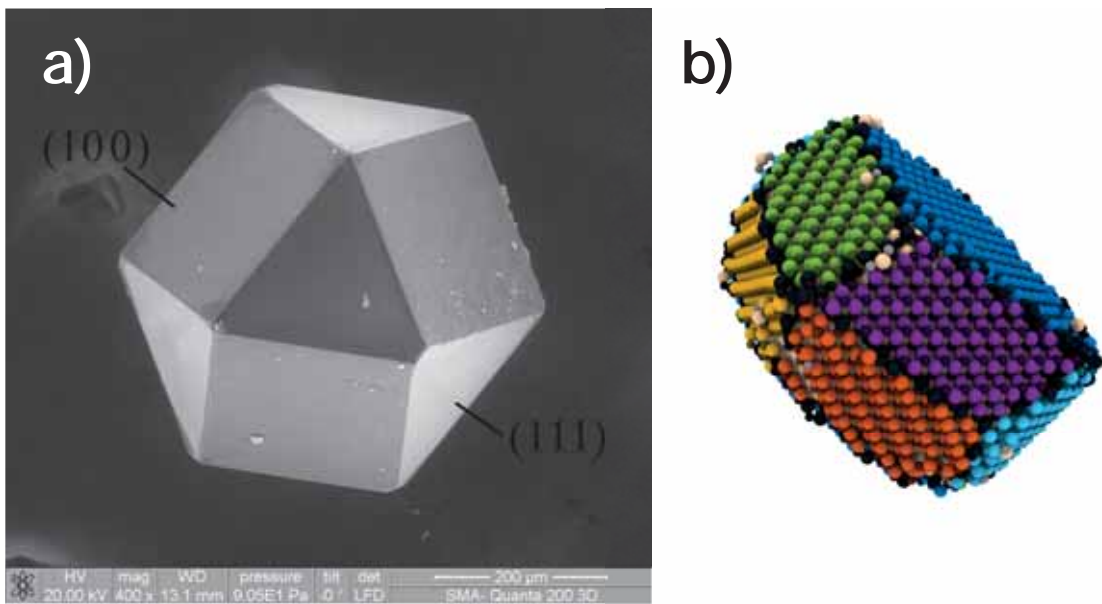
Crystals are everywhere. Almost the entire Earth's crust consists of crystalline rocks, and reveal important historical events that have happened on earth[107]. In fact, the physics of crystal surfaces has been a popular field of study for nearly a century with several groundbreaking papers. Most crystals grow from a solution or a melt. With the advent of *molecular beam epitaxi* (MBE, a technique to grow crystals layer by atomic layer in a controlled manner) and electron microscopy, the physics of crystal growth took a new step forward.

Understanding this process is also important the development of modern technology. Semiconductors are a class of materials that have electric conductivity between insulators and metals, but by doping<sup>2</sup> them, they get very interesting properties and can be used to create transistors and modern processors. Modern processors are grown from silicon crystals and doping is done by the surface being in contact with vapor. See [72] for more details.

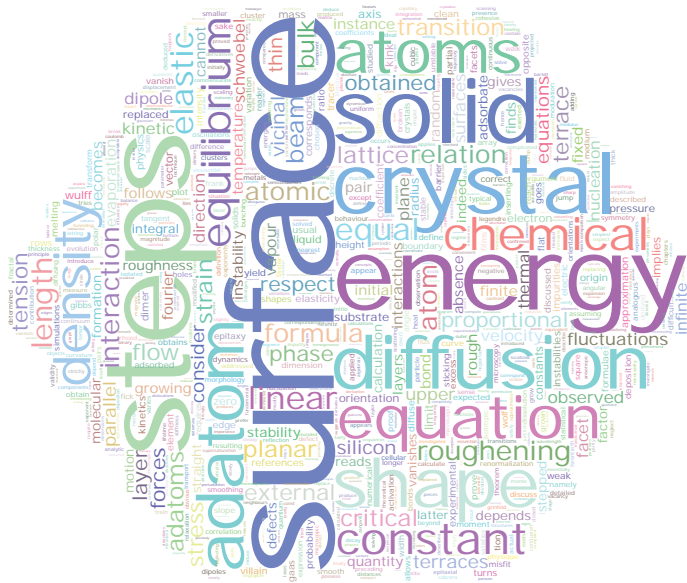
---

<sup>1</sup>I have removed the most common english words.

<sup>2</sup>Doping means adding impurities (other elements) to the material.

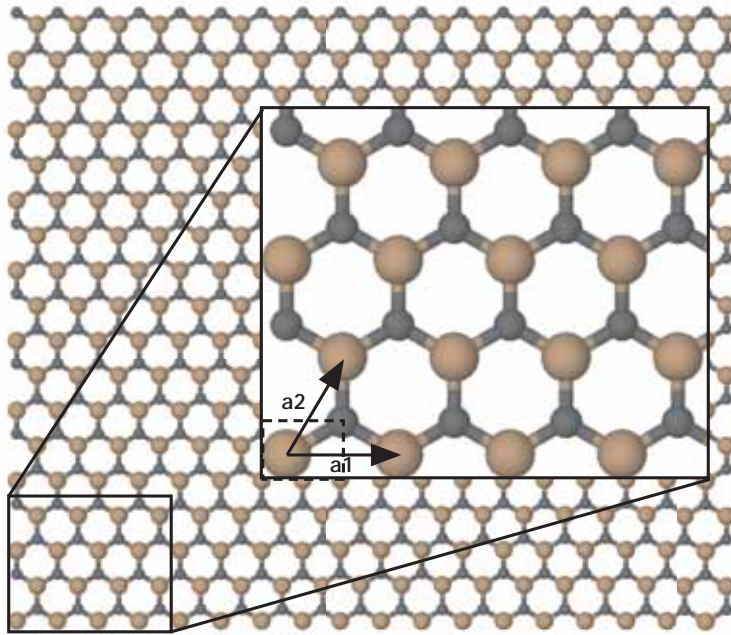


**Figure 41:** Crystals at low temperatures form facets - flat faces along crystal planes. a) (100) and (111) facets on a diamond cuboctahedron, image captured with scanning electron microscopy. The crystal has a diameter of almost 1 mm with clear facets. ©User:Ludvig14 / Wikimedia Commons / CC-BY-SA-3.0. b) Metastable faceted silicon carbide nanocrystal from publication 3. The colors have no direct meaning except highlighting the different (110) facets.



**Figure 42:** A word cloud generated from *Physics of crystal growth* by Pimpinelli and Villain[72]. The top 5 words are *surface*, *energy*, *crystal*, *diffusion* and *steps* which reveals important concepts in the field. Not surprisingly, these are all important terms in publication 3 where we study the transformation of nanocrystals from different initial shapes to shapes with facets.





**Figure 43:** Two-dimensional silicon carbide crystal. The two lattice vectors  $\mathbf{a}_1$  and  $\mathbf{a}_2$  are shown in the inset with the unit cell highlighted in the dashed box.

Physics often deals with questions like *what is the equilibrium state?* and *what is the path toward equilibrium?* For crystals, this translates to *what is the equilibrium shape?* and *how does the crystal grow or transform to obtain this stable shape?*

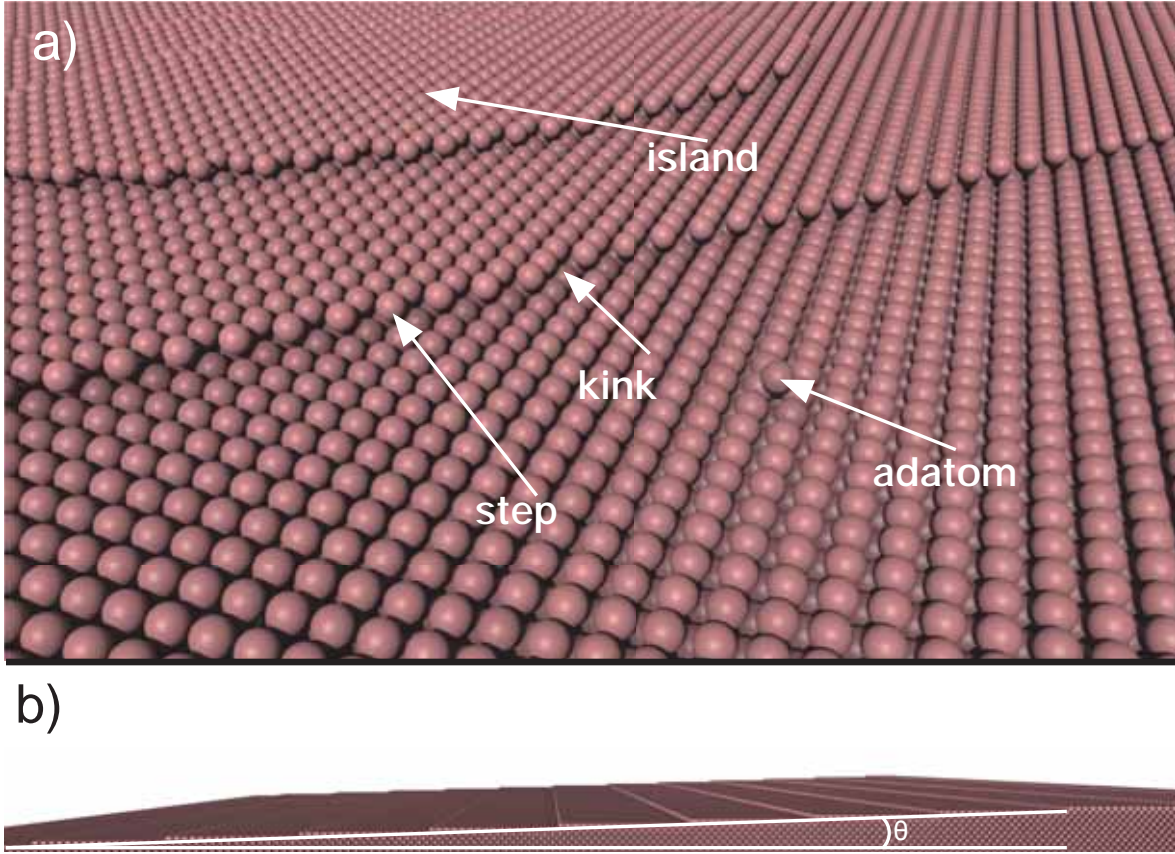
In this chapter, I briefly discuss what a crystal *is*, as well as some basic terminology and concepts that will be useful to understand why crystal equilibrium shapes in fact are variants of polyhedra with facets. I then discuss the Wulff construction[110] - a technique to construct the equilibrium shape of a crystal from the surface tension which can be measured experimentally. Some details about facet formation is then discussed before relevant topics like surface diffusion and energy barriers conclude the chapter. This chapter should not be seen as a review, but rather a collection of important results on the path to a understanding of the faceting phenomenon.

## 4.1 Terminology

A solid is a crystal if the atoms are placed a structured way on a lattice with translational symmetries[45]. In figure 43, I show a two-dimensional silicon carbide crystal with clear translational symmetries.

A *unit cell* consists of a set of atoms - building blocks - that can be replicated along *lattice vectors*  $\{\mathbf{a}_i\}$  to create a larger system. Since crystals are not infinite, they always have surfaces exposing some atoms to either vacuum or some other substance such as air.

The thermodynamics of crystal surfaces is very interesting due to its complicated nature. At zero temperature, it is possible to have a perfectly flat surface, but once atoms are allowed to move thermally, some of them may gain enough energy to escape its low energy state in the crystal lattice site, and become an *adatom* which will diffuse around on the surface. Several adatoms may meet and form *islands* with boundaries



**Figure 44:** A crystal surface to illustrate a) steps, kinks, adatoms and islands, and in b), a *vicinal* surface, or a stepped surface with a small angle  $\theta$  from one of the high symmetry directions.

called *steps* which may have *kinks* in them. These terms are illustrated in figure 44a.

If a surface has steps, kinks or adatoms, the combinatorics of the possible configurations lead to an important entropic term in the free energy  $F = U - TS$  that makes the physics of crystals both difficult and interesting.

A flat crystal surface is described as a *crystal plane* which is defined by at least three non-collinear<sup>3</sup> points in the underlying lattice. A plane with three lattice points actually contains an infinite number of points due to the required translational symmetry. Such planes are described by Miller indices  $h, k, \ell$  which, together with the lattice vectors  $\{\mathbf{a}_i\}$ , defines the normal vector to the plane

$$\mathbf{n}_{h,k,\ell} = h\mathbf{a}_1 + k\mathbf{a}_2 + \ell\mathbf{a}_3. \quad (4.1)$$

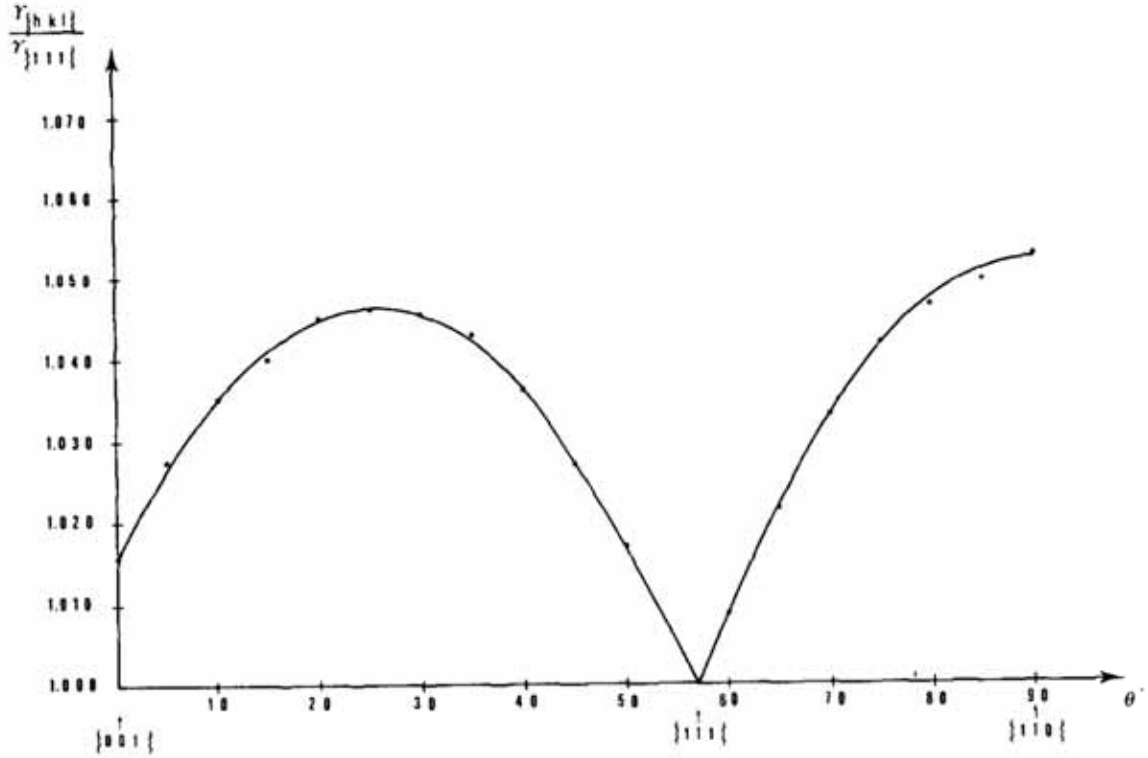
Depending on the underlying crystal, some of these planes are identical due to rotational symmetry. For instance, in a single cubic crystal, the (100), (010) and (001) planes are identical since the crystal can be rotated 90 degrees and still look exactly the same. Such collections of identical planes are referred to as  $\{100\}$  planes.

If we cut a crystal along one of these planes, we break some bonds, which requires some work  $W$  to be applied. Two new surfaces are then created, each with area  $A$ . We define the surface tension  $f$  as the amount of work per area

$$f = \frac{W}{2A}, \quad (4.2)$$

<sup>3</sup>Collinear points are points in a straight line.





**Figure 45:** Ratios of surface tension ratio  $\gamma_{hkl}/\gamma_{111}$  of indium in [36]. The  $\{111\}$  surface clearly has the lowest surface tension. Reprinted with permission from Elsevier.

where the factor 2 arises because we in fact have created two surfaces. Surface tension is an important quantity in the theory of crystal surfaces, and depends on the normal vector.  $f$  is in fact a free energy, so it has an entropic term, and is thus temperature dependent. Generally, we can write  $f$  as[88]

$$f(T, \mathbf{n}) = \epsilon(T, \mathbf{n}) - Ts(T, \mathbf{n}), \quad (4.3)$$

where  $\epsilon$  is the internal energy per area and  $s$  is the entropy per area.

In figure 45, the surface tension for indium is shown[36]. The surface tension is in the figure denoted as  $\gamma$ , but is identical to  $f$  in this chapter. As I will explain soon, the cusp (discontinuity in  $df/d\theta$ ) in the  $\{111\}$  direction corresponds to facets.

The full surface free energy of a crystal is found by integrating over the surface

$$\mathcal{F}_{\text{surf}} = \int_{\partial\Omega} f(T, \mathbf{n})dS, \quad (4.4)$$

and is one of the main quantities that determines the physics of a crystal since this is the quantity that is minimized to reach equilibrium. At zero temperature, the entropic term vanishes. In this case, only the internal energy defines the equilibrium shape.

A lot of the research on crystal surfaces has been on the growth on flat surfaces and *stepped surfaces* - surfaces with an angle  $\theta$  from a high symmetry direction such as (100) or (110) (see figure 44b). Surfaces grow when new atoms migrate to sites where they attach. Such sites are usually holes in the surface, or in kinks when steps are present. Migration to the surface may be controlled in different manners in vacuum, like MBE, or if it occurs in a fluid phase. In a fluid phase, the particles diffuse in

3 dimensions, whereas in vacuum diffusion is restricted to the 2 dimensional crystal surface. This limitation causes different growth phenomena in vacuum and in fluid phases.

In three groundbreaking papers by Burton, Cabrera and Frank[11–13], they present many ingredients of modern crystal growth science. These papers are focused on crystal growth on flat crystals and how temperature-dependent phenomena like step roughness, surface diffusion and nucleation affects how a crystal grows. I will in the following only treat growth in vacuum.

Theoretically, we assume that we have a nanoparticle in vacuum, where any shape transformation is caused only by reorganization of atoms within the nanoparticle. The nanoparticles consists of  $N$  atoms with some initial shape. Given that the system is allowed<sup>4</sup> to reach its equilibrium shape, what will it be? This question was answered by George Wulff in 1901 [110].

## 4.2 Wulff construction

The work in publication 3 is concerned with equilibrium shapes of volume-preserving nanocrystals. It was Gibbs[28], who in the 1880s realized that the equilibrium shape for a fixed volume substance will be the shape that minimizes the integrated free energy  $\mathcal{F}_{\text{surf}}$ . The problem can be formulated mathematically as minimizing  $\mathcal{F}_{\text{surf}}$  while keeping the volume constant. What follows is presented as in [72].

Assuming that the equilibrium shape is a polyhedra<sup>5</sup>, the surface free energy can be found by summing the contributions per facet

$$\mathcal{F}_{\text{surf}} = \sum_i f_i A_i, \quad (4.5)$$

where  $f_i \equiv f(\mathbf{n}_i)$  and  $A_i$  is the area of facet  $i$  (The temperature is assumed to be zero, so I have dropped it here). Assuming that the crystal can be constructed as a set of connected pyramids, one per facet, the volume of the crystal is  $V = 1/3 \sum_i A_i h_i$ , where  $h_i$  is the distance from the crystal center to the nearest point on the facet  $i$ . A constraint minimization using a Lagrange multiplier  $\lambda$  to enforce constant volume gives

$$\delta(\lambda V + \mathcal{F}_{\text{surf}}) = \sum_i \left( \frac{\lambda}{3} h_i + f_i \right) \delta A_i = 0. \quad (4.6)$$

This results in

$$\frac{f_i}{h_i} = \frac{\lambda}{3} \quad (4.7)$$

for all facets. Furthermore, this can be written as

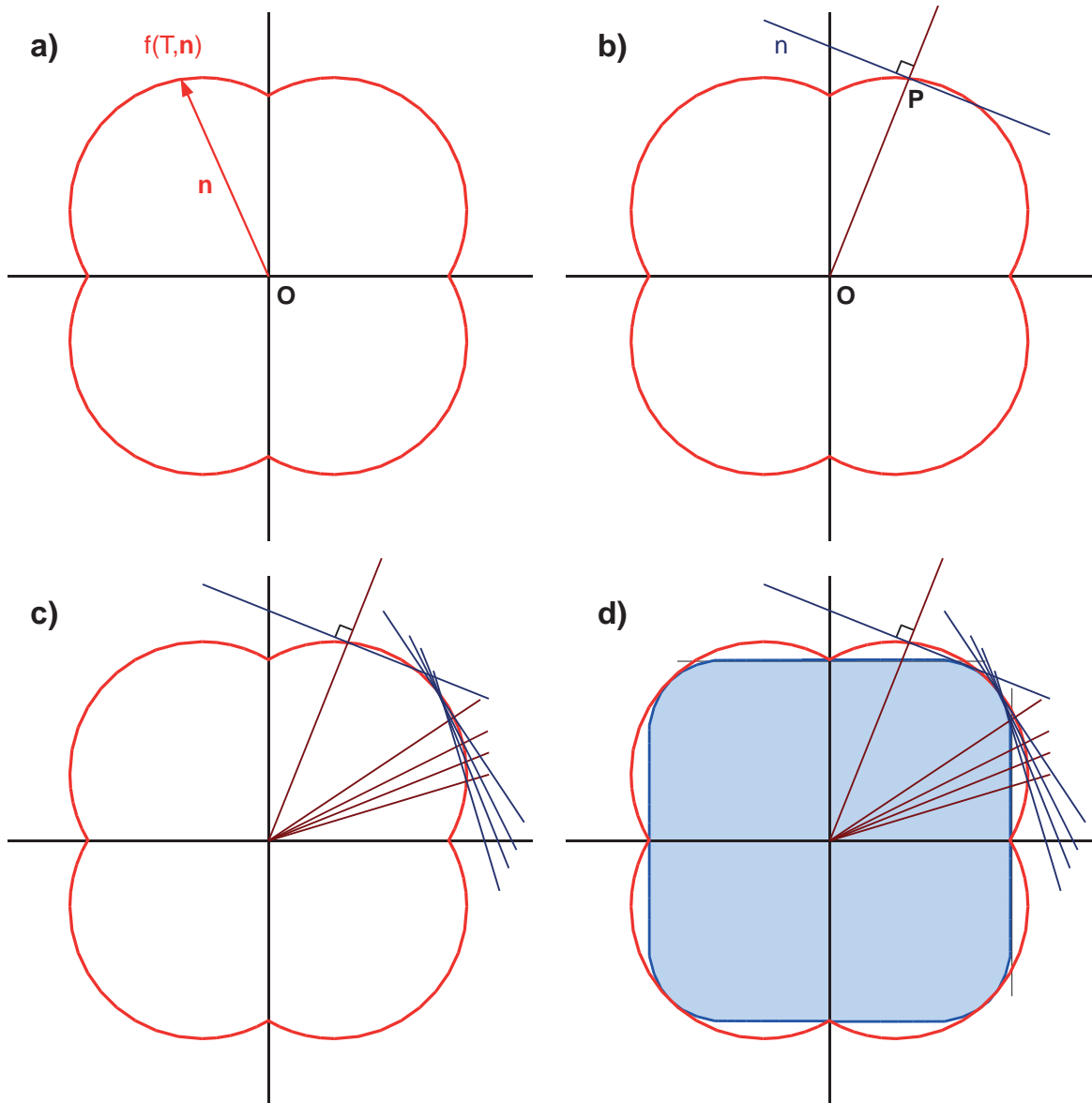
$$h(\mathbf{n}) = k f(\mathbf{n}), \quad (4.8)$$

for  $k = 3/\lambda$ , and means that *the distance from the crystal center to a facet is proportional to the surface tension*. This powerful result is the foundation for the Wulff

---

<sup>4</sup>Energy barriers may prevent this from happening, and the system may be stuck in a metastable state[58, 77].

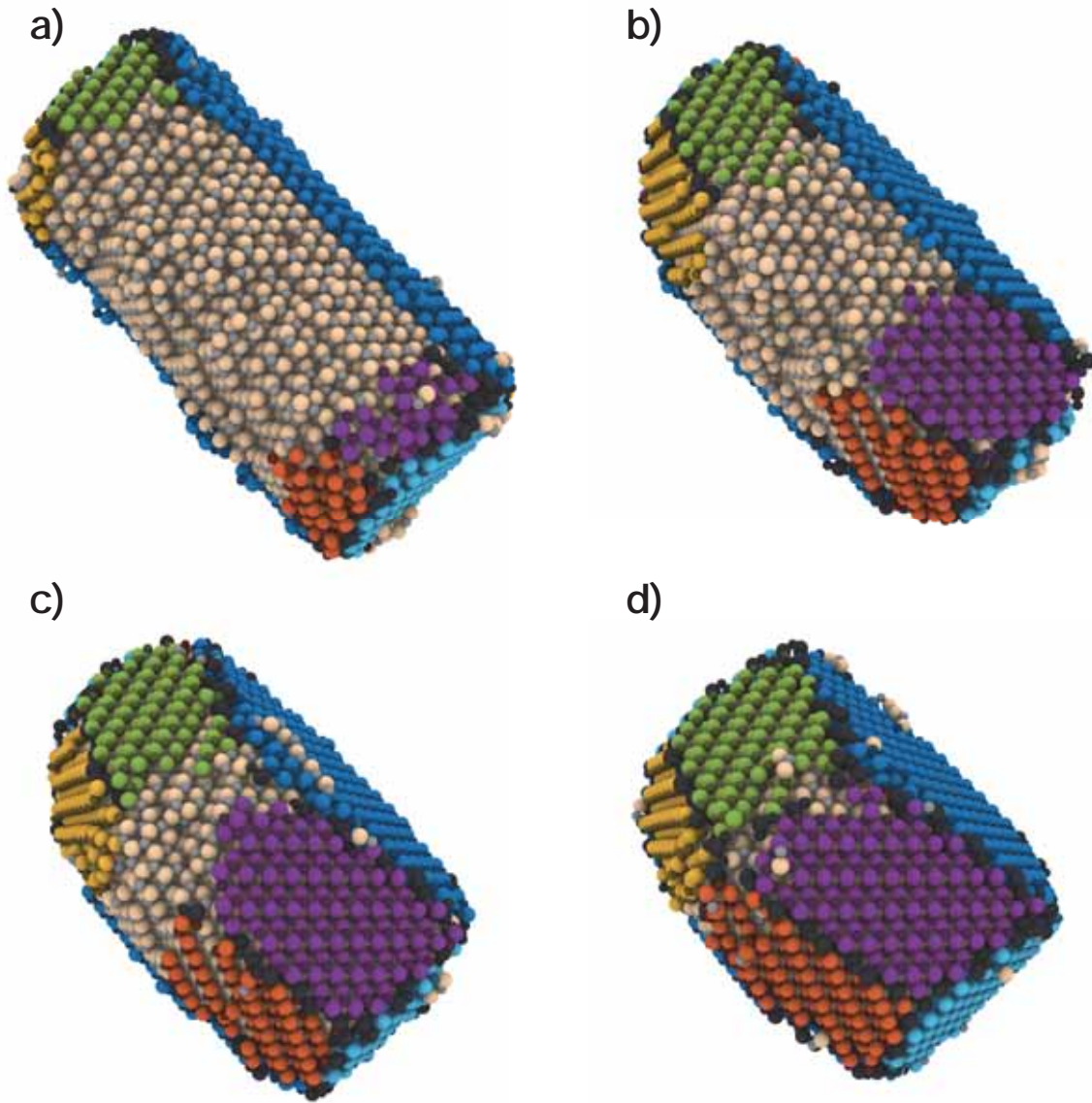
<sup>5</sup>This is not obvious at all, but turns out to be true at zero temperature.



**Figure 46:** The Wulff construction is a technique to create the equilibrium crystal shape from the surface tension  $f(T, \mathbf{n})$ . It works as follows: (a) Create a polar plot of the surface tension. This is the curve defined so that for each direction  $\mathbf{n}$ , the distance from the crystal center  $O$  to the intersection point on the curve is proportional to the surface tension. (b) For a point  $P$  on the curve, draw a perpendicular line to  $OP$  at the  $P$ . (c) Repeated this for all points on the curve. A few such normals are shown for different points  $P$  on the curve. (d) The crystal shape will be similar to the interior envelope of all these perpendicular lines. Notice how the cusps define facets, i.e. large flat areas on the crystal. This is a modified version of ©Michael Schmid / Wikimedia Commons / CC-BY-SA-2.5.

construction, which is used to create the equilibrium shape given the surface tension  $f(\mathbf{n})$ . The method is best described in two dimensions (this can be seen as a cross section of a three-dimensional crystal), but is easily extended to three dimensions. It works as follows.

*Draw the polar plot of the surface tension  $f(T, \mathbf{n})$ . This means, for each direction  $\mathbf{n}$ , put a point  $P$  in that direction with distance proportional to  $f(T, \mathbf{n})$  from the crystal*



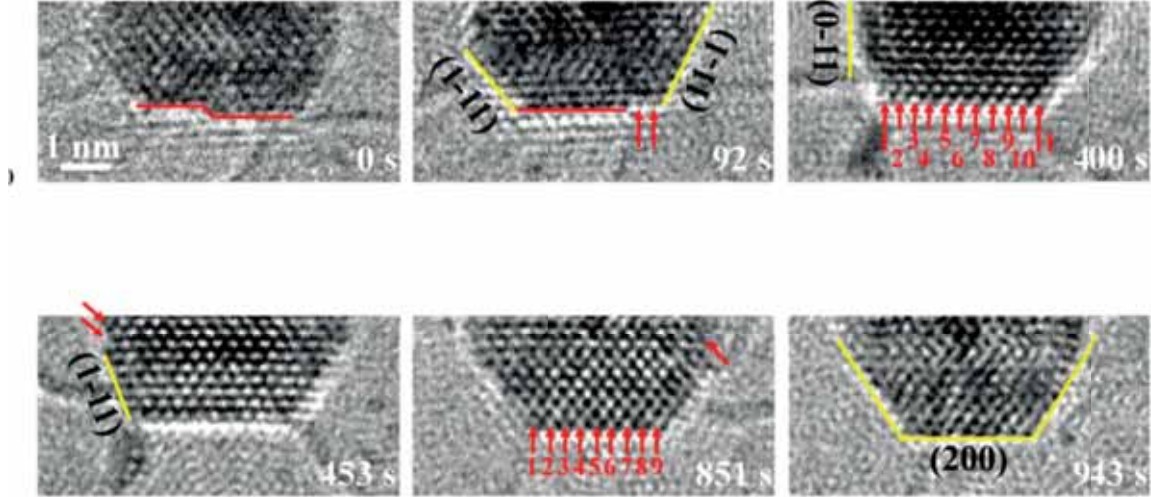
**Figure 47:** Facet formation in a silicon carbide nanoparticle. This simulation was performed with the Vashishta potential[104] at 2200 K. The strong colors have no meaning other than highlighting the different (110) facets. Edges between facets are colored black. a) shows the simulation after 1 ns where we see that the particle has started showing small facets. b) is taken at 40 ns, c) at 125 ns and d) at 1  $\mu$ s. Notice how the (110) facets grow laterally by adding step by step towards the metastable shape in d). This is metastable due to an energy barrier preventing the dark blue facets to grow normally which is required for the particle to reach its equilibrium shape. See publication 3 for more details.

*center  $O$ . Then, at each such point  $P$ , construct a perpendicular line to  $OP$ , which gives a family of lines<sup>6</sup>. The crystal shape will then be geometrical similar to the interior envelope of all these lines.*

In figure 46, I have shown how this works geometrically. By using eq (4.8), we can in principle measure the surface tension for all high index crystal planes and use these

---

<sup>6</sup>In three dimensions, these perpendicular lines are perpendicular planes.



**Figure 48:** State of the art images on faceting of  $\text{Mo}_2\text{C}$  nanoparticles at 1473 K taken from *Atomic-Scale Mechanism on Nucleation and Growth of  $\text{Mo}_2\text{C}$  Nanoparticles Revealed by in Situ Transmission Electron Microscopy*. These images are captured with TEM and shows how the nanocrystal grows by adding single atomic steps at the boundary. In the first (top left) image, we see a kink. Reprinted (adapted) with permission from [24]. Copyright (2018) American Chemical Society.

values to construct the equilibrium shape at low temperatures. As seen in figure 46, cusps in the polar plot correspond to facets. These cusps can for simple 2D Kossel crystals[22, 48, 49] be explained by that the energy to open a surface (creating a cut along an angle  $\theta$ ) is given as the Manhattan distance between the end points

$$f(\theta) \approx k(1 + |\theta| + \dots) \quad (4.9)$$

for small angles at zero temperature. Here, the constant  $k$  determines the strength of the interaction. The derivative of  $f$  at  $\theta = 0$  is discontinuous, and we have a cusp. At zero temperature, the equilibrium shape of a crystal is then a polyhedra with sharp edges and corners between the facets[22].

### 4.3 Facet formation

Atoms on high energy surfaces will diffuse more easily than on low energy surfaces. They may therefore jump around until they attach to a energetically favorable site. Typically, these sites are in steps on the edge of existing facets. Facets can either grow in the normal direction by adding a new layer, or laterally by adding new steps on the edge. When one facet grows normally, its neighboring facets will grow laterally since they then get a new step at the edge.

Figure 47 shows the facet formation in one of the nanocrystals in publication 3. Here we see that the area of (110) facets increase by atoms attaching at the edge of the facets. In simulations we can follow the trajectory of every atom. However, studying equilibrium shape crystals experimentally on the atomic scale turns out to be difficult. Observations are usually done with TEM, *Scanning Tunneling Microscopy* (STM) or *atomic force microscopy* (AFM)[6, 24, 51, 57, 72]. Such scanning methods have limited temporal resolution on atomic time scales, and require the growth to be very slow to



capture. In addition, small crystals may either float freely around in a vacuum chamber or, if grown on a already existing surface, have their shape altered. In figure 48, state of the art experimental images of atomic faceting mechanisms is shown from [24] using TEM. If the growth is slow enough, the formation of single steps with kinks can be captured, but the evolving facet dynamics have yet not been studied experimentally and still is considered to be largely unknown[51]. However, as we show in publication 3, the atoms can be followed in detail in simulations. We can then obtain an improved understand of how diffusion transforms a particle into its equilibrium shape. It does happen through surface diffusion.

## 4.4 Surface diffusion

Surface diffusion is the random motion of atoms jumping between adjacent surface lattice sites. The potential energy of an atom at a site is lower than between two sites, so to jump between them, the atom will have to overcome an energy barrier, a so-called activation energy[56]. The rate at which surface diffusion happens depends on different possible paths<sup>7</sup>[27, 78], but usually follows an Arrhenius law[72]

$$D \propto \exp(-E_d/kT), \quad (4.10)$$

for some activation energy  $E_d$ . We here see that at low temperatures, diffusion happens slowly. In figure 49, I show a simulation of a silicon carbide nanocrystal in its equilibrium shape with temperatures in the range 300 K to 2100 K over 400 ns. At 300 K, all atoms are locked in their lattice position and we see no diffusion. At 1800 K, the atoms at edges and corners<sup>8</sup>, are diffusing. All edges, corners and (111) facets diffuse at 2100 K. Although we have many diffusing atoms at high temperatures, a nanoparticle may be stuck in a metastable shape.

## 4.5 Energy barriers

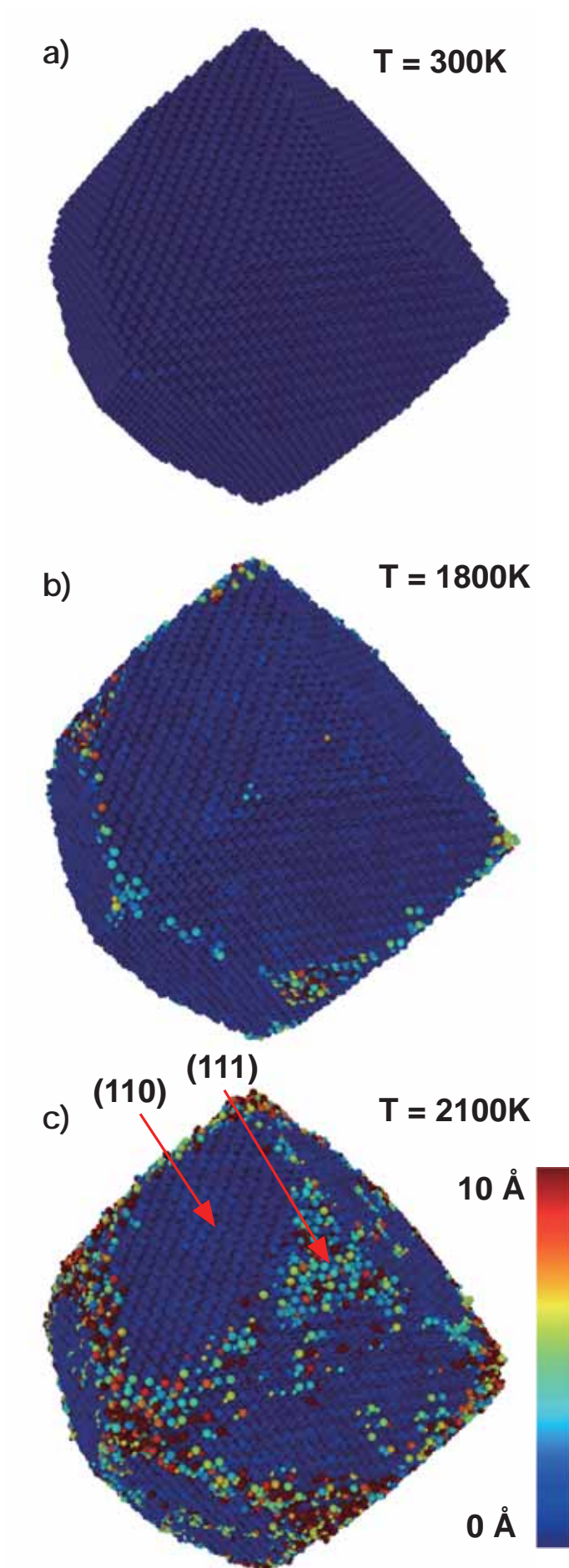
The nanoparticle in 47d is stuck in a metastable shape. In order for the particle to reach its equilibrium shape, two of the layers need to grow a layer normal to the facet. If the facet is sufficiently large, this growth may be prevented due to an energy barrier[58, 77]. This is explained by the fact that for a new layer to be grown, single adatoms have to randomly diffuse on top of a low energy surface where the adatom free energy is high.

Atoms located at one of these facets will quickly fall off again since edges and corners are energetically more favourable than the facet. This is evident from figure 49d, where the facets remain flat and have very few adatoms. In publication 3, we study this energy barrier in detail.

---

<sup>7</sup>Some directions may have higher energy barriers than others.

<sup>8</sup>The sharp corner in figure 49b is in the (100) direction.



**Figure 49:** Molecular dynamics simulation of an silicon carbide nanocrystal at its equilibrium shape. The particle consists of 103498 atoms, and was heated from 300 K to 2100 K over 0.4  $\mu\text{s}$ . Colors indicate the displacement of each atom over the past 2 ns (red means that the atom has moved at least 10  $\text{\AA}$ ). In a), we see that at 300 K, there is very little movement. In b), the temperature is 1800 K, with some edges and corners slowly diffusing. Note that some edges are still completely frozen. This indicates that this crystal is Wortis type B[109]. In c), the temperature is 2100 K. Here all edges, corners and the (111) facets are diffusing, but the (110) facets are still quite stable.





## Summary of publications

In this chapter, I summarize the four publications in this thesis.

**Publication 1: Effective workflow in molecular dynamics simulations using Atomify — a real-time LAMMPS visualizer**

Anders Hafreager, Svenn-Arne Dragly, Anders Malthe-Sørensen.

Submitted to *Modelling and Simulation in Materials Science and Engineering*.

Publication 1 is a proposed solution to the problem described in subsection 2.4.4: there is too much context switching overhead between different applications when running molecular dynamics simulations compared to how our minds work. Visual feedback is crucial during script development to verify the initial geometry and boundary conditions. In this paper, we introduce Atomify - a real-time LAMMPS visualizer that combines all the three stages described in section 2.4. The main focus of this application is to enable a user to quickly perform changes to the simulation script, and get immediate visual feedback on the results, either through visualization of the atoms, or by plotting a compute or variable real-time.

**My contribution:** I am the main developer in this project and I have written most of the paper.

**Publication 2: Game-Engine-Assisted Research platform for Scientific computing (GEARS) in Virtual Reality**

Brandon K. Horton, Rajiv K. Kalia, Erick Moen, Aiichiro Nakano, Ken-ichi Nomura, Michael Qian, Priya Vashishta and Anders Hafreager.

SoftwareX, **9**, 112-116, (2019).

[doi.org/10.1016/j.softx.2019.01.009](https://doi.org/10.1016/j.softx.2019.01.009)

While Atomify in publication 1 improves the workflow of script development, it does not yet support VR headsets or interactions with the simulation. Publication 2 is a collaboration with the *Materials Genome Innovation for Computational Software* (MAGICS) center. We here used existing game engines like Unity and Unreal Engine - both of which supports VR - to perform simulations in real-time, and visualize the contents using VR headsets like the HTC Vive and Oculus Rift combined with Leap Motion Controllers. High sensitivity sensors allow a user to walk around, crouch and rotate the head to get a full 3D view inside a simulation.

The Leap Motion Controller detects the position of hands and fingers. This is used to rotate, translate and zoom in on different features in a simulation.

Several different simulation codes were tested in different programming languages to explore the strengths of the different game engines. For instance, by using a similar data communication layer as in publication 1, we can run real-time simulations in LAMMPS using VR headsets. We implemented the same rendering technique as in Atomify to achieve high rendering performance.

**My contribution:** I planned and designed the LAMMPS application in Unreal Engine, how to efficiently render large amounts of atoms, and helped developing the code.

### **Publication 3: Direct atomic simulations of facet formation and equilibrium shapes of SiC nanoparticles**

Anders Hafreager, Henrik Andersen Sveinsson, Anders Malthe-Sørensen,  
Rajiv Kalia and Priya Vashishta  
In preparation.

Transformation of a nanoparticle from an initial shape to its equilibrium shape may require a huge mass transfer, and take a very long time depending on the surface diffusion. The rate at which surface diffusion happens follows an Arrhenius form[72], and is temperature dependent.

In this publication, we first used Atomify to simulate small silicon carbide nanoparticles in vacuum at different initial temperatures to see whether or not faceting happened on molecular dynamics time scales. At low temperatures, surface diffusion is too slow for faceting to happen, and at high temperatures, the system may melt or be above the roughening transition temperature at which facets disappear.

At 2200 K, we did see indications of facet formations from an initially spherical particle, and started further studies using supercomputers. This allowed us to study a phenomenon that has several unanswered questions today. Will all initial shapes eventually transform to the same shape? Where does most of the mass transport happen?

To answer these questions, several microsecond simulations were performed using the GPU implementations of the Vashishta potential discussed in section 3.2. Initially spherical, cubic and cylindrical nanocrystals with different sizes were simulated. We found that the equilibrium shape indeed is independent of the initial shape, and that the (110) and (111) crystal planes are the dominant facets. This makes the equilibrium shape a rhombic dodecahedron<sup>1</sup> truncated by (111) facets.

However, the initially cylindrical nanoparticle ended up in a metastable shape where two of the (110) facets did not grow normally<sup>2</sup> over a 1  $\mu$ s simulation<sup>3</sup>. It is a known phenomenon that if a facet is sufficiently large, an energy barrier may prevent the facet to grow normally[58, 77].

Our simulations indicated that we had found such a barrier. By heating the nanoparticle to 2361 K, both facets started to grow normally, and the particle ended up in the same equilibrium shape as the initially spherical and cubic nanoparticles. Most of the mass transport happens on edges and the corners between the facets.

---

<sup>1</sup>Twelve different (110) facet directions exist, which makes a dodecahedron the expected equilibrium shape if these facets have the lowest surface tension[107].

<sup>2</sup>Normal growth means adding or removing a full atomic layer.

<sup>3</sup>We also continued this simulation at 2200 K for a total of 3  $\mu$ s, but the facets showed no normal motion.

This is, as far as I know, the first direct atomic simulation of faceting.

**My contribution:** Henrik Sveinsson and I have contributed equally to this work and has worked together on all aspects of the project. We used my GPU implementation of the potential. This was necessary to run many very long simulations. Analysis and paper writing have also been done by both of us.

#### **Publication 4: Structural replication of nanoporous media using procedural noise**

Anders Hafreager, Nicolaas Groeneboom and Anders Malthe-Sørensen  
Submitted to *Physical Review E*.

This publication is concerned with the first stage of a molecular dynamics project: preparing an initial geometry. From [87], we have a method to generate physically realistic nanoporous materials in SiO<sub>2</sub> using molecular dynamics, but generating many statistical similar geometries may be computationally expensive. We show that by using procedural noise, a well-known technique from games and movies, we can reproduce the statistical properties of these geometries within seconds, once the optimal parameters have been found. The radial distribution function  $g(r)$  is used as a statistical measure to compare the original physically based geometry to our generated one. Procedural noise functions are typically sums of contributions on different length scales with several free parameters.

A fully automatic Markov chain Monte Carlo method was developed to find optimal parameters in a high-dimensional parameter space. We ignore the intramolecular part of  $g(r)$  during the parameter search since this does not capture the porous network. The model is capable of capturing most features of  $g(r)$ . Our best-fit model also reproduce porosity and surface area, but the surface gets bad chemical properties like the coordination number.

**My contribution:** I proposed the original idea. Then I developed, together with Nicolaas Groeneboom, the software implementation of the model. I have also written important parts of the paper.



# CHAPTER 6

## Outlook

How technology develops further is hard to predict. It seems reasonable that with the transistor size getting closer to the physical limit, new hardware will not only be inherently parallel, but also more specialized for certain tasks. Machine learning is a popular term these days, and has recently found its way into research in physics[15, 90].

One of the most popular machine learning softwares today is Tensorflow[1], developed and maintained by Google. It was released in 2016 - the same year as they revealed their *Tensor Processing Unit* (TPU), a hardware, specialized to perform machine learning very efficiently. This year, NVIDIA released their new GPU, NVIDIA V100, which also is specialized in machine learning[66].

Development of software should be done with this changing hardware landscape in mind, and I believe Sandia’s efforts with KOKKOS is a good way to go. Java’s saying, “*write once, run anywhere*”, also applies to the philosophy behind KOKKOS.

A single computer with eight high end GPU’s can perform better than 1000 modern supercomputer CPU cores<sup>1</sup>, and scales better in inhomogeneous systems with the default MPI domain splitting implementation found in most codes. I therefore expect real-time visualization tools like Atomify to be more common over the next years.

**Atomify:** Atomify (publication 1) needs to be improved. For instance, direct interaction with a running simulation is not yet possible. The lack of *virtual reality* (VR) support comes from the fact that it is built using Qt3D which does not support VR out of the box yet<sup>2</sup>. VR support in Qt3D will probably be supported sometime in the near future. While the VR environment GEARS in publication 2 is nice, it so far works more as a proof of concept than a production ready framework.

There also seems to be a change in how rendering is done. Traditional rendering works by representing geometrical objects as 3D primitives that are transformed by matrices on the GPU. These polygons are projected onto the two-dimensional screen where coloring of each pixel is calculated for each triangle covering a pixel. This rendering technique has been used for many years now, but lacks photorealistic quality.

GPU’s are now getting powerful enough to perform real-time physically based rendering, using raytracing. In raytracing, the paths of *light rays* are being traced, using physics to determine how the rays interact with surfaces. VMD and Ovito do support

---

<sup>1</sup>This number comes from the benchmark shown in figure 33 in section 3.2.

<sup>2</sup>There exists several VR projects using Qt already, and some progress on Qt3D support is done here: <https://github.com/dabulla/qt3d-vr/tree/virtual-reality>.

rendering images using physically based raytracing, but these methods are too slow to be run in real-time. Microsoft announced this year that they are adding support for real-time raytracing in their Direct3D API. I expect this new method to be entering scientific visualization softwares over the next few years.

While more realistic-looking rendering is nice, I don't think it adds value to the analysis process - at least not compared to what ambient occlusion<sup>3</sup> and VR adds. In publication 3, ambient occlusion was crucial during the analysis, since we more clearly could see the facets on the nanocrystals.

**Faceting of nanocrystals:** As far as I know, our work in publication 3 shows the first direct atomic simulations of faceting formation. Following the trajectories of each atom reveal a great deal about the process. Although we in this publication explain how faceting happens, there are many details we haven't discussed. For instance, the corners and edges in faceted silicon carbide nanoparticles remain sharp at temperatures up to several hundred kelvin. The crystal then seems to be of Wortis type B[109]. Such properties are important for our understanding of the roughening transition[5, 109].

**Procedural noise:** The use of procedural noise models like Perlin noise[69] and simplex noise[70] has shown to be promising in our pilot study in publication 4. Players of games like Minecraft<sup>4</sup>, enjoy interesting and rich worlds following simple rules from procedural noise. While our method reproduces the radial distribution function, surface area and porosity, we see clear problems on the atomic scale where the coordination number of surface atoms reveals that they are not chemically stable. The method only cares about geometry, not chemistry. Perhaps applying a passivation step may work, but the molecular topology is probably non-physical.

Using our model to produce geometries for continuum flow simulations would be interesting. I do not know whether or not the permeability would be reproduced, but this quantity is closely related to porosity and surface area[44].

---

<sup>3</sup>Ambient occlusion adds the effect of depth by making objects inside the system appear darker.

<sup>4</sup>Minecraft is one of the most popular games of all times. Visit [minecraft.net](http://minecraft.net) to find a new way to procrastinate.

# Bibliography

- [1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. Tensorflow: a system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.
- [2] B. Alder and T. Wainwright. Phase transition for a hard sphere system. *The Journal of chemical physics*, 27(5):1208–1209, 1957.
- [3] B. J. Alder and T. E. Wainwright. Studies in molecular dynamics. i. general method. *The Journal of Chemical Physics*, 31(2):459–466, 1959.
- [4] M. P. Allen and D. J. Tildesley. *Computer simulation of liquids*. Oxford university press, 2017.
- [5] M. Bienfait. Roughening and surface melting transitions: consequences on crystal growth. *Surface science*, 272(1-3):1–9, 1992.
- [6] G. Binnig and H. Rohrer. Scanning tunneling microscopy — from birth to adolescence. *reviews of modern physics*, 59(3):615, 1987.
- [7] D. W. Brenner, O. A. Shenderova, J. A. Harrison, S. J. Stuart, B. Ni, and S. B. Sinnott. A second-generation reactive empirical bond order (rebo) potential energy expression for hydrocarbons. *Journal of Physics: Condensed Matter*, 14(4):783, 2002.
- [8] W. M. Brown and M. Yamada. Implementing molecular dynamics on hybrid high performance computers - three-body potentials. *Computer Physics Communications*, 184(12):2785–2793, 2013.
- [9] W. M. Brown, P. Wang, S. J. Plimpton, and A. N. Tharrington. Implementing molecular dynamics on hybrid high performance computers - short range forces. *Computer Physics Communications*, 182(4):898–911, 2011.
- [10] W. M. Brown, A. Kohlmeyer, S. J. Plimpton, and A. N. Tharrington. Implementing molecular dynamics on hybrid high performance computers - particle-particle particle-mesh. *Computer Physics Communications*, 183(3):449–459, 2012.
- [11] W. Burton and N. Cabrera. Crystal growth and surface structure. part i. *Discussions of the Faraday Society*, 5:33–39, 1949.
- [12] W.-K. Burton, N. Cabrera, and F. Frank. The growth of crystals and the equilibrium structure of their surfaces. *Phil. Trans. R. Soc. Lond. A*, 243(866):299–358, 1951.



- [13] N. Cabrera and W. Burton. Crystal growth and surface structure. part ii. *Discussions of the Faraday Society*, 5:40–48, 1949.
- [14] T. Campbell, R. K. Kalia, A. Nakano, F. Shimojo, K. Tsuruta, P. Vashishta, and S. Ogata. Structural correlations and mechanical behavior in nanophase silica glasses. *Physical review letters*, 82(20):4018, 1999.
- [15] G. Carleo and M. Troyer. Solving the quantum many-body problem with artificial neural networks. *Science*, 355(6325):602–606, 2017.
- [16] V. Cnudde and M. N. Boone. High-resolution x-ray computed tomography in geosciences: A review of the current technology and applications. *Earth-Science Reviews*, 123:1–17, 2013.
- [17] M. O. Connor, H. M. Deeks, E. Dawn, O. Metatla, A. Roudaut, M. Sutton, B. R. Glowacki, R. Sage, P. Tew, M. Wonnacott, et al. Sampling molecular conformations and dynamics in a multi-user virtual reality framework. *arXiv preprint arXiv:1801.02884*, 2018.
- [18] M. E. Curtis, R. J. Ambrose, C. H. Sondergeld, et al. Structural characterization of gas shales on the micro-and nano-scales. In *Canadian unconventional resources and international petroleum conference*. Society of Petroleum Engineers, 2010.
- [19] H. Darabi, A. Eftehad, F. Javadpour, and K. Sepehrnoori. Gas flow in ultra-tight shale strata. *Journal of Fluid Mechanics*, 710:641–658, 2012.
- [20] P. Eastman, M. S. Friedrichs, J. D. Chodera, R. J. Radmer, C. M. Bruns, J. P. Ku, K. A. Beauchamp, T. J. Lane, L.-P. Wang, D. Shukla, et al. Openmm 4: a reusable, extensible, hardware independent library for high performance molecular simulation. *Journal of chemical theory and computation*, 9(1):461–469, 2012.
- [21] H. C. Edwards, C. R. Trott, and D. Sunderland. Kokkos: Enabling manycore performance portability through polymorphic memory access patterns. *Journal of Parallel and Distributed Computing*, 74(12):3202–3216, 2014.
- [22] T. L. Einstein. Equilibrium shape of crystals. In *Handbook of Crystal Growth (Second Edition)*, pages 215–264. Elsevier, 2015.
- [23] D. Faken and H. Jónsson. Systematic analysis of local atomic structure combined with 3d computer graphics. *Computational Materials Science*, 2(2):279–286, 1994.
- [24] L. Fei, S. M. Ng, W. Lu, M. Xu, L. Shu, W.-B. Zhang, Z. Yong, T. Sun, C. H. Lam, C. W. Leung, et al. Atomic-scale mechanism on nucleation and growth of mo<sub>2</sub>c nanoparticles revealed by in situ transmission electron microscopy. *Nano letters*, 16(12):7875–7881, 2016.
- [25] R. P. Feynman, R. B. Leighton, and M. Sands. *The Feynman lectures on physics, Vol. I: The new millennium edition: mainly mechanics, radiation, and heat*, volume 1. Basic books, 2011.
- [26] D. Frenkel and B. Smit. *Understanding molecular simulation: from algorithms to applications*, volume 1. Elsevier, 2001.

- [27] N. Freyer and H. Bonzel. Directional anisotropy of surface self-diffusion on pt (110). *Surface Science Letters*, 160(2):L501–L508, 1985.
- [28] J. W. Gibbs. Art. lii.—on the equilibrium of heterogeneous substances. *American Journal of Science and Arts (1820-1879)*, 16(96):441, 1878.
- [29] github. Gromacs contributors, 2018 (accessed June 30, 2018). URL <https://github.com/gromacs/gromacs/graphs/contributors>.
- [30] github. Lammps contributors, 2018 (accessed June 30, 2018). URL <https://github.com/lammps/lammps/graphs/contributors>.
- [31] github. Openmm contributors, 2018 (accessed June 30, 2018). URL <https://github.com/pandegroup/openmm/graphs/contributors>.
- [32] A. Hafreager. Molecular dynamics code written in lammps scripting language, 2018 (accessed June 30, 2018). URL <https://github.com/lammps/lammps/issues/862>.
- [33] G. Harp and B. Berne. Time-correlation functions, memory functions, and molecular dynamics. *Physical Review A*, 2(3):975, 1970.
- [34] D. R. Hartree. The wave mechanics of an atom with a non-coulomb central field. part i. theory and methods. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 24, pages 89–110. Cambridge University Press, 1928.
- [35] W. Haynes. *CRC Handbook of Chemistry and Physics, 92nd Edition*. 100 Key Points. CRC Press, 2011. URL <https://books.google.no/books?id=pYPRBQAAQBAJ>.
- [36] J. Heyraud and J. Métois. Surface free energy anisotropy measurement of indium. *Surface science*, 177(1):213–220, 1986.
- [37] R. W. Hockney and J. W. Eastwood. *Computer simulation using particles*. crc Press, 1988.
- [38] W. Humphrey, A. Dalke, and K. Schulten. Vmd: visual molecular dynamics. *Journal of molecular graphics*, 14(1):33–38, 1996.
- [39] F. Javadpour, D. Fisher, M. Unsworth, et al. Nanoscale gas flow in shale gas sediments. *Journal of Canadian Petroleum Technology*, 46(10), 2007.
- [40] F. Javadpour et al. Nanopores and apparent permeability of gas flow in mudrocks (shales and siltstone). *Journal of Canadian Petroleum Technology*, 48(08):16–21, 2009.
- [41] A. I. Jewett, Z. Zhuang, and J.-E. Shea. Moltemplate a coarse-grained model assembly tool. *Biophysical Journal*, 104(2):169a, 2013.
- [42] J.-W. Jiang, J.-S. Wang, and B. Li. Young’s modulus of graphene: a molecular dynamics study. *Physical Review B*, 80(11):113405, 2009.
- [43] J. E. Jones. On the determination of molecular fields. *Proc. R. Soc. Lond. A*, 106(738):463–477, 1924.

- [44] G. Karniadakis, A. Beskok, and N. Aluru. *Microflows and nanoflows: fundamentals and simulation*, volume 29. Springer Science & Business Media, 2006.
- [45] C. Kittel, P. McEuen, and P. McEuen. *Introduction to solid state physics*, volume 8. Wiley New York, 1996.
- [46] J. L. Klepeis, K. Lindorff-Larsen, R. O. Dror, and D. E. Shaw. Long-timescale molecular dynamics simulations of protein structure and function. *Current opinion in structural biology*, 19(2):120–127, 2009.
- [47] W. Kohn and L. J. Sham. Self-consistent equations including exchange and correlation effects. *Physical review*, 140(4A):A1133, 1965.
- [48] V.-W. Kossel. Zur energetik von oberflächenvorgängen. *Annalen der Physik*, 413(5):457–480, 1934.
- [49] W. Kossel. Zur theorie des kristallwachstums. *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse*, 1927:135–143, 1927.
- [50] A. Lagae, S. Lefebvre, R. Cook, T. Derose, G. Drettakis, D. S. Ebert, J. P. Lewis, K. Perlin, and M. Zwicker. State of the art in procedural noise functions. In *EG 2010-State of the Art Reports*. The Eurographics Association, 2010.
- [51] H.-G. Liao, D. Zhrebetsky, H. Xin, C. Czarnik, P. Ercius, H. Elmlund, M. Pan, L.-W. Wang, and H. Zheng. Facet development during platinum nanocube growth. *Science*, 345(6199):916–919, 2014.
- [52] R. M. Lynden-Bell, S. C. Morris, J. D. Barrow, J. L. Finney, and C. Harper. *Water and life: the unique properties of H<sub>2</sub>O*. CRC Press, 2010.
- [53] G. C. Maitland, M. Rigby, E. B. Smith, and W. Wakeham. *Intermolecular forces: their origin and determination*. Number 3. Oxford University Press, 1981.
- [54] G. J. Martyna, D. J. Tobias, and M. L. Klein. Constant pressure molecular dynamics algorithms. *The Journal of Chemical Physics*, 101(5):4177–4189, 1994.
- [55] N. Matthews, R. Easdon, A. Kitao, S. Hayward, and S. Laycock. High quality rendering of protein dynamics in space filling mode. *Journal of Molecular Graphics and Modelling*, 78:158–167, 2017.
- [56] Y. Mo, J. Kleiner, M. Webb, and M. Lagally. Activation energy for surface diffusion of si on si (001): A scanning-tunneling-microscopy study. *Physical review letters*, 66(15):1998, 1991.
- [57] Y.-W. Mo, R. Kariotis, B. Swartzentruber, M. Webb, and M. Lagally. Scanning tunneling microscopy study of diffusion, growth, and coarsening of si on si (001). *Journal of Vacuum Science & Technology A: Vacuum, Surfaces, and Films*, 8(1):201–206, 1990.
- [58] W. W. Mullins and G. S. Rohrer. Nucleation barrier for volume-conserving shape changes of faceted crystals. *Journal of the American Ceramic Society*, 83(1):214–16, 2000.

- [59] A. Nakano. Parallel multilevel preconditioned conjugate-gradient approach to variable-charge molecular dynamics. *Computer Physics Communications*, 104 (1-3):59–69, 1997.
- [60] A. Nakano, R. K. Kalia, and P. Vashishta. First sharp diffraction peak and intermediate-range order in amorphous silica: finite-size effects in molecular dynamics simulations. *Journal of non-crystalline solids*, 171(2):157–163, 1994.
- [61] A. Nakata, Y. Futamura, T. Sakurai, D. R. Bowler, and T. Miyazaki. Efficient calculation of electronic structure using o (n) density functional theory. *Journal of chemical theory and computation*, 13(9):4146–4153, 2017.
- [62] M. T. Nelson, W. Humphrey, A. Gursoy, A. Dalke, L. V. Kalé, R. D. Skeel, and K. Schulten. NAMD: a parallel, object-oriented molecular dynamics program. *The International Journal of Supercomputer Applications and High Performance Computing*, 10(4):251–268, 1996.
- [63] F. Ning, Y. Yu, S. Kjelstrup, T. J. Vlugt, and K. Glavatskiy. Mechanical properties of clathrate hydrates: status and perspectives. *Energy & environmental science*, 5(5):6779–6795, 2012.
- [64] K.-i. Nomura, R. K. Kalia, A. Nakano, and P. Vashishta. A scalable parallel algorithm for large-scale reactive force-field molecular dynamics simulations. *Computer Physics Communications*, 178(2):73–87, 2008.
- [65] S. Nosé. A unified formulation of the constant temperature molecular dynamics methods. *The Journal of chemical physics*, 81(1):511–519, 1984.
- [66] NVIDIA. Nvidia tesla v100 gpu architecture, 2018. URL <http://images.nvidia.com/content/volta-architecture/pdf/volta-architecture-whitepaper.pdf>. [Online; accessed 30-June-2018].
- [67] I. Omelyan, I. Mryglod, and R. Folk. Algorithm for molecular dynamics simulations of spin liquids. *Physical review letters*, 86(5):898, 2001.
- [68] J. F. Ouyang and R. Bettens. Modelling water: a lifetime enigma. *CHIMIA International Journal for Chemistry*, 69(3):104–111, 2015.
- [69] K. Perlin. An image synthesizer. *ACM Siggraph Computer Graphics*, 19(3):287–296, 1985.
- [70] K. Perlin. Improving noise. In *ACM Transactions on Graphics (TOG)*, volume 21, pages 681–682. ACM, 2002.
- [71] S. Phillpot, J. Lutsko, D. Wolf, and S. Yip. Molecular-dynamics study of lattice-defect-nucleated melting in silicon. *Physical Review B*, 40(5):2831, 1989.
- [72] A. Pimpinelli and J. Villain. *Physics of crystal growth*, volume 19. Cambridge university press Cambridge, 1998.
- [73] S. Plimpton. Fast parallel algorithms for short-range molecular dynamics. *Journal of computational physics*, 117(1):1–19, 1995.

- [74] S. Pronk, S. Páll, R. Schulz, P. Larsson, P. Bjelkmar, R. Apostolov, M. R. Shirts, J. C. Smith, P. M. Kasson, D. Van Der Spoel, et al. Gromacs 4.5: a high-throughput and highly parallel open source molecular simulation toolkit. *Bioinformatics*, 29(7):845–854, 2013.
- [75] A. Rahman. Correlations in the motion of atoms in liquid argon. *Physical Review*, 136(2A):A405, 1964.
- [76] A. K. Rappe and W. A. Goddard III. Charge equilibration for molecular dynamics simulations. *The Journal of Physical Chemistry*, 95(8):3358–3363, 1991.
- [77] G. S. Rohrer, C. L. Rohrer, and W. W. Mullins. Nucleation energy barriers for volume-conserving shape changes of crystals with nonequilibrium morphologies. *Journal of the American Ceramic Society*, 84(9):2099–2104, 2001.
- [78] C. Roland and G. H. Gilmer. Epitaxy on surfaces vicinal to si (001). i. diffusion of silicon adatoms over the terraces. *Physical Review B*, 46(20):13428, 1992.
- [79] S. J. Rose. Aspects of the thorium fuel cycle. 2017.
- [80] Sandia. Lammmps history, 2018. URL <http://lammmps.sandia.gov/history.html>.
- [81] Sandia. Lammmps benchmarks, 2018. URL <http://lammmps.sandia.gov/bench.html>.
- [82] T. Schneider and E. Stoll. Molecular dynamics study of a three-dimensional one-component model for distortive phase transitions. *Physical Review B*, 17(3):1302, 1978.
- [83] I. Shavitt and R. J. Bartlett. *Many-body methods in chemistry and physics: MBPT and coupled-cluster theory*. Cambridge university press, 2009.
- [84] D. E. Shaw, M. M. Deneroff, R. O. Dror, J. S. Kuskin, R. H. Larson, J. K. Salmon, C. Young, B. Batson, K. J. Bowers, J. C. Chao, et al. Anton, a special-purpose machine for molecular dynamics simulation. *Communications of the ACM*, 51(7):91–97, 2008.
- [85] D. E. Shaw, P. Maragakis, K. Lindorff-Larsen, S. Piana, R. O. Dror, M. P. Eastwood, J. A. Bank, J. M. Jumper, J. K. Salmon, Y. Shan, et al. Atomic-level characterization of the structural dynamics of proteins. *Science*, 330(6002):341–346, 2010.
- [86] A. Shekhar, K.-i. Nomura, R. K. Kalia, A. Nakano, and P. Vashishta. Nanobubble collapse on a silica surface in water: Billion-atom reactive molecular dynamics simulations. *Physical review letters*, 111(18):184503, 2013.
- [87] A. Shekhar, R. K. Kalia, A. Nakano, P. Vashishta, C. K. Alm, and A. Malthe-Sørensen. Universal stretched exponential relaxation in nanoconfined water. *Applied Physics Letters*, 105(16):161907, 2014.
- [88] A.-C. Shi and M. Wortis. Equilibrium shape of nacl crystals: A first-principles calculation. *Physical Review B*, 37(13):7793, 1988.

- [89] U. Sigma2. *Fram*, (accessed June 22, 2018). URL <https://www.sigma2.no/Fram>.
- [90] M. Spellings and S. C. Glotzer. Machine learning for crystal identification and discovery. *AIChE Journal*, 64(6):2198–2206, 2018.
- [91] F. H. Stillinger and T. A. Weber. Computer simulation of local order in condensed phases of silicon. *Physical review B*, 31(8):5262, 1985.
- [92] A. Stukowski. Visualization and analysis of atomistic simulation data with ovito—the open visualization tool. *Modelling and Simulation in Materials Science and Engineering*, 18(1):015012, 2009.
- [93] A. Stukowski. Structure identification methods for atomistic simulations of crystalline materials. *Modelling and Simulation in Materials Science and Engineering*, 20(4):045021, 2012.
- [94] A. Stukowski. *Ovito features*, (accessed June 26, 2018). URL <http://ovito.org/index.php/about/features>.
- [95] M. Sun and C. Ebner. Molecular-dynamics simulation of compressible fluid flow in two-dimensional channels. *Physical Review A*, 46(8):4813, 1992.
- [96] W. C. Swope, H. C. Andersen, P. H. Berens, and K. R. Wilson. A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters. *The Journal of Chemical Physics*, 76(1):637–649, 1982.
- [97] J. Tersoff. New empirical approach for the structure and energy of covalent systems. *Physical Review B*, 37(12):6991, 1988.
- [98] J. Tersoff. Modeling solid-state chemistry: Interatomic potentials for multicomponent systems. *Physical Review B*, 39(8):5566, 1989.
- [99] A. P. Thompson, L. P. Swiler, C. R. Trott, S. M. Foiles, and G. J. Tucker. Spectral neighbor analysis method for automated generation of quantum-accurate interatomic potentials. *Journal of Computational Physics*, 285:316–330, 2015.
- [100] J. Tranchida, S. Plimpton, P. Thibaudau, and A. Thompson. Massively parallel symplectic algorithm for coupled magnetic spin dynamics and molecular dynamics. *Journal of Computational Physics*, 2018.
- [101] A. C. Van Duin, S. Dasgupta, F. Lorant, and W. A. Goddard. Reaxff: a reactive force field for hydrocarbons. *The Journal of Physical Chemistry A*, 105(41):9396–9409, 2001.
- [102] P. Vashishta, R. K. Kalia, G. A. Antonio, and I. Ebbsjö. Atomic correlations and intermediate-range order in molten and amorphous gese 2. *Physical review letters*, 62(14):1651, 1989.
- [103] P. Vashishta, R. K. Kalia, J. P. Rino, and I. Ebbsjö. Interaction potential for sio 2: a molecular-dynamics study of structural correlations. *Physical Review B*, 41(17):12197, 1990.



- [104] P. Vashishta, R. K. Kalia, A. Nakano, and J. P. Rino. Interaction potential for silicon carbide: a molecular dynamics study of elastic constants and vibrational density of states for crystalline and amorphous silicon carbide. *Journal of applied physics*, 101(10):103515, 2007.
- [105] P. Vashishta, R. K. Kalia, A. Nakano, and J. P. Rino. Interaction potentials for alumina and molecular dynamics simulations of amorphous and liquid alumina. *Journal of Applied Physics*, 103(8):083504, 2008.
- [106] P. Vashishta, R. K. Kalia, A. Nakano, J. P. Rino, C. for Advanced Computing, and Simulations. Interaction potential for aluminum nitride: A molecular dynamics study of mechanical and thermal properties of crystalline and amorphous aluminum nitride. *Journal of Applied Physics*, 109(3):033514, 2011.
- [107] H.-R. Wenk and A. Bulakh. *Minerals: Their constitution and origin*. Cambridge University Press, 2016.
- [108] Wikipedia contributors. List of nvidia graphics processing units — Wikipedia, the free encyclopedia, 2018. URL [https://en.wikipedia.org/wiki/List\\_of\\_Nvidia\\_graphics\\_processing\\_units](https://en.wikipedia.org/wiki/List_of_Nvidia_graphics_processing_units). [Online; accessed 26-June-2018].
- [109] M. Wortis. Equilibrium crystal shapes and interfacial phase transitions. In *Chemistry and Physics of Solid Surfaces VII*, pages 367–405. Springer, 1988.
- [110] G. Wulff. Zur frage der geschwindigkeit des wachstums und der auflösung der kristalle. *Zeitschrift für Krystallographie und Mineralogie*, 34:449–530, 1901.





# Game-Engine-Assisted Research platform for Scientific computing (GEARS) in Virtual Reality

Brandon K. Horton, Rajiv K. Kalia, Erick Moen, Aiichiro Nakano, Ken-ichi Nomura,  
Michael Qian, Priya Vashishta and Anders Hafreager  
*SoftwareX*, **9**, 112-116, (2019).  
[doi.org/10.1016/j.softx.2019.01.009](https://doi.org/10.1016/j.softx.2019.01.009)





Original software publication

## Game-Engine-Assisted Research platform for Scientific computing (GEARS) in Virtual Reality



Brandon K. Horton<sup>a</sup>, Rajiv K. Kalia<sup>a,b,c,d</sup>, Erick Moen<sup>b,e</sup>, Aiichiro Nakano<sup>a,b,c,d,f</sup>, Ken-ichi Nomura<sup>a,d,\*</sup>, Michael Qian<sup>a</sup>, Priya Vashishta<sup>a,b,c,d</sup>, Anders Hafreager<sup>g</sup>

<sup>a</sup> Collaboratory for Advanced Computing and Simulations, University of Southern California, Los Angeles, CA 90089-0241, USA

<sup>b</sup> Department of Computer Science, University of Southern California, Los Angeles, CA 90089-0241, USA

<sup>c</sup> Department of Physics & Astronomy, University of Southern California, Los Angeles, CA 90089-0241, USA

<sup>d</sup> Department of Materials Science and Chemical Engineering, University of Southern California, Los Angeles, CA 90089-0241, USA

<sup>e</sup> Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089-0241, USA

<sup>f</sup> Department of Biological Science, University of Southern California, Los Angeles, CA 90089-0241, USA

<sup>g</sup> Department of Physics, University of Oslo, 0316 Oslo, Norway

### ARTICLE INFO

#### Article history:

Received 26 April 2018

Received in revised form 21 November 2018

Accepted 11 January 2019

#### Keywords:

Virtual reality

Real-time simulation visualization

Molecular dynamics

LAMMPS

### ABSTRACT

The Game-Engine-Assisted Research platform for Scientific computing (GEARS) is a visualization framework developed at the Materials Genome Innovation for Computational Software (MAGICS) center to perform simulations and on-the-fly data exploration in virtual reality (VR) environments. This hardware-agnostic framework accommodates multiple programming languages and game engines in addition to supporting integration with a widely-used materials simulation engine called LAMMPS. GEARS also features a novel data exploration tool called virtual confocal microscopy, which endows scientific visualization with enhanced functionality.

© 2019 Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

### Code metadata

Current code version  
 Permanent link to code/repository used of this code version  
 Legal Code License  
 Code versioning system used  
 Software code languages, tools, and services used  
 Compilation requirements, operating environments & dependencies  
 If available Link to developer documentation/manual  
 Support email for questions

v1.0  
[https://github.com/ElsevierSoftwareX/SOFTX\\_2018\\_42](https://github.com/ElsevierSoftwareX/SOFTX_2018_42)  
 GPL 3.0  
 git  
 C++, C#, Python  
 Windows 10, Visual Studio, Unity, Unreal Engine, Python,  
 LAMMPS  
<https://github.com/USCCACS/GEARS/blob/master/README.md>  
<https://magics.usc.edu/gears/>  
[cacs@usc.edu](mailto:cacs@usc.edu)

### 1. Motivation and significance

Data visualization plays a key role in scientific discovery. Though quantitative analysis is indispensable, researchers are often forced to apply summary metrics blindly. Unfortunately, some of these statistics are limited in their ability to describe the system under test and can be misleading, as illustrated by Anscombe's Quartet [2].

\* Corresponding author at: Collaboratory for Advanced Computing and Simulations, University of Southern California, Los Angeles, CA 90089-0241, USA.

E-mail address: [knomura@usc.edu](mailto:knomura@usc.edu) (K.-i. Nomura).

Visualization software such as VisIt [3], ParaView [4], Visual Molecular Dynamics (VMD) [5], and OVITO [6] provide a straightforward interface to create three-dimensional (3D) images and observe patterns in the output of simulations. Though often used among experienced researchers, it remains challenging to extract information from datasets that consist of complex geometries or contain a large number of entities because of the intrinsic limitation of two-dimensional (2D) representations of 3D objects for traditional displays. Immersive 3D visualization technology like ImmersaDesk, Tile Wall, and CAVE2 [7] provide unique spaces for collaboration and scientific communications [8]. These solutions, however, require specialized knowledge of the respective systems

<https://doi.org/10.1016/j.softx.2019.01.009>

2352-7110/© 2019 Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).



**Fig. 1.** (Left) A user wearing a head-mounted display. (Right) A user manipulating simulation data of a tellurene material [1]. The skeletal hands are processed by the LeapMotion library in the Unity 5 Game Engine. Using finger pinches and gesture controls, the user can rotate, translate, and scale the structure intuitively in immersive VR. The surrounding graphical user interface is the Unity editor.

and clear expectations of the simulation's outcome. Furthermore, the high costs of these systems make them prohibitive for most researchers.

An affordable alternative is an easily-accessible, agnostic software platform designed for the increasingly available head-mounted displays (HMDs) developed for consumer virtual reality (VR). VR is a powerful visualization tool that has grown steadily in popularity over the past two decades, fueled by the success of HMDs like Oculus Rift [9] and HTC Vive [10]. Despite its widespread adoption for video games and media applications, as well as its potential as the most immersive and intuitive method for viewing data, VR has been underutilized by the scientific community. To transform VR into a common scientific tool, we have developed a software framework, called Game-Engine-Assisted Research platform for Scientific computing (GEARS) that facilitates the adoption of VR technologies and allows researchers to take advantage of the unique analytical advantages that the medium offers (Fig. 1).

## 2. Software description

### 2.1. Interactive data visualization

GEARS makes use of commodity game engines, like Unity [13] and Unreal Engine [14], to simplify access to VR headsets. The most straightforward application of VR to data visualization is interactive viewing of pre-computed results for data exploration. To realize this feature, GEARS employs the iBET workflow [15], in which external 3D modelers, such as VMD and Blender [16], are used to create a 3D object that can be added to the scene along with the appropriate script (for example, LeapRTS.cs for the Leap Motion controller [17]) (see Fig. 2).

This aspect of GEARS allows users a quick, straightforward outlet for immediate visualization of snapshots of data from materials simulations or molecular structures. Though the current interactive visualization mode supports single frame data, we plan to expand this feature by creating multiple scenes containing different simulation time steps for more dynamical data exploration. Fig. 3 showcases several examples of GEARS interactive visualizations used in computational researches.

### 2.2. Real-time simulation visualization

GEARS also takes advantage of the programming capabilities provided by game engines, such as C# and JavaScript supported by Unity and C++ for Unreal Engine, to facilitate reuse of existing simulation codes. This mode of GEARS is suitable to explore simulation results in real-time, rendered entirely within the game engine.

To realize this real-time rendering, it is critically important to design an efficient data-bridging method between the game engine and user-developed simulation program. GEARS employs two approaches, called *Run-when-Ready* and *Render-when-Ready*, depending on the size of the data and the complexity of the simulation engine. *Run-when-Ready* calls the simulation engine upon a

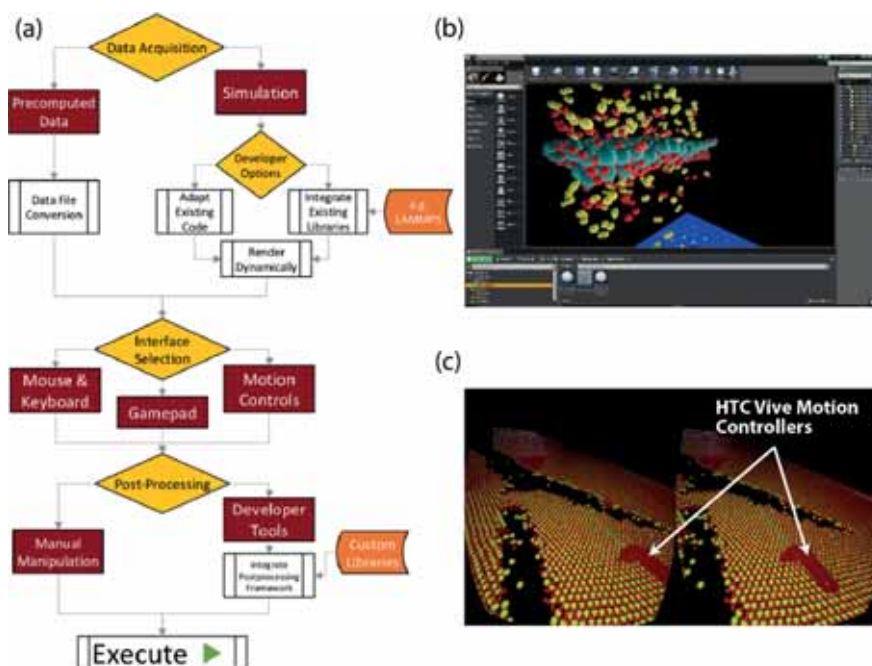
frame update to advance the state of the simulation (for example particle positions) of the frame by one timestep. *Render-when-Ready* makes use of the multi-threading optimization technique to offload the simulation engine computation onto a new thread while the main thread is only responsible for handling the render state of the game engine. In this approach, when the simulation engine finishes one timestep computation, the main thread either updates the frame state or stores the simulation states for rendering later while the worker thread continues to produce new states as background. Using this technique as well as other optimizations, like impostors and GPU instancing [18], we have demonstrated real-time simulation sizes reaching up to 500,000 particles. To minimize the amount of coding necessary for GEARS users, we provide two demos – Lennard-Jones molecular dynamics (MD) [19] and electron transfer simulation by kinetic Monte Carlo [20] – implementing the *Run-when-Ready* and *Render-when-Ready* approaches in the GEARS GitHub repository. See Demo3 of UnityGEARS on the GEARS repository.

In an effort to make our immersive scientific computing suite accessible to a broad research community, we have also interfaced GEARS with one of the most widely used MD simulation engine called LAMMPS (Large-scale Atomic/Molecular Massively Parallel Simulator) [21]. LAMMPS was developed at Sandia National Laboratory and supports a variety of interatomic potentials, statistical ensembles, and flexible simulation setups. The LAMMPS interface enables users to visualize their MD simulations in VR environment without any coding. A “How-to” for setting up the LAMMPS integration environment and an example demo are documented in detail in the Lammmps Compilation section of UnrealGEARS on the GEARS repository.

### 2.3. Virtual confocal microscopy

To fully realize the promise of immersive scientific computing, we have developed a new tool to enhance GEARS' visualization capability, called virtual confocal microscopy. Confocal microscopy has become an indispensable biomedical tool, and it has proven useful in many other scientific fields [22–25]. It allows for increased resolution, contrast, and optical sectioning while minimizing photodamage and bleaching concerns. The technique uses point illumination and a pinhole to scan the sample and reject out-of-focus light. As a result, the image only contains information very close to the focal plane, offering greater resolution [24]. An analogous approach in VR could enable more methodical investigations of simulated complex structures.

Virtual confocal microscopy utilizes Unity's surface shader capabilities to control how each vertex on the structure is rendered in order to highlight certain areas or planes of the material in the simulation. Our solution seeks to generate a viewing plane that sits in front of the user's head, follows their head movement, and always maintains a set distance from the user. This distance, as well as the thickness of the highlighted viewing plane can be specified



**Fig. 2.** (a) Three primary steps (Data Acquisition, Interface Selection, and Post-Processing) of the GEARS visualization workflow. In the Data Acquisition step, a researcher supplies data to the engine by either converting pre-computed results to a particular format (e.g. .obj, .fbx) or computing data on the fly via simulation. In the Interface Selection step, a GEARS user selects the method or interface through which they desire to interact with their pre-computed data or real-time simulation. In the Post-Processing step, a researcher selects a post-processing technique, which can either utilize the control scheme they chose in the Interface Selection step or make use of custom code. (b) An example of visualizing a chemical vapor deposition (CVD) simulation [11,12] through GEARS' LAMMPS integration. (c) Simulating and exploring data related to a molybdenum disulfide (MoS<sub>2</sub>) fracture via the LAMMPS integration. Using the controllers indicated by white lines, researchers can freely navigate through the structure, pause and resume the animation, and even highlight regions of interest with the provided laser attachment (shown in green).

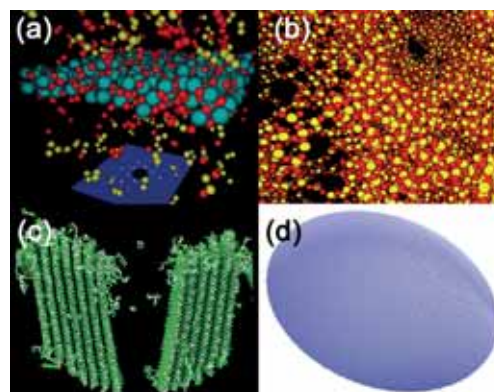
and changed dynamically by the user. The rest of the simulated system will stay mostly transparent (with the opacity also dictated by user input), except for the vertices that intersect with this plane. In this way, the user can scan through complicated geometries using just their head movements - a unique experimental advantage only possible in VR.

#### 2.4. Control schemes

A carefully designed control scheme is essential to facilitate interaction between the user and various data representations within the immersive simulation. These interactions range from object manipulation and spatial translations to temporal scaling. It is critical that these interactions mimic natural motion as much as possible to limit user discomfort, especially as it relates to user motion within the environment [26–28]. GEARS supports input from a variety of sources that includes the Leap Motion controller, Oculus Touch, and HTC Vive motion for immersive environments, as well as keyboard and mouse for small and precise adjustments to aspects of the simulation.

### 3. Illustrative examples

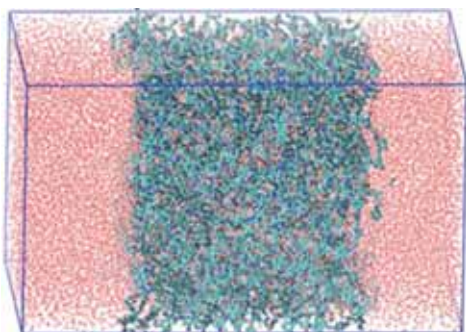
Recently, researchers performed MD simulations of desalination membranes to understand and characterize how local, atomic structures contribute to macroscopic behavior [31]. The relevant dimensions of the polyamide membranes of interest range from a tenth of a nanometer to 100 nm, making it an excellent candidate for computational studies to extract optimization information not easily accessible experimentally. They have found that water molecules permeated the membrane through benzene rings and



**Fig. 3.** Examples of interactive visualization: (a) Realtime rendering and simulation of MoS<sub>2</sub> synthesis. (b) Walkthrough of three-dimensional nanometric cage structure in a porous amorphous silica. (c) Exfoliation process due to dynamic shear-flow on a 2D material surface [29]. (d) Strain-induced structural transformation in an atomic telluride layer [30].

that the degree to which these monomers were cross-linked governed the speed of permeation along various paths [31]. However, the construction of these membranes is very complex, making a quick identification of preferential paths difficult with traditional 2D visualizations. The ability to section such a complicated super-structure in VR environment with simple head motion helps researchers identify prevalent substructures of interest in these simulated materials, and will greatly accelerate the research production cycle (see Fig. 4).





**Fig. 4.** Crosslinked aromatic polyamide membrane immersed in water. Water molecules (red) diffuse through nanometric channels in the membrane (green) at the middle of the simulation box.

Source: Adapted with permission from [27].  
© 2016 American Chemical Society.

#### 4. Impact

Physical simulations, like MD, inherently exist in 3D space, yet modern, desktop viewing platforms tend to only support 2D visualizations. GEARS, as an open-source, commodity platform, makes exploratory visualization and immersive data analysis more accessible for researchers. Using our workflow, outlined above, and our provided vertex shader, one can apply 3D virtual confocal microscopy to any discrete data set. Alternatively, should a researcher need to create their own customized shader (depending on their investigative needs), they can make use of the sample environments we have provided to productively bring their own data to an interactive, immersive environment. Consequently, this manuscript sets a foundational codebase for scientists to make use of modern graphics shaders for experimental data analysis.

Even if shader technology or virtual confocal microscopy is not a primary goal, researchers can still make use of the immersive playback capabilities in GEARS. For pre-computed simulations, this generally means pausing animations then exploring and analyzing sequestered regions. However, not only have we provided an immersive visualization engine for researchers, but also a simulation platform that encapsulates much of the complicated logic necessary for real-time data analysis. Researchers can choose to either port their existing data (e.g. LAMMPS DUMP files) or run entirely new simulations on-the-fly. With our framework, LAMMPS scripts can be written with VR visualization accounted for by design. Researchers can even pause a simulation mid-run, giving them opportunities to make adjustments to their simulation before supplying new commands to the linked LAMMPS library through our code interface. Afterwards, they can continue running the simulation and note the effects of their adjustments to the system. This option for immersive simulation steering is just one example of how our software also acts as a platform for VR-oriented extensions in LAMMPS. By integrating LAMMPS with Unreal, much like Atomify [32] integrated the MD simulator with mobile operating systems, we have improved the framework's extensibility as a research tool. Additionally, build upon our sample environments, researchers can more productively take advantage of the engine's rich features — for instance, its native server framework for multi-user LAMMPS VR collaboration, built-in blueprint system for interactive UI, and plugin controls for immersive navigation of a 3D simulation.

GEARS in Unity and Unreal act as an extensible hybrid simulation and visualization engine. It has already seen use in classroom and research settings, acting as a new medium for presenting data. This medium allows researchers to take advantage of humanity's

innate spatial awareness to improve its user's productivity as well as a new avenue for scientific communication. It has facilitated the exchange of simulation data to both experts and non-experts, most notably at U.S. Department of Energy (DOE) Materials Genome Innovation for Computational Software (MAGICS) workshops and at a DOE conference in Washington DC.

#### 5. Conclusions

We have developed a hardware-agnostic visualization framework called GEARS that takes advantage of the unique possibilities and power associated with the use of VR environments and rapidly evolving game engine technology to explore scientific datasets. VR has become a powerful new method of engaging with users in a more tactile, visceral way. The resurgence in hardware to support VR has resulted in a rich software environment that allows developers access to these tools with minimal investment. While VR technologies have a myriad of applications, they could be particularly useful for scientific data visualization and exploration, collaboration with domain experts, as well as communications with non-scientific audiences. With GEARS, we provide various approaches and modalities for researchers to interactively explore their simulation data, from visualization of pre-computed datasets in VR to the integration of external simulation engines such as LAMMPS. GEARS even allows for researchers to take advantage of immersive data visualization and simulation without requiring additional coding. Driven by the multimedia and gaming industries, it is likely that VR platforms will continue to enhance user functionality and interactivity at lower costs, making them even more approachable to researchers. Therefore, the novel data exploration and collaboration capabilities GEARS offers will be broadly beneficial for scientific community.

#### Acknowledgments

This work was supported as part of the Computational Materials Sciences Program funded by the U.S. Department of Energy, Office of Science, Basic Energy Sciences, under Award Number DE-SC00014607, for our MAGICS (Materials Genome Innovation for Computational Software) center, <https://magics.usc.edu>.

#### References

- [1] Apte A, et al. Structural phase transformation in strained monolayer MoWSe<sub>2</sub> alloy. *ACS Nano* 2018;12(4):3468–76.
- [2] Anscombe FJ. Graphs in statistical analysis. *Amer Statist* 1973;27(1):17–21.
- [3] Childs H, et al. VisIt: An end-user tool for visualizing and analyzing very large data. 2012.
- [4] Ahrens J, Geveci B, Law C, Hansen CD, Johnson CR. An end-user tool for large-data visualization. In: *The Visualization Handbook*. 2005, p. 717.
- [5] Humphrey W, Dalke A, Schulten K. VMD: Visual molecular dynamics. *J Mol Graph* 1996;14(1):33–8.
- [6] Stukowski A. Visualization and analysis of atomistic simulation data with OVITO—the open visualization tool. *Modelling Simulation Mater Sci Eng* 2010;18(1).
- [7] Febretti A, et al. CAVE2: A hybrid reality environment for immersive simulation and information analysis. In: *Engineering Reality of Virtual Reality 2013*, vol. 8649. 2013.
- [8] Manjrekar S, Sandilya S, Bhosale D, Kanchi S, Pitkar A, Gondhalekar M. CAVE: An Emerging Immersive Technology – A Review. *IEEE*; 2014, p. 131–6.
- [9] Oculus Rift. <https://www.oculus.com/rift/>.
- [10] HTC Vive. <https://www.vive.com/us/>.
- [11] Hong S, et al. Computational synthesis of MoS<sub>2</sub> layers by reactive molecular dynamics simulations: initial sulfidation of MoO<sub>3</sub> surfaces. *Nano Lett* 2017;17(8):4866–72.
- [12] Hong S, et al. Chemical vapor deposition synthesis of MoS<sub>2</sub> layers from the direct sulfidation of MoO<sub>3</sub> surfaces using reactive molecular dynamics simulations. *J Phys Chem C* 2018.
- [13] Unity. <https://unity3d.com>.
- [14] Unreal Engine 4 Documentation. <https://docs.unrealengine.com/latest/INT/index.html>.



- [15] Nakano CM, et al. iBET: Immersive visualization of biological electron-transfer dynamics. *J Mol Graph Modelling* 2016;65:94–9.
- [16] Blender. <https://www.blender.org>.
- [17] Leap Motion Developers. <https://developer.leapmotion.com>.
- [18] Tarini M, Cignoni P, Montani C. Ambient occlusion and edge cueing to enhance real time molecular visualization. *IEEE Trans Vis Comput Graphics* 2006;12(5):1237–44, (in English).
- [19] Frenkel D, Smit B. *Understanding Molecular Simulation*. second ed.. San Diego, CA: Academic Press; 2001.
- [20] Byun HS, El-Naggar MY, Nakano A, Vashishta P. A derivation and scalable implementation of the synchronous parallel kinetic Monte Carlo method for simulating long-time dynamics. *Comput Phys Comm* 2017;219:246–54.
- [21] Plimpton S. Fast parallel algorithms for short-range molecular dynamics. *J Comput Phys* 1995;117(1):1–19.
- [22] Franck C, Hong S, Maskarinec SA, Tirrell DA, Ravichandran G. Three-dimensional full-field measurements of large deformations in soft materials using confocal microscopy and digital volume correlation. *Exp Mech* 2007;47(3):427–38.
- [23] Rajadhyaksha M, Grossman M, Esterowitz D, Webb RH, Rox Anderson R. In vivo confocal scanning laser microscopy of human skin: Melanin provides strong contrast. *J Invest Dermatol* 1995;104(6):946–52.
- [24] Webb RH. Confocal optical microscopy. *Rep Progr Phys* 1996;59(3):427–71.
- [25] Diaspro A. *Confocal and Two-Photon Microscopy: Foundations, Applications and Advances*. Wiley; 2011.
- [26] Slater M, Usoh M, Steed A. Taking steps: the influence of a walking technique on presence in virtual reality. *ACM Trans Comput-Human Interact* 1995;2(3):201–19.
- [27] Templeman JN, Denbrook PS, Sibert LE. Virtual locomotion: walking in place through virtual environments. *Presence: Teleoperators Virtual Environ* 1999;8(6):598–617, (in en).
- [28] Wann JP, Rushton S, Mon-Williams M. Natural problems for stereoscopic depth perception in virtual environments. *Vis Res* 1995;35(19):2731–6.
- [29] Zhou G, et al. Molecular simulation of MoS<sub>2</sub> exfoliation. *Sci Rep* 2018;8(1):16761.
- [30] Amey A, et al. Polytypism in ultrathin tellurium. *2D Materials* 2019;6(1). 015013.
- [31] Wei T, et al. Aromatic polyamide reverse-osmosis membrane: An atomistic molecular dynamics simulation. *J Phys Chem B* 2016;120(39):10311–8.
- [32] Atomify - a real time LAMMPS visualizer. <https://github.com/ovilab/atomify>.

