Mechanical Characteristics and Design of Faceted Silicon Carbide Nanosystems using Molecular Dynamics and Machine Learning

Alexander Harold Sexton

Thesis submitted for the degree of
Master in Computational Science: Materials Science
60 credits

Department of Physics
Faculty of mathematics and natural sciences

UNIVERSITY OF OSLO

Autumn 2021

Mechanical Characteristics and Design of Faceted Silicon Carbide Nanosystems using Molecular Dynamics and Machine Learning

Alexander Harold Sexton

Mechanical Characteristics and Design of Faceted Silicon Carbide Nanosystems using
Molecular Dynamics and Machine Learning

http://www.duo.uio.no/

Printed: Reprosentralen, University of Oslo

Abstract

This thesis examines the mechanical properties of a faceted silicon carbide system using molecular dynamics simulations and machine learning. Silicon carbide nanoparticles have in simulations been observed to transform into a faceted shape. Here, a porous system is studied, where the voids are in the shape of a faceted nanoparticle. The faceting process of the pore during energy relaxation from its initial shape of a sphere is studied, and the potential energy of the system is well described by a stretched exponential relaxation, $\exp\bigl(-(t/\tau)^{\beta}\bigr)$. The time-stretching exponent is found to be consistent with an exact solution, $\beta = 1/2$, for surface diffusion in a system with random, static traps. The strength of this system is studied in the simulations by subjecting it to a compressive force until failure, and the Young's modulus and yield strength is measured, revealing linear dependencies on temperature. The rupture time of the system is also studied under conditions of high constant loads. These are found to increase exponentially below a temperature dependent critical stress value.

Repeated simulations of a larger system with numerous faceted inclusions are performed, where the strength is studied as a function of the spatial locations of the inclusions. Here, a machine learning model is used to approximate the the yield stress for a given geometry. The model, a convolutional neural network, achieved an R^2 score of 0.57 using the results from 193 simulations. We use this model to predict the yield stress of approximately 4 million unseen geometries, identifying high and low strength structures, a subset of which is subsequently verified by simulation. The model is retrained on the simulation results in an evolutionary manner, improving its predictive performance to an R^2 score of 0.87.

This methodology is used for inverse design of the porous system, such that it exhibits a predefined strength characteristic. The inclusions are permitted to occupy locations on a regular $3 \times 3 \times 3$ grid. Designs pertaining to the highest and lowest possible strengths are pursued, where it is found that this method is most successful in identifying high strength geometries in this specific system. The predictive model identifies designs of yield strength 10.91 GPa, compared to a baseline random search which gave a maximum strength of 9.67 GPa. We find that the high strength geometries have inclusions which are stacked vertically along the axis of applied stress. Those of low strength are found to organize similarly, but where the row of pores is rotated 45° out from this axis. Visually, the geometries of these two extremes can respectively be perceived as *elevators* and *escalators.*

Acknowledgements

I started working on this thesis with the expectation that it would prove challenging, but at the same time I had the rather naïve idea that once I got going, it would be relatively straightforward. Surprisingly, this was not the case, and I am incredibly grateful for the invaluable guidance provided by my supervisors, Anders Malthe-Sørenssen and Henrik Andersen Sveinsson. Not only have they been vital to the progress of my work, but also a source of inspiration.

I have always had an interest for physics, but it was not until I attended the lectures of Morten Hjorth-Jensen that I was captivated by *computational* physics. This sparked an immediate fascination for programming and how it can be used for solving problems in the natural sciences - Most significant was the learning benefit which was gained by actually defining these problems in code. I would like to thank him for his enthusiasm in teaching, which has most definitely had an impact on the path I have ended up choosing.

I would also like to thank the whole community at the Center for Computing in Science Education for contributing to a great environment, both socially and academically. I may have learned more this past year from discussions over lunch, than through countless hours of delving into books. In this regard, I want to thank Even Marius Nordhagen especially, for insightful discussions of simulation techniques and physics in general.

Lastly, I would like to express my gratitude towards my ever-caring girlfriend Lotte, who has been a pillar of support throughout this whole process. If this has been any indication, you will make a great mother to our girl currently jumping around in your belly.

Contents

Chapter 1

Introduction

1.1 Topics

Understanding the strength of materials has always been important for the advancement of technology, from tools used in the stone age, to carbon fibers used today. In materials science, we are often interested in utilizing what we know about the strength and failure mechanisms of materials in order to design them with optimized properties, such as high strength to weight ratios. With the recent machine learning tsunami, many applications have been found within the field of materials science, where techniques used in computer vision and deep learning especially have proven to be useful in predicting material properties and their strength [1, 2]. Through the integration of data mining and machine learning, a new paradigm is emerging in the design of materials, where it is now possible to build predictive models using the results of physics simulations in the discovery of new meta-materials with high-strength properties, such as titanium alloys [3] and highly stretchable graphene sheets [4].

In this thesis I study the mechanical properties of *faceting* systems, i.e. nanostructures with surfaces that are made up by facets oriented along certain planes. The shape of these structures, or *nanocrystals*, are uniquely determined by the geometry which minimizes the surface free energy [5], though their sizes are constrained by an energy barrier which becomes prohibitively large for facet sizes greater than typically one nanometer [6]. The resulting geometries can play a large role in determining the properties of the nanocrystals [7]. Recently, high-temperature silicon carbide was observed in a molecular dynamics simulation to transform into a faceted nanoparticle, where the resulting shape resembled that of a dodecahedron [8]. Silicon carbide is an attractive compound due to its high mechanical strength [9] and melting point [10]. It is chosen as the objective of study because it is the only ceramic yet to be observed which exhibits this faceting behaviour in a molecular dynamics simulation [8]. Rather than studying the actual nanoparticles themselves, this thesis examines a system which contains *pores* in the shape of these nanoparticles. We wish to design systems of faceted structures with specific mechanical properties, and this is a step in the direction of doing so.

In addition to experimental testing of the behaviour of a material under stress, it is of great interest to examine *why* it behaves in this way. Such questions can be answered by studying these systems on the atomic scale, and molecular dynamics is an excellent choice for such a task. Molecular dynamics (MD) is a simulation method that is founded on the assumption that the interactions between atoms behave classically, and thus their trajectories can be solved by integrating Newton's equation of motion, given a sufficiently accurate description of these interactions. We use MD as a coarse-grained approximation to the behaviour of atoms, and this currently allows for simulating systems consisting of up to billions of atoms on millisecond timescales [11]. Most importantly, it gives us a method of explaining observations made on a global scale by studying the underlying mechanisms on the an atomic scale. Of primary relevance to this thesis, this approach can connect the mechanisms contributing to the material fracture with mechanical loading.

The first topic of this thesis relates to the dynamics of a single pore, where the focus is the faceting process itself. Of immediate interest is determining whether it is at all possible to observe a faceting effect in porous silicon carbide - In addition, this study seeks to determine whether it is possible to measure the *force of crystallization* of the faceting process. Growing nanocrystals confined within porous materials have long been known to generate stress [12], and understanding the underlying mechanisms is important, as this process, salt crystallization in particular, is a major cause of reduced longevity in materials used in construction and underground engineering [13], as well as in the deterioration of historical monuments and sculptures [14]. Weathering effects cause salt transportation through the porous network of the material, providing the crystal with the necessary reagent to facilitate growth [14]. This gives rise to a *crystallization pressure*, which may lead to crack propagation within the porous material [15], ultimately damaging it. Experiments have reported induced pressures in the order of 150 MPa [16]. Previous work in trying to gain an understanding of the mechanisms of this growth process has largely been based on a thermodynamic framework which provides a solute-crystal equilibrium condition involving the supersaturation of the liquid and the salt characteristics [17]. This makes it difficult to separate the effects of surface diffusion from adsorption and dissolution effects in the surface energy of the crystal. The physical system in this thesis, on the other hand, is of a pure substance where the only effect during growth is the reorganization of surface atoms. Here, this process is studied at different temperatures, and the results are compared to the *Kohlrausch-Williams-Watts* [18] or *stretched exponential* relaxataion.

The second topic of this thesis is dedicated to investigating the strength of a single pore system under different stressed conditions. Within the framework of continuum mechanics, we have a theory for describing the *elastic* properties of materials in general. Determining a materials actual strength or how it behaves in the plastic regime is a little more tricky, as this requires either a theory or model specific to the material and its geometry, or actually performing an experiment in order to measure it. The deformation process of materials on the atomic scale may be facilitated by imperfections in the crystal lattice known as *dislocations* or *slip planes*, which propagate throughout the material and enable plastic deformation [19]. The activation of a particular dislocation occurs when the stress acting in the direction of the slip reaches a critical value [20, p. 331]. The faceted pore in the silicon carbide system of this study contains geometric irregularities and sharp edges which may be regarded as crack tips, and thus potential nucleation sites for dislocations due to high concentrations of stress in such regions [21, 22]. The elastic properties of this system and its transition to the plastic regime by compressing it are studied at different temperatures. The relaxation time for the system to yield under a constant load is also compared to *Zhurkov*'s kinetic theory of rupture [23].

The final topic of this thesis looks at the strength of a much larger system consisting of several faceted inclusions at various locations within the material, organized in a grid pattern. The goal is to design the system, i.e. identify the optimal pore locations, such that it exhibits a desirable strength characteristic. In order to optimize for strength, it is not immediately obvious where these pores should be distributed. In addition, with increasing system sizes the number of possibilities which must be explored scales very rapidly. This is hardly viable to investigate by brute-force molecular dynamics alone - However, by utilizing machine learning methods, a mapping can be made between the geometry of the physical system and its strength [24], alleviating the need for large amounts of costly simulations. Using a methodology inspired by *Hanakata et al.* [4], I perform a handful simulations of systems with differing pore locations and use a machine learning model to approximate the yield strength. This model is subsequently used to effectively search through the entire space of possible configurations and identify those with the optimal strength characteristics, which are then verified by molecular dynamics simulations.

1.2 Goals

The work presented in this thesis is the result of attempting to achieve the following goals:

1. Investigate whether it is possible to grow a pore with an inverse faceted shape in a system of silicon carbide using molecular dynamics.
2. Study the faceting process, measure the pressure induced by it and compare this with relevant theories for energy relaxation.
3. Study the effects a faceted inclusion has on the mechanical properties of silicon carbide, and how this depends on temperature.
4. Simulate systems consisting of numerous faceted pores and use predictive modelling to approximate its yield strength.
5. Investigate whether this model can be used to effectively design the multi-pore system and identify optimal geometries with respect to specific strength characteristics.

1.3 Contributions

With the goals listed in the previous section in mind, I here provide a list of my contributions.

- Shown that it is possible to observe a faceting effect in nanoporous silicon carbide.
- Measured the force of crystallization of the faceting process and its energy barrier.
- Measured the Young's modulus and yield strength of this system under uniaxial compression, and its time to failure under constant external loads.
- Implemented a simple interface for creating molecular systems with faceted pores organized in a grid-pattern.

- Implemented a crude pipeline for generatively training a machine learning model based on simulation results, which can be used to search for system geometries that exhibit specific strength properties. Demonstrated that this method can be used effectively for inverse design of such systems, and found the presumably optimal geometries for high and low strength systems of nine faceted pores located on a regular $3 \times 3 \times 3$ grid.

1.4 Thesis structure

Part I presents the methods used in this thesis, as well as the underlying theories. It starts with the introduction of the basic continuum mechanical concepts of stress and strain, as well as several theoretical predictions which are used, before moving on to describing how molecular dynamics simulations are performed. This is followed by a review of the basic theory of artificial neural networks and how they can be used in predictive modelling.

Part II contains the results of the series of experiments performed in this study. It starts with the results of simulations of a rather simple system, subsequent sections are generally motivated by their predecessors, so the reader is encouraged to follow along in a sequential manner. The final part of the results are dedicated to more complex simulations, where molecular dynamics and machine learning are combined in order to optimize the mechanical properties of the system in question. In the final chapter of Part II, the work is summarized together with an outlook on possibilities for future research.

Part III is the appendices, which contains some supporting material to Part II.

Part I

Background Theory and Methods

Chapter 2

Elasticity and Failure

One of the most fundamental concepts in continuum mechanics is the description of how a system deforms when it is subjected to an external force. This has a rigorous formulation within the theory of elasticity, which will be described in this chapter. In addition, a theoretical prediction for the time-dependent rupture of materials is introduced, as well as a relation which will be used to describe slow relaxation processes.

2.1 Stress and Strain

This section introduces the basic concepts of stress, strain and linear elasticity, and the following derivations are based on the presentation by *Markus J. Buehler* [20, p. 97 - 108].

The relationship between stress and strain in the elastic regime has its root in the famous observation by Robert Hooke that the extension Δu of a system is directly proportional to the applied force, i.e. $F = k\Delta u$, where k is a constant describing the stiffness of the system. Considering a specimen of length L and cross-sectional area A, this relationship can be renormalized,

$$\frac{F}{AL} = \frac{k\Delta u}{AL}. \tag{2.1}$$

From this equation, we define the quantity *stress* as

$$\sigma = \frac{F}{A}, \tag{2.2}$$

and *strain* as

$$\varepsilon = \frac{\Delta u}{L}. \tag{2.3}$$

A common rewriting gives

$$\sigma = E\varepsilon, \tag{2.4}$$

known as Hooke's law. Here, $E = Lk/A$ is the Young's modulus. A generalization of the stresses in a system in the 3D case can be made by explicitly considering the spatial directions of the forces acting on the surfaces of a volume element, illustrated in Figure 2.1. This gives rise to the *stress tensor*, which is defined as

$$\sigma_{ij} = \lim_{A \to 0} \frac{\Delta P_j}{\Delta A_i}, \tag{2.5}$$

where j indicates the direction of the force and i the orientation of the surface it acts upon. The diagonal elements σ_{ii} of this tensor are referred to as the *normal stresses* and the off-diagonal elements σ_{ij} are the *shear stresses*. Similarly, the *strain tensor* defines the relative change in the position of a point within a deformed system,

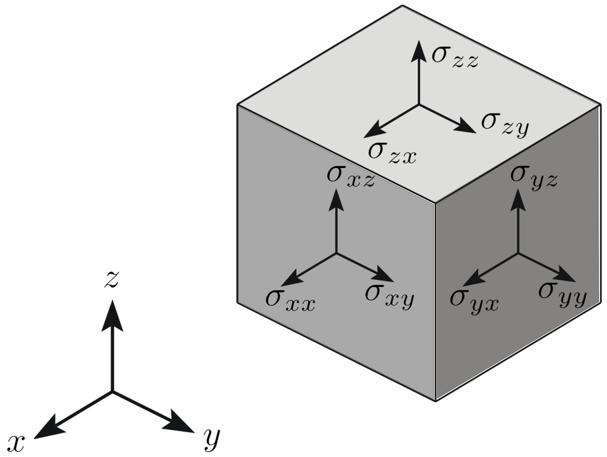


Figure 2.1: Illustration of the stress components acting on the surfaces of an infinitesimal volume element. Adapted from [25].

$$e_{ij} = \frac{\partial u_i}{\partial x_j}. \tag{2.6}$$

This definition assumes a linear relationship between the change in displacement u_i with respect to the direction x_j, and is generally only valid for small deformations. In addition to the strain components ε_{ij} that contribute to deforming the system, this formulation also includes the rigid body rotation ω_{ij} which does not have an effect on the shape of the system. It is therefore common to to express it as a linear combination of these two parts, i.e.

$$e_{ij} = \varepsilon_{ij} + \omega_{ij}, \tag{2.7}$$

where

$$\varepsilon_{ij} = \tfrac{1}{2}(e_{ij} + e_{ji}) \quad \text{and} \quad \omega_{ij} = \tfrac{1}{2}(e_{ij} - e_{ji}). \tag{2.8}$$

The stresses and strains in a system can be related through a generalization of Hooke's law,

$$\sigma_{ij} = c_{ijkl}\varepsilon_{kl}, \tag{2.9}$$

where c_{ijkl} is the *elasticity tensor*. This fourth-order tensor describes the elastic properties of a material in the most general way. Through a series of energy arguments provided by *Buehler* [20, p. 105-106], it can be shown that the stress tensor can be written as

$$\sigma_{ij} = \frac{\partial \Psi}{\partial \varepsilon_{ij}}, \tag{2.10}$$

which leads to the definition of the elasticity tensor,

$$c_{ijkl} = \frac{\partial^2 \Psi}{\partial \varepsilon_{ij} \partial \varepsilon_{kl}}. \tag{2.11}$$

Here, Ψ is the free energy density, and this shows that the elastic properties of a material are intimately related to how its internal energy is changed under deformation. In total, the elasticity tensor has 81 entries, but this may be reduced to 21 due to the symmetries of the stress and strain tensors, i.e. $c_{ijkl} = c_{jikl} = c_{ijlk}$, and because of the fact that the order of differentiation in Equation 2.11 does not matter. Furthermore, the relationship between stress and strain for an isotropic solid may be described by only *two* independent parameters, known as the *Lamè coefficients* λ and μ. Their relation to the elasticity tensor are given by

$$\mu = c_{1212} = \tfrac{1}{2}(c_{1111} - c_{1122}) \tag{2.12}$$

$$\lambda = c_{1122} \tag{2.13}$$

$$\lambda + 2\mu = c_{1111}, \tag{2.14}$$

from which important elastic moduli can be calculated, such as the Young's modulus.

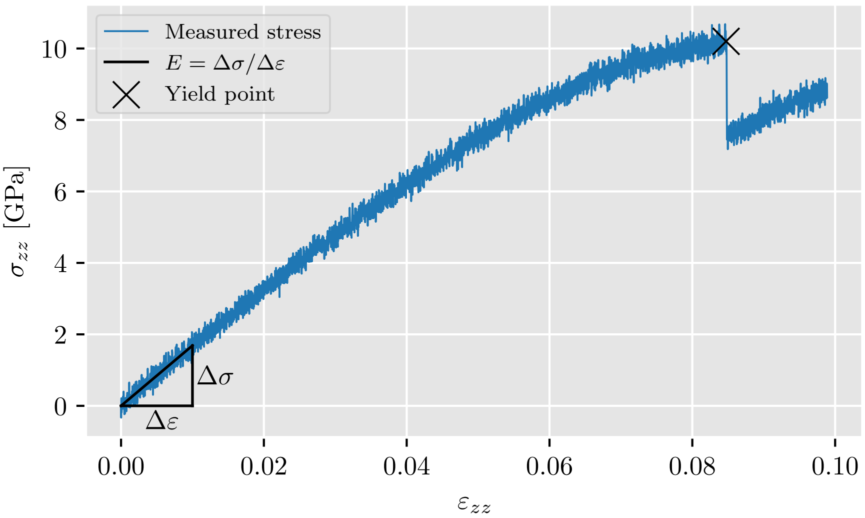


Figure 2.2: Stress-strain curve of silicon carbide in a molecular dynamics simulation. The plot serves to illustrate how the Young's modulus and yield stress and strain can be estimated.

These coefficients can be calculated given that the free energy function of the material in question is known, or by measuring how it changes due to deformation. A more practical approach is to use the stress-strain relationship in Hooke's law (Eq. 2.9) directly. Of interest in this thesis in particular is the Young's modulus. In a molecular simulation we can measure the (scalar) stress as a function of strain directly, and estimate the Young's modulus as the slope of this curve, as shown in Figure 2.2. The yield stress and strain is also pointed out in the figure, which indicates the end of the elastic regime of the material, and the beginning of permanent, plastic deformation. The figure also illustrates why Hooke's law generally is valid for small deformations only, as the stress-strain relationship in a material is not always linear.

2.2 Time-dependent Rupture

The durability of a material is intuitively related to the amount of stress it is exposed to - high loads will cause the material to fail more rapidly than smaller loads, for instance by the propagation of cracks in methane hydrates [26] or in sheets of paper [27]. Slow rupture has long been described as a consequence of thermally activated creep flow [28]. Early observations demonstrated an exponentially decreasing rupture time with increased loads, proposed to be proportional to the strain-rate in Eyrings's theory of viscous flow [29]. This has has led to what is referred to as the *BBZ*-model for time-dependent rupture, which encapsulates the ideas of *Busse et al.* [28], *Bueche* [30] and *Zhurkov* [23], summarized nicely by *Vanel et al.* [27]:

The strain-rate in Eyring's reaction-rate theory is given by

$$\tau^{-1} \sim \tau_0^{-1} \exp\left(\frac{-E_0}{k_b T}\right) \sinh\left(\frac{\sigma V}{k_b T}\right). \tag{2.15}$$

where σ is the applied stress, E_0 is an energy barrier, V an activation volume, τ_0 is the characteristic breaking time, T is the temperature and k_b is Boltzmann's constant. The exponential in Equation 2.15 is the Arrhenius factor, and the sinh-term accounts for the thermally activated probability that molecules have to jump a unit distance in either direction of the viscous flow. When the externally applied pressure is large, such that $\sigma V \gg k_b T$, the probability of jumping against the direction of flow is negligible, and so the equation can be reduced to

$$\tau = \tau_0 \exp\left(\frac{E_0 - \sigma V}{k_b T}\right). \tag{2.16}$$

The parameters E_0 and V in this model pertain to the bond energy of the molecules in the material in question, and the critical distance (cubed) at which the bonds are broken.

2.3 Stretched Exponentials

Many phenomenons in nature are subject to exponential decay, such as the radioactivity of materials or the rates of chemical reactions. Such processes have in common that the rate of change of a given quantity is proportional to its current value, which leads to the governing equation to be of the form $f(t) \sim \exp(-t/\tau)$. Relaxation processes in many disordered systems are often observed to decay slower than any exponential. This is seen in, for instance, glassy materials. These are considered to be mechanically stable even though the free energy of the system is not at a local minimum, i.e. a glass is continuously relaxing, but at a rate often too slow to measure [31]. Such observations have led to a phenomenological generalization of the decay function:

$$f(t) = \exp\left(-\,(t/\tau)^{\beta}\right), \tag{2.17}$$

first introduced in 1854 by *Rudolf Kohlrausch* in his studies of discharging capacitors [32]. In more recent years it has come to be known as the *Kohlrausch-Williams-Watts* (KWW) relation, after the *Graham Williams* and *David Watts* used it in examining the dielectric relaxation of polymers [18]. The time-stretching factor β may be dependent on the material in question, as well as the underlying mechanism. For instance, the density of diffusing atoms in a medium of static traps has a long-time behaviour which is described by this equation [33]. Here it has an analytical value $\beta = d/(d+2)$, where d is the spatial dimensionality of the system.

Although the KWW relation is mostly empirical, *Palmer et al.* [34] show that this result emerges naturally when considering glassy dynamics as being *hierarchically constrained.* They model this behaviour as if it has a wide range of relaxation times, where faster degrees of freedom successively constrain slower ones. In their original paper [34], they associate this with the movement of a small group of atoms A, enabling the movement of a larger group B if group A moved in just the right way, leading to increasingly longer relaxation times.

Chapter 3

Molecular Dynamics

Molecular dynamics (MD) is the main method used for studying the strength properties of a nanoporous silicon carbide system in this thesis. MD is a logical choice for this task, as it allows for examining the dynamics of the system as a whole, as well as the behaviour of individual atoms which contribute to the global observations. In this chapter I will give an overview of the underlying theory of molecular dynamics as well as how it can be performed in practice.

3.1 Potentials and Forces

The behaviour of a system will depend on it's potential energy, which in the simplest case depends only on the pairwise positions the individual particles,

$$U = U\left(\{\mathbf{r}_i\}\right) = \sum_{i<j} U(\boldsymbol{r}_{ij}). \tag{3.1}$$

The potential energy in a system gives rise to a force field, which can be evaluated at any point by taking the gradient of the potential energy function. The contributions to the force acting on a particle i is therefore

$$\boldsymbol{F}_i = -\sum_j \nabla U_{ij}. \tag{3.2}$$

Using this fact, the trajectory of a particle can be determined by calculating its acceleration from Newton's second law,

$$\boldsymbol{a}_i = -\frac{1}{m_i}\sum_j \nabla U_{ij}. \tag{3.3}$$

This is in principle all the information which is required to determine the time-evolution of the system in question. How these equations can be solved, will be explained in Section 3.2.

3.1.1 Lennard-Jones

One of the simplest interatomic pair potentials we have, was developed in the 1920's by *John Lennard-Jones* [35], and has proven to be an important building-block in the more advanced models of later years:

$$U(r_{ij}) = 4\epsilon\left[\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6}\right]. \tag{3.4}$$

Although simplistic in it's form, the Lennard-Jones (LJ) model is able to reproduce many of the fundamental thermodynamic properties of noble gases to high precision [36]. It encapsulates two of the most intrinsic atomic interactions; the short-range Pauli repulsion and the attractive van der Waals interaction, described by the $(1/r)^{12}$ and $(1/r)^6$ terms respectively. The general form of the LJ potential takes two parameters which are specific to the type of atom, ϵ and σ. These parameters, known as the characteristic energy and length, determine the depth of the potential well and the point at which the repulsive and attractive force cancel each other out.

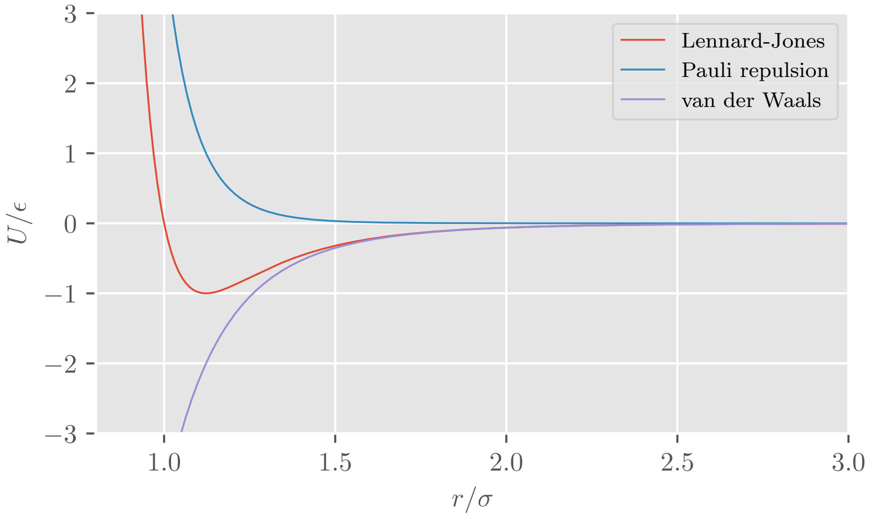


Figure 3.1: Illustration of the Lennard-Jones potential in Equation 3.4 as a function of interatomic distance. Included in the plot is also the Pauli repulsion and the van der Waals interaction. The parameter ϵ determines the depth of the potential well, and σ the point at which it is zero.

The illustration of the LJ potential in Figure 3.1 shows what we intuitively know about atoms; they strongly repel each other at short distances and attract when they are further apart, although decreasingly as the distance grows. This fact can be used to great advantage when conducting simulations, as will be shown in Section 3.5.2. The minimum of the potential is found at the distance where the force is equal to zero,

$$F(r) = \frac{\mathrm{d}}{\mathrm{d}r}U(r) = 24\epsilon\left[\left(\frac{\sigma}{r_{ij}}\right)^{6} - 2\left(\frac{\sigma}{r_{ij}}\right)^{12}\right] = 0, \tag{3.5}$$

which occurs at

$$\frac{r}{\sigma} = 2^{1/6}, \tag{3.6}$$

acting as a natural resting distance between the atoms.

3.1.2 Stillinger-Weber

Pair potentials, simple and elegant as they may be, lack the ability to accurately model the interactions of atoms where more complex physical effects come into play, e.g. the formation of chemical bonds. Symmetrical potentials, such as LJ, may also impose an instability on the diamond structure of the atoms, which is prevalent in many solid-state systems [37]. It is therefore essential that three-body interactions are included in such cases,

$$U(\{\boldsymbol{r}_i\}) = \sum_{i<j} U_2(r_{ij}) + \sum_{i<j<k} U_3(r_{ijk}). \tag{3.7}$$

The three-body term will often be dependent on both the angle between triplets of particles and their distance, illustrated in Figure 3.2.

Crystals such as Silicon, consist of atoms which are held together by strong covalent bonds, and in the 1985 article by *Frank H. Stillinger* and *Thomas A. Weber*, they found it reasonable to approximate this effect with a combination of pair and triplet potentials, known as the Stillinger-Weber (SW) potential [38]:

$$U_2(r_{ij}) = A_{ij}\epsilon_{ij}\left[B_{ij}\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{p_{ij}} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{q_{ij}}\right]\exp\left(\frac{\sigma_{ij}}{r_{ij}-a_{ij}\sigma_{ij}}\right), \tag{3.8}$$

$$U_3(r_{ij}, r_{ik}, \theta_{ijk}) = \lambda_{ijk}\epsilon_{ijk}\left[\cos(\theta_{ijk}) - \cos(\theta_{ijk,0})\right]^2 \tag{3.9}$$

$$\times \exp\left(\frac{\gamma_{ij}\sigma_{ij}}{r_{ij}-a_{ij}\sigma_{ij}}\right)\exp\left(\frac{\gamma_{ik}\sigma_{ik}}{r_{ik}-a_{ik}\sigma_{ik}}\right). \tag{3.10}$$

The two-body interaction term is similar to that of the LJ potential in Equation 3.4, as it allows for both first-order attractive and repulsive effects. More notably, the three-body term accounts for the potential energy in the directional bonds of triplets, where θ_{ijk} is the angle between $\boldsymbol{r}_{ij}$ and $\boldsymbol{r}_{ik}$. Deviations from the equilibrium angle $\theta_{ijk,0}$ will contribute to increased potential energy. The inverse proportionality of bond strength and distance is introduced by the exponential functions, which serve as cut-offs for $r/\sigma \geq a$.

It should be pointed out that in monoatomic systems, the parameters across pairs and triplets would be the same when using this potential, but this general formulation is rather handy, as it can describe the behaviour of structures which involve more than one atom type, e.g., SiO_2 and SiC, given the appropriate parameter set.

3.1.3 Vashishta

In more primitive models, the main forces considered are the repulsive part due to the size of the atoms and the attractive dipole-induced dipole part. This approach of modelling provides a

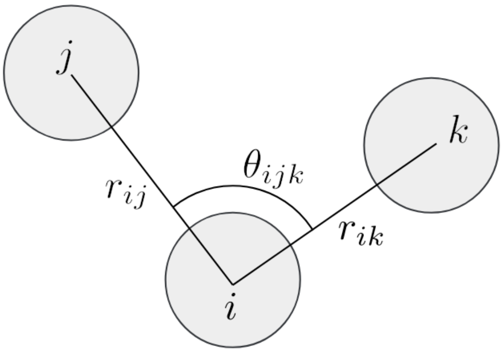


Figure 3.2: Illustration of a three-body interaction. The vectors $\boldsymbol{r}_{ij}$ and $\boldsymbol{r}_{ik}$ between the atoms form an angle θ_{ijk}, which is a contributing factor to the potential energy of particle i.

good approximation to many monoatomic systems. However, in multi atomic compounds, such as silicon carbide, there is a net charge transfer between the elements [39], and as a result it is necessary to account for Coulomb and charge-dipole interactions. These effects are included in a slightly more complicated model, which in 1990 was proposed as an interaction potential for silica, known as the Vashishta potential [40]. Since then, it has been extended to also model indium phosphide [41] and silicon carbide [39].

The Vashishta potential is given as a sum of two and three-body interactions, where the two-body terms for SiC are

$$U_2(r_{ij}) = \frac{H_{ij}}{r_{ij}^{\eta_{ij}}} + \frac{Z_i Z_j}{r_{ij}} e^{-r_{ij}/\lambda} - \frac{D_{ij}}{2r_{ij}^4} e^{-r_{ij}/\xi} - \frac{W_{ij}}{r_{ij}^6}. \tag{3.11}$$

Here, the steric repulsion and van der Waals interaction strengths are represented by the constants H_{ij} and W_{ij}. Coulomb interactions are described by the second term, where Z_i is the effective charge of atom i, and the third term accounts for the charge-dipole interaction with strength D_{ij}. The parameters λ and ξ are screening lengths for the Coulomb and charge-dipole interactions respectively. The exponent η_{ij} determines the strength of the steric repulsion. Similar to the Stillinger-Weber potential in Equation 3.10, a three-body term is necessary to describe the covalent characteristics of the material,

$$U_3(r_{ij}, r_{ik}, \theta_{ijk}) = B_{ijk} \frac{\left[\cos(\theta_{ijk}) - \cos(\theta_{ijk,0})\right]^2}{1 + C_{ijk}\left[\cos(\theta_{ijk}) - \cos(\theta_{ijk,0})\right]^2} \exp\left(\frac{\gamma_{ij}}{r_{ij} - r_0} + \frac{\gamma_{ik}}{r_{ik} - r_0}\right), \tag{3.12}$$

the main difference being that the angular dependence saturates when the deviation from the equilibrium angle $\theta_{ijk,0}$ is large. This allows for reconfiguration of bonds in structural transformation under pressure and in melting [39].

3.2 Integrating the Equations of motion

The equations of motion in a Molecular dynamics system are effectively an N-body problem, with a set of coupled differential equations of the form

$$\frac{\partial^2}{\partial t^2}\boldsymbol{R}(\{\boldsymbol{r}_i\},t) = -\frac{1}{m_i}\nabla U\big(\boldsymbol{R}(\{\boldsymbol{r}_i\},t)\big), \tag{3.13}$$

In order to solve these equations, we need to make use of numerical approximations when integrating the particles' trajectories through time. The most important properties in a scheme of choosing is that they preserve energy and momentum, are time reversible, numerically stable and computationally efficient, i.e we wish the law's of physics to be upheld, and that the implementation makes as few rounding errors and runs as quickly as possible.

There are a number of algorithms to choose from, and one which strikes a good middle ground between each of the requirements is the Verlet algorithm [42], most recently discovered in the 1960's. To derive it, we consider the Taylor expansion of a position coordinate r around a time t in two directions,

$$r(t+\Delta t) = r(t) + v(t)\Delta t + \tfrac{1}{2}a(t)\Delta t^2 + \mathcal{O}(\Delta t^3), \tag{3.14}$$
$$r(t-\Delta t) = r(t) - v(t)\Delta t + \tfrac{1}{2}a(t)\Delta t^2 + \mathcal{O}(\Delta t^3). \tag{3.15}$$

Summing these two gives the equation to the updated position:

$$r(t+\Delta t) = 2r(t) - r(t-\Delta t) + a(t)\Delta t^2 + \mathcal{O}(\Delta t^4), \tag{3.16}$$

where the local error is reduced to $\mathcal{O}(\Delta t^4)$ due to the odd terms cancelling out. No velocity calculations are needed in this scheme, and it relies solely on the acceleration and the position at the two previous time steps. When performing molecular dynamics simulations, however, it is advantageous to have control over the particles' velocities, as we are usually interested in the temperature and kinetic energy of the system, both of which are velocity dependent. We can easily see that the velocity can be inferred as a second-order approximation from eqs. 3.14 and 3.15,

$$v(t) = \tfrac{1}{2\Delta t}\left[r(t+\Delta t) - r(t-\Delta t)\right] + \mathcal{O}(\Delta t^2), \tag{3.17}$$

but this means that the velocity at time t is not known until time $t+\Delta t$. A choice which is often more popular, is to use a mathematically equivalent [43] rewriting of the original Verlet scheme, where the velocity term is explicitly used when updating the position. It is known as the Velocity-Verlet algorithm, and the propagation through time of a particle i is here given by:

$$v_i(t+\tfrac{\Delta t}{2}) = v_i(t) + \tfrac{1}{2}a_i(t)\Delta t \tag{1}$$
$$r_i(t+\Delta t) = r_i(t) + v_i(t+\tfrac{\Delta t}{2})\Delta t \tag{2}$$
$$a_i(t+\Delta t) = -\frac{1}{m_i}\sum_j \nabla U_i\left(r_{ij}(t+\Delta t)\right) \tag{3}$$
$$v_i(t+\Delta t) = v_i(t+\tfrac{\Delta t}{2}) + \tfrac{1}{2}a_i(t+\Delta t)\Delta t. \tag{4}$$

Notice that step (2) must be completed for all particles before the acceleration is calculated in step (3). This algorithm offers both higher numerical stability in the velocity update, due to not taking the difference between two approx. equal quantities (as done in Eq. 3.17). It is also less memory intensive. Most importantly, it requires only one force calculation per particle per time step, which is by far the most time-consuming operation.

3.3 Macroscopic Observables

In a molecular dynamics simulation, the physical quantities we immediately have access to are the positions and momenta of the particles, as well as a description of their interactions. The statistical properties of the system we wish to gain insights into must therefore be expressible as a function of these quantities. The following briefly describes how we calculate the most fundamental thermodynamic properties based on this premise.

In a classical context, the total energy of a system is quantified by the potential and kinetic energy of the particles, the latter of which can be written as

$$E_k = \sum_i \tfrac{1}{2} m_i |\boldsymbol{v}_i|^2, \tag{3.18}$$

where m_i and $\boldsymbol{v}_i$ denote the mass and velocity of particle i. Accordingly, the total energy of the system is

$$E_{\text{tot}} = \sum_i \tfrac{1}{2} m_i |\boldsymbol{v}_i|^2 + U_i, \tag{3.19}$$

where U is determined by the interaction potential. Additionally, by the equipartition theorem, the kinetic energy term can be used to measure the thermodynamic temperature of the system

$$T = \frac{2\langle E_k \rangle}{dNk_b}, \tag{3.20}$$

where k_b is the Boltzmann constant, N and d the number of particles and the spatial dimensionality of the system. The system pressure can simply be found by the equation of state, which is accurate enough for a low-density noble gas. A more precise description commonly used in simulations, however, includes in addition a virial term applicable to both arbitrary interatomic potentials and periodic boundary conditions [44]. The pressure is here given by

$$P = \frac{Nk_bT}{V} + \frac{\sum_i \boldsymbol{r}_i \cdot \boldsymbol{F}_i}{dV}, \tag{3.21}$$

where $\boldsymbol{r}_i$ is the position of atom i and $\boldsymbol{F}_i$ the forces acting upon it.

It should be noted that time averaging is important when measuring all of these quantities, as the instantaneous velocities of the particles are subject to fluctuations from one time step to another.

3.4 Thermostats

A natural result of a molecular dynamics simulation such as it has been outlined thus far, is that the energy of the system is conserved, and therefore samples the phase-space of the microcanonical ensemble. However, we are in many cases interested in studying systems at constant temperature, i.e., the canonical ensemble, and this leads us to requiring a method of controlling it. In the simplest case, this can be achieved by simulating collisions between atoms in the system and virtual atoms in an external heat bath, as is done in the Andersen thermostat

[45]. At each time step, randomly selected atoms have their velocities re-scaled according to a Maxwell-Boltzmann distribution centered around the desired temperature. This thermostat accurately reproduces the canonical ensemble and is useful when equilibrating systems, but the sudden changes in the atoms' momenta does affect dynamics such as the lattice vibrations of solids and the viscosity and diffusivity of liquids [46, 47]. Using stochastic thermostats for long simulations may also introduce other undesirable effects, due to repeating sequences of pseudo-random numbers [48].

A more sophisticated method of simulating the coupling of the system with a heat bath was formulated by Nosé in 1984 [49]. Rather than relying on stochasticity, his proposition was to extend the physical system by an additional degree of freedom, $\tilde{s}$. This parameter acts as a time-stretching factor in the Hamiltonian of this extended system, and is dependent on the difference in the instantaneous and desired temperature. As is shown in the original paper [49], the equations of motion of the extended Hamiltonian samples a microcanonical ensemble, and exchanges heat with the physical system[1] based on the sign of $\dot{\tilde{s}}$. Though this formulation is both deterministic and time-reversible, it leads to sampling of the real-system trajectories at uneven intervals, due to fluctuations in $\tilde{s}$ [50]. The issue of virtual-time sampling was later solved by Hoover, and resulted in perhaps the most popular thermostat in use, the Nosè-Hoover thermostat [51]. Here, a slightly different set of equations is introduced, which is entirely composed of real-system variables. The resulting (Lagrangian) equations of motion are here given by [50]

$$\boldsymbol{a}_i = \frac{\boldsymbol{F}_i}{m_i} - \gamma \boldsymbol{v}_i, \tag{3.22}$$

where γ is a thermodynamic drag-coefficient dependent on the difference in temperature of the system and the heat bath. This idea was also extended to the isothermal-isobaric case, where an additional drag coefficient is introduced to control the pressure [51].

3.5 Computational Considerations

In a molecular dynamics simulation, the atoms are represented as point-particles that each have a spatial coordinate in a virtual *simulation box* defined by us. This gives rise to several challenges that must be considered carefully. The size of the simulation box is mainly limited by the number of atoms N which we are able to simulate, $V = N/\rho$, where V is the box volume and ρ the atom density. A naïve implementation of a simulation would require an order of $\mathcal{O}(N^2)$ force calculations per time step for a two-body interaction potential, and $\mathcal{O}(N^3)$ for three-body interactions. This, combined with the fact that a time step typically must be chosen in the pico to femto second range, quickly puts hard constraints on the size of the system and the length of time we can possibly hope to simulate it. The number of atoms in a single grain of sand is in the order of 10^{19} - So if our goal with a MD simulation is to provide information about the properties of macroscopic samples, it is obvious that certain approximations must be introduced.

[1] which can be recovered by a set of conversion factors

3.5.1 Boundary Conditions

The simulation box is rather finite, and in order for the system to mimic a bulk material, we need a method of tricking a particle into believing that the system extends beyond the boundaries of the box. This is achieved by employing *periodic boundary conditions*. It involves two steps; allowing particles at the edge of the simulation box to pass through and wrap around to the other side, and computing the interactions between two particles based on their minimum displacement. E.g. two particles placed at a distance Δx from two opposite sides of the box's edge, would interact as if their displacement were $2\Delta x$ rather than $L_x - 2\Delta x$. The use of periodic boundary conditions is often very effective, and seemingly solves the problem posed, although one should be aware that it may introduce unnatural side effects which are not present in real world systems [52].

3.5.2 Cut-off

In our understanding of nature, the range of intermolecular forces are in principle infinite, which leads to the number of required calculations in a simulation to scale poorly when the number of particles is increased. An important property of these forces is that they tend quickly towards zero for larger displacements, which is something we can use to our advantage. When calculating the forces acting on a particle, it is common to truncate the interaction range by introducing a *cut-off* distance r_c, defining a sphere around each particle, outside of which no other particles interact with it. When truncating the potential energy function, it should also be shifted such that it is continuous and has a value of zero at $r = r_c$, i.e.

$$U(r)_{\text{truncated, shifted}} = \begin{cases} U(r) - U(r_c), & r \leq r_c \\ 0, & r > r_c \end{cases}. \tag{3.23}$$

This speeds up the simulations considerably, as it effectively reduces the global time complexity of the force calculations of both two and three-body interactions to $\mathcal{O}(N)$. However, this approximation can come with side-effect which one should be aware of [53, p. 35].

3.5.3 Initialization

Initial atom placements in a system should be appropriately chosen such that they resemble the real-world structure of the material in question. For a bulk system, this is done by first constructing a *unit cell*, wherein a fixed number of atoms are given specific coordinates. This cell is then duplicated to adjacent positions on a lattice, resulting in the systems simulation box. In a d-dimensional system, the atom density and size can be determined by $\rho = nb^{-d}$ and $V = (Lb)^d$ respectively, where n is the number of atoms per unit cell of side length b, and L the number of lattice points along each spatial dimension.

The simplest lattice configuration consists of placing a single atom at the origin of the unit cell, resulting in a *simple cubic lattice*. Noble gases such as the Lennard-Jones fluid, however, have an equilibrium structure corresponding to the *face-centered cubic* (FCC) lattice, and it is therefore more logical to arrange them in this way to begin with. This unit cell has one atom placed at the origin, and three at the center of the connecting faces. On the other hand, a compound

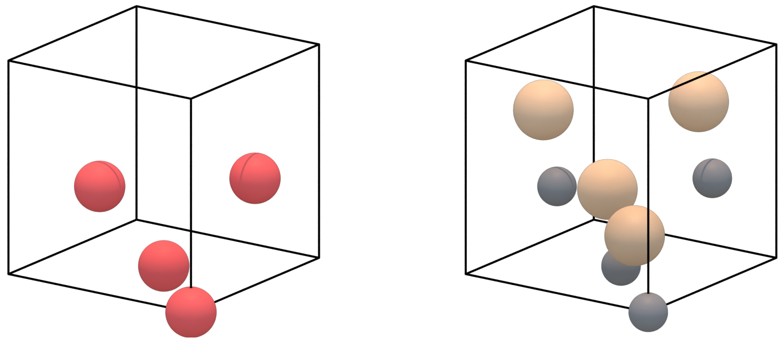


Figure 3.3: Unit cells of a face-centered cubic lattice (left) and the cubic lattice of SiC in the 3C-polytype crystal structure(right). Illustration constructed using Ovito [54].

material like silicon carbide can exist in various structure formations, but only one of them can be represented by a cubic unit cell, namely the 3C-polytype [55]. This configuration consists of four carbon atoms located at the same positions as in the FCC cell, with the addition of four silicon atoms, which are tilted relative to the positions of the carbon atoms. The FCC and SiC-3C lattices are illustrated in Figure 3.3.

Chapter 4

Neural Networks

Artificial neural networks (ANNs) are a class of machine learning models inspired by the neurons within the biological brain and their connections. In a general sense, a neural network is used to map a set of inputs or *features* to one or several output values, also known as *targets*, without any prior knowledge of how the two are related. In the context of *supervised learning*, they achieve this by gradually learning from examples, thereof the term machine *learning*. There exist many neural network architectures, and their strengths are often dependent on the type of data we wish to map, e.g. images with spatial dependencies or time-series data which may be dependent on some previous state. In this section I will introduce the theory behind two commonly used architectures, *dense* and *convolutional* neural networks.

4.1 Dense Neural Networks

The simplest ANN is known as a *fully-connected feed-forward*, or *dense* neural network (DNN), and forms the basis for how neural networks work in general. It consists of an *input* layer, one or more *hidden* layers and one *output* layer, where each of the layers have a certain number of *nodes*. All nodes in one layer have a connection to each of the nodes in the following layer, and the strength of these connections are determined by their individual *weights*. The network makes predictions on a set of feature values, which are propagated from the input layer in a sequential manner throughout the network, until it reaches the output layer, representing the target value we want to be able to predict. This is illustrated in Figure 4.1, where the nodes are represented as white circles and the arrows represent the connections. In this schematic, the output layer consists of a single output node, but it may also have several outputs, for instance if we are dealing with multivariate regression or classification.

4.1.1 Forward propagation

Passing a data sample from the input layer, through the network and to the output layer is called a *feed-forward pass*. This process is done by multiplying the values of the nodes in a given layer

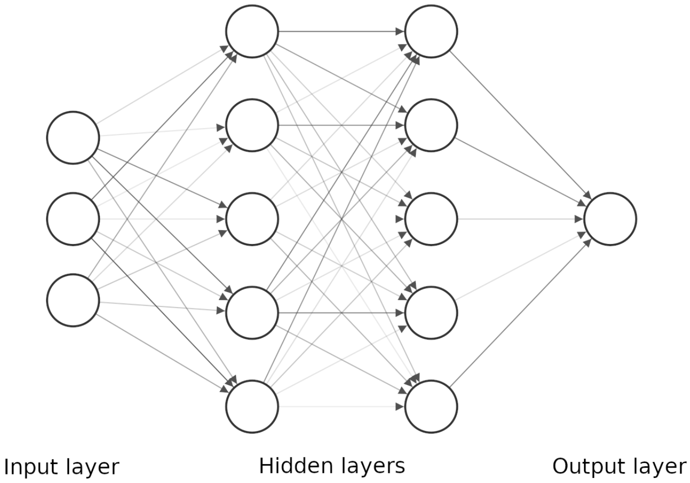


Figure 4.1: An illustration of a dense neural network consisting of the input layer, two hidden layers and an output layer of one node. The circles represent the nodes and the connections are shown as arrows. Varying opacity of the arrows is used to illustrate that some connections may be stronger than others. Created using NN-SVG [56].

with the weights of the connections to the following layer, until the output layer is reached. This can be represented as a series of matrix multiplications, where for a given layer l, the resulting values are given by

$$\boldsymbol{z}_l = W_l \boldsymbol{a}_{l-1} + \boldsymbol{b}_l. \tag{4.1}$$

Here, the weight-matrix W_l has one row for each of the nodes in layer l and one column for each output $\boldsymbol{a}_{l-1}$ in the preceding layer. This transformation also includes a bias term $\boldsymbol{b}_l$ for each of the nodes in layer l. These weights and biases of the nodes are ultimately the parameters which will be learned by the network. In order to introduce non-linearity into the network, the values $\boldsymbol{z}_l$ are passed through an *activation function* ϕ, which is what determines the actual activation of the nodes, i.e.

$$\boldsymbol{a}_l = \phi(\boldsymbol{z}_l) = \phi(W_l \boldsymbol{a}_{l-1} + \boldsymbol{b}_l). \tag{4.2}$$

A common choice in deep neural networks is the *rectified linear unit* (ReLU) function, which is easy to compute and does not suffer from vanishing gradients [57]. It is defined as ReLU$(z) = max(0, z)$, and in this thesis I use a variant known as *leaky* ReLU,

$$\text{LeakyReLU}(z, \alpha) = \begin{cases} \alpha z, & z < 0 \\ z, & z \geq 0, \end{cases} \tag{4.3}$$

which also allows for negative activations.

The initialization of the network weights plays an important role in how easy it will be to train, and a choice which works well in practice is to draw them from a uniform distribution in the range $r = \pm\sqrt{6/(n_j + n_{j+1})}$, proposed by *Xavier Glorot* and *Yoshua Bengio* [57]. The quantities n_i are the number of nodes connecting two subsequent layers.

4.1.2 Backpropagation and Training

So far I have introduced how data is propagated through the neural network to produce an output, but without performing any actual *training*, it will most definitely not be very useful. The training of the network involves having a number of data samples which are made up by the feature values and the *labels* pertaining to the specific samples. When the inputs are propagated through the network, we compare the output, or prediction, with the label, and the network is trained by adjusting its parameters such that it minimizes the difference of the prediction and the label. This process uses a rather revolutionary algorithm called *backpropagation*, introduced in 1985 by *David E. Rumelhart*, *Geoffrey E. Hinton* and *Ronald J. Williams* [58].

Following a forward pass, we measure the network's *prediction error* using a pre-defined *loss function.* This thesis is oriented around solving a classical regression problem, and so a common choice is to use the *mean squared error* (MSE) for this task,

$$\mathcal{C} = \frac{1}{N}\sum_{i}^{N}(y_i - \tilde{y}_i)^2. \tag{4.4}$$

Here, N is the number of samples, y_i the target value and $\tilde{y}_i$ the network prediction. Based on the loss, the algorithm uses the differential chain rule to propagate the error back through the network, computing the gradient of the loss with respect to all the parameters in the network. Details given in the original paper [58]. Once the gradients have been computed, the weights and biases are updated accordingly. We do this using a technique called *gradient descent* [59, p. 83]:

$$W \leftarrow W - \eta \nabla_W \mathcal{C}(W). \tag{4.5}$$

The parameter η is referred to as the *learning rate* of the network, and determines the magnitude of the weight updates. In practice it is common to instead use schemes which are dependent on some previous state of the weights, such as the *Adam* optimizer [60], which often improves the convergence of the network, and is the optimizer of choice in this thesis.

Computing these gradients during backpropagation is generally quite expensive computationally, and a common technique is to evaluate the model output with a small number of samples, as opposed to one sample at a time. The gradients are then computed based on the average model output from these samples [59, p. 275], known as a *minibatch.* One full pass of the data set back and forth through the model is referred to as an *epoch.* The number of training epochs required for the network to converge may be dependent on many things, including the size of the data set and the architecture of the network.

4.1.3 Model evaluation and design

The *design* of the neural network must be decided on before the aforementioned steps in the previous sections can be performed. This means that we need to specify a number of so-called *hyperparameters* before the model can be trained. These involve, among other things, the number of hidden layers, how many nodes in each layer, the choice of activation function and the learning rate of the optimizer. A simple problem may only require a *shallow* network with few layers, but generally it is not possible to know beforehand which hyperparameters are best for a given problem.

The performance of a network involves using a metric based on the difference between the predictions and labels of the data set. In regression problems, the *mean-squared-error* (Eq. 4.4) is commonly used. Another metric, with its origin in statistical modeling, is the *coefficient of determination* or R^2 *score*, which is a measure of how well the variance in the data is explained by the model. The R^2 score is defined as [61, p. 220]

$$R^2 = 1 - \frac{\sum_i (y_i - \tilde{y}_i)^2}{\sum_i (y_i - \overline{y})^2}, \tag{4.6}$$

where again y_i denotes the label, $\tilde{y}_i$ the prediction and $\overline{y}$ is the mean values of the labels.

Since we usually do not know what type of network architecture is best for a given problem, there are techniques which can be used to evaluate the performances of a range of topologies. One such technique which is used in this thesis, is the so-called *k-fold cross-validation* [62, p. 241]. For a given network architecture, it involves training it with one part of the data, and a different part to test it. The data is split into k equal subsets, or *folds*, illustrated in Figure 4.2. Here, $k-1$ subsets are used as *training data* and the remaining 1 is used for calculating the prediction error of the fitted model, often referred to as the *validation data*. It is important that the prediction error is calculated with data *not* used in training, as we wish to assess its performance on *unknown* samples. Next, the subsets are shuffled and the training and evaluation are redone. This process is performed until each of the k subsets are used for testing, or *validating* the model exactly once. The estimates of the prediction error is then taken as the average of the k estimates. This technique for error estimation can be quite useful for evaluating a range of network architectures, and will likely point to which of them is best for the problem in question. An important note is that the *actual* estimate of the model performance is found by measuring its performance on a part of the data set which has not been involved in the cross-validation process, known as the *test* data.

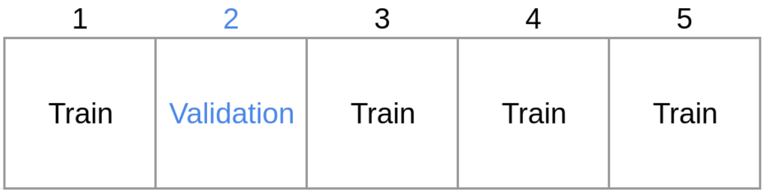


Figure 4.2: Illustration of the subsets of data used for prediction error estimation of a neural network with k-fold cross-validation. Here $k = 5$.

A sufficiently complex neural network is capable of approximating *any* function to an arbitrarily small error [24], and it is therefore important to keep in mind that they are prone to overfitting, i.e. not generalizing to data samples which it hasn't been exposed to during training. This can be avoided by using *regularization* and other techniques. The reader is referred to the excellent book by *Aurélien Géron* [63] for further details on this and other techniques used in network design and training.

4.2 Convolutional Neural Networks

A traditional DNN is heavily reliant on the structure of the data and the connections between specific nodes, whereas the main idea of a convolutional neural network (CNN) is that it learns to extract features and patterns within the input data which is invariant to translations. This makes it an excellent candidate for tasks such as image classification or segmentation, where distinct features may show up in different locations of the image. The main ingredient to a CNN is the *convolutional layer*, which will be explained here.

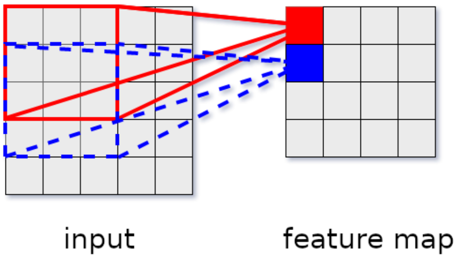


Figure 4.3: Illustration of a 3 × 3 kernel being applied on data in a grid pattern, resulting in a feature map.

The principle of a convolutional layer is that it is built up of a number of *kernels* also known as *filters*. The nodes in a convolutional layer are organized in a grid pattern, and the kernels are slid over these nodes producing a *feature map*, which constitutes the next layer. This process is illustrated in Figure 4.3. For two-dimensional data, the feature map S in layer l is determined by what is known as *convolution* of the nodes I in layer $l-1$ and the kernel K, and it is calculated as [59]:

$$S(i,j) = (K * I)(i,j) = \sum_{q}^{n_x} \sum_{r}^{n_y} I(i-q, j-r) \cdot K(q,r), \tag{4.7}$$

where i, j are the indices of the image and n_x, n_y determine the size of the kernel. The complete feature map is then computed by performing this operation for all i, j. This operation results in activations in the feature map where the kernel correlates with the local values of the nodes. In a CNN it is the kernel parameters which are learned, and accordingly the kernels are tuned such that they identify features in the input data (and subsequent feature maps) which are important to predicting the correct label. In practice, a convolutional layer may be made up of many kernels, resulting in one feature map per kernel, which in turn gives depth to the layer. The convolution is then performed as in Equation 4.7, with the addition of summing over the depth-wise dimension as well. Figure 4.4 shows an illustration of a typical CNN architecture. After the input is fed forward through the convolutional layers, the feature maps in the last layer are flattened and connected to one or more fully connected layers, before finally reaching the output node(s).

CNNs are not restricted to operate exclusively on images. They may also be applied on one-dimensional data, such as time-series, as well as on volumetric data, which is the case in this thesis. The feature maps in such networks are computed as in the 2D-case, but here kernels have one more dimension and are slid along the input data in three different directions, producing

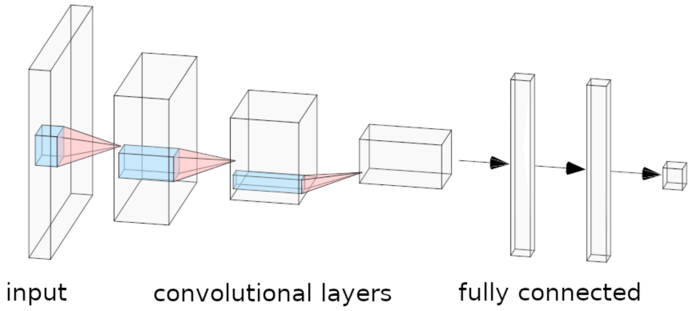


Figure 4.4: Illustration of a CNN with three convolutional layers, two fully connected layers and one output node. The feature maps are shown as grey rectangles, kernels in blue, and the output from the convolutional operations are indicated in red. Created using NN-SVG [56].

a 3D-feature map. Using more than one kernel per layer then results in an output from a 3D-convolutional layer which is four dimensional.

Part II

Simulations

Chapter 5

Supporting Software

The experiments presented in this thesis have been heavily reliant on existing open-source libraries for performing both molecular dynamics and machine learning, namely LAMMPS [64] and PyTorch [65]. Both of these libraries are highly efficient, offer GPU support and have high level APIs which has allowed me to focus on the actual problems at hand, rather than on the details of the implementation of the methods. For creating molecular structures to be used in the simulations, I have used the `molecular-builder`[1] package, which offers a simple interface for creating these. The following section provides a basic example of how I set up one of my simulations using LAMMPS.

The analysis of the results and other implementations not covered by these libraries, are done using Python primarily. The source code and scripts for running the simulations is provided at the GitHub repository https://github.com/alexahs/DeepFacet.

5.1 LAMMPS

This section shows a basic example of how a LAMMPS script is set up, for a simple system under uniaxial compression.

The first part of the script declares variables for the initial temperature, time step and number of time steps for the integration. In this example, the units are of LAMMPS `metal`-style[2], which gives the temperature in units of Kelvin and time in picoseconds. The strain rate of the compression is also declared together with a seed for the initial velocities.

```
variable T          equal 2200
variable dt         equal 0.002
variable Nsteps     equal 500000
variable Nrelax     equal 10000
variable erate      equal -1.5e-5
variable seed       equal 42
```

[1] https://github.com/henriasv/molecular-builder

[2] https://docs.lammps.org/units.html

Following this, the Vashishta potential is chosen as the interaction potential, which also requires an input file, here named `SiC.vashishta`, containing its parameters. The boundary conditions are set, together with the masses of silicon and carbon. The initial atom positions are also read in, and given initial velocities corresponding to the temperature.

```
units           metal
boundary        p p p #periodic
aotm style      atomic

read_data       "atoms.data"
mass            1 12.011 #C
mass            2 28.085 #Si

pair_style      vashishta
pair_coeff      * * SiC.vashishta

timestep        ${dt}
velocity        all create $T ${seed}
```

Further, the system is thermalized for a number of time steps, such that it reaches desired temperature and pressure, the latter of which is set to zero initially. Here, a Nosè-Hoover thermostat and barostat is used, specified by the `fix npt` command. Once the thermalization is done, the simulation box is deformed along the z-axis with the `fix deform`, and the pressure continues to be controlled only in the x and y directions. The actual time integration of the atom trajectories are done by invoking the `run` command, which performs the Velocity-Verlet integration.

```
fix             1 all npt temp $T $T 1 aniso 0 0 10

dump            1 all custom 10000 trajectories.dump id type x y z
thermo_style    custom time temp press ke pe lz pzz
thermo          100
run             ${Nrelax}
unfix           1

fix             2 all deform 10 z erate ${erate}
fix             3 npt temp $T $T 1 x 0 0 10 & y 0 0 10 & nreset 100000
run             ${Nsteps}
```

Here, the desired output from the simulation, such as the temperature and pressure, is also specified with the `thermo_style` command, as well as the atom positions which are written to file by the `dump` command.

Chapter 6

Verifications

A computer does exactly what it is told to do, whether or not its programming is in accordance with the laws of nature or not. In order to have confidence in the results of a physics simulation, it is essential that we have some way of verifying it. In this chapter, some simple molecular dynamics simulations of silicon carbide are performed, and the results are validated by comparing them to previously conducted experiments.

6.1 Melting Temperature of Silicon Carbide

Silicon carbide is the material which will be studied by simulation in this thesis, and in this section I measure its melting temperature. This is done in order to check that it is consistent with the experimental value, and I perform the same simulation procedure which was done in the original paper of the interaction potential by *Vashishta et al.* [39].

The system is initialized as a block of 8 × 8 × 8 unit cells of SiC-3C with a lattice constant 4.3596Å, totalling 4096 atoms. Starting at 300 K, the system is sequentially heated in steps of 200 K, up to 3900 K. At each temperature, the system is thermalized for 80 ps, before the average energy and volume of the system is measured during 20 ps. The pressure is set to 35 bar, and is controlled together with the temperature using a Nosé-Hoover thermo/barostat with damping coefficients of 1 and 10 ps respectively. The time step is set to 2.0 fs and the simulation is run with periodic boundary conditions.

Figure 6.1 shows the energy per particle of the system, together with the system volume. In the interval 3300 K and 3500 K, there is an abrupt change in them both, indicating a phase transition in the material. Subsequently, the melting point is therefore measured to be 3400(100) K. This agrees reasonably well with the reported reference values

$$T = 3250(50)\,\mathrm{K},\ \text{MD [39]}, \tag{6.1}$$

$$T = 3103(40)\,\mathrm{K},\ \text{experiment [10]}. \tag{6.2}$$

As is also mentioned in the discussion by *Vashishta et al.* [39], it is expected that the MD melting

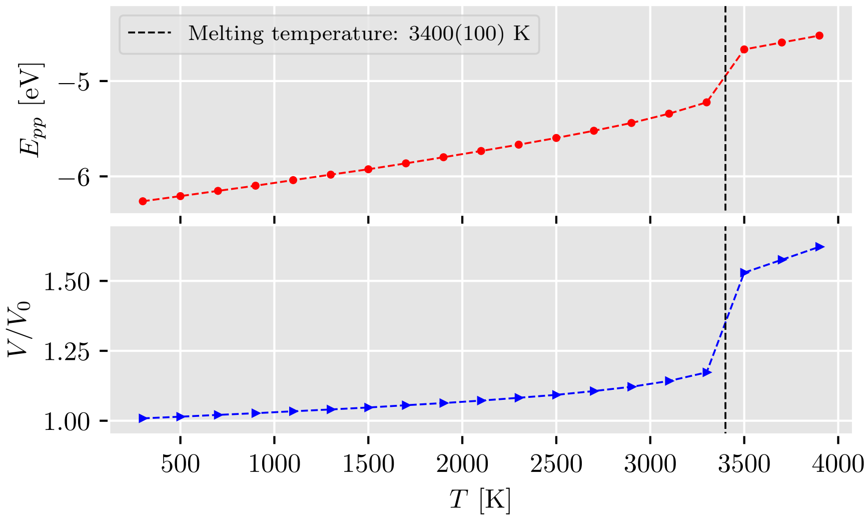


Figure 6.1: Energy per particle and system volume as a function of temperature. V_0 is the volume of the system at $T = 0$ K. The vertical dashed line indicates the measured melting temperature of 3400(100) K.

temperature is higher than the experimental one, since we are here dealing with a perfect crystal and the effects of periodic boundary conditions are also present. The slightly higher melting point in my simulation compared to *Vashishta et al.* [39] is a little mysterious, but it is of little concern as all simulations in this thesis are performed well below the melting point.

6.2 Modulus of Elasticity

The main focus of this thesis is to study the strength properties of a silicon carbide system. The Young's modulus is an important measure of elasticity, which has been well established by previous experiments. In this section I measure this quantity in a simulation of bulk SiC and compare it with two reference values.

The initialization of the system follows the same procedure as in the previous section. The temperature is set to zero and thermalized for 80 ps. Uniaxial compression is applied to the system by deforming it in the z-direction at a rate of $\dot{\varepsilon} = 0.06\,\text{ns}^{-1}$ for 200 ps. The x and y lengths of the simulation box are allowed to expand, such that the corresponding pressure components are 0. I sample the z-component of the system stress and strain every 10 time steps, and use these measurements to determine the Young's Modulus.

By performing a linear fit to the stress-strain curve from this simulation, I measure the Young's Modulus to be

$$E = 311.22(3) \text{ GPa},$$

using `numpy.polyfit`[1]. This corresponds well with the reference values of

$$E = 313.6\,\text{GPa}, \quad \textit{Vashishta et al.}\ [39]\ \text{(MD)}$$
$$E = 314.2\,\text{GPa}, \quad \textit{Feldman et al.}\ [39]\ \text{(experiment)}$$

It should be pointed out that the reference values are calculated from measured values of the components of the elasticity tensor, whereas I have simply measured it directly from the stress-strain curve. For the purposes of this thesis, I find this agreement sufficient.

[1] https://numpy.org/doc/stable/reference/generated/numpy.polyfit.html

Chapter 7

Isolate Inclusions

This chapter describes the simulations of a silicon carbide system with an empty pore located within its center. Initially, the geometry of this pore is perfectly spherical. In the first two sections the relaxation process towards its faceted shape is examined, illustrated in Figure 7.1. Following this, an external compressive force is applied to the system, and its strength is studied at different temperatures. In these experiments, the system is first strained linearly to measure its yield strength, and later the relaxation time of the yielding is studied at varying constant loads.

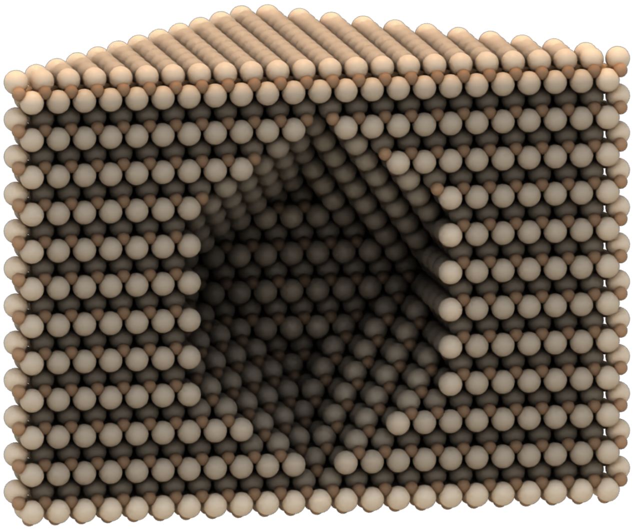


Figure 7.1: Illustration of the silicon carbide system with a faceted inclusion. Constructed using Ovito [54].

7.1 Faceting of a Spherical Inclusion

This describes the simulations of silicon carbide material surrounding an isolate inclusion with the initial shape of a sphere. The equilibrium nanoparticle shape of silicon carbide resembles that of a dodecahedron [8], and these simulations are performed with the goal of investigating whether a similar effect can be observed in this system, specifically whether the space transforms into a faceted shape. This will form the basis for simulations conducted later in this thesis, where the goal is to create a larger system consisting of several such pores. To this end it is important to find a suitable configuration in terms of the system size, such that its scaling is manageable when replicated.

7.1.1 Simulation Setup

The systems are initialized as a cubic block of SiC-3C with a unit cell lattice constant of 4.3596 Å. Atoms located within a sphere of a given radius at the center of the cube are then removed. I use the `molecular-builder`[1] package in Python for setting up the initial atom positions and performing the spherical cut, shown in Listing 1. The system specifications for each simulation are listed in Table 7.1. As was done by *Sveinsson et al.* [8], the temperature is set to 2200 K, controlled by a Nosè-Hoover thermostat with a damping coefficient of 1 ps. The simulation is

```python
from molecular_builder import create_bulk_crystal, carve_geometry, write
from molecular_builder.geometry import SphereGeometry

def create_spherical_pore(size:int, radius:float, filename:str="atoms.data"):
    """
    creates a molecular system of bulk SiC-3C with a spherical pore located at its center,
    and writes the atom positions to file.
    params:
    * size: side length of the simulation box
    * radius: radius of the pore
    * filename: filename to output atom positions
    """

    #create the bulk material
    atoms = create_bulk_crystal("silicon_carbide_3c", [size]*3)

    #define the geometry to cut
    center = size / 2
    geometry = SphereGeometry([center]*3, radius)

    #carve sphere from bulk material
    carve_geometry(atoms, geometry, side="in")

    #write atom positions to file
    write(atoms, filename atom_style="atomic")
```

Listing 1: Python function for creating the initial atom positions of the silicon carbide system with a single spherical pore. The positions are written to a file which can be read by LAMMPS directly.

[1] https://github.com/henriasv/molecular-builder

run with periodic boundary conditions and with a time step of 0.002 ps. The atom positions are dumped every 100 ps and used for visual inspection of the systems in Ovito [54], in order to determine a suitable configuration.

Table 7.1: System specifications in the simulations of the spherical inclusion.

# of unit cells	Sphere radius [nm]	System side length [nm]	# of atoms	Porosity	Simulation time [ns]
26^3	4.0	11.33	114754	0.18	100
21^3	4.0	9.16	48222	0.35	100
17^3	2.5	7.41	33047	0.16	100
12^3	1.5	5.23	12457	0.10	200

7.1.2 Pore Characterizations

Figure 7.2 shows snapshots of the initial and final configurations of the systems with side lengths 11.33 nm and 9.16 nm. Subfigures 7.2b,c show the cross-sectional snapshots of the system in the (100) and (110) planes respectively, where we see that the surface of the pore has transformed to a crystalline shape with 12 distinct (110) facets. In Subfigure 7.2c we can also see slight formations of hexagonal crystal structures, propagating out from one of the pore vertices (indicated by the orange atom coloring). Subfigures 7.2e and 7.2f show that the 9 nm simulation box is too small for isolate growths of a pore of this size. The edges of the crystal in the (100) planes reach the boundaries of the simulation box, fusing with itself and effectively creating a continuous porous region.

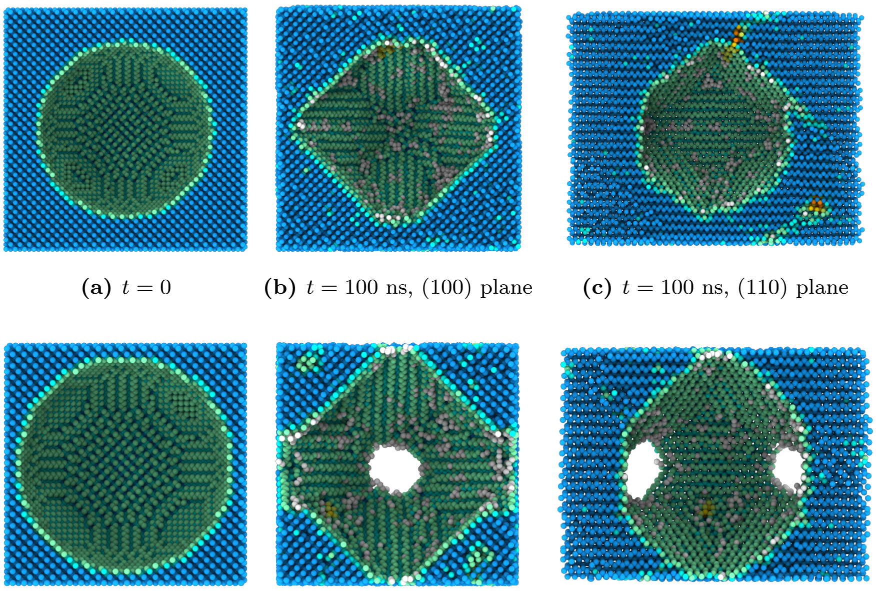
Similarly, Figure 7.3 shows the faceted pore of the two smaller systems. The familiar crystal shapes are visible here as well, but they are not as clear-cut as in the two larger systems in Figure 7.2. The smallest system needed to be simulated for 200 ns before the hexagonal shape was recognizable in the (110) cross-section (Subfig. 7.3f).

Lastly, Table 7.2 lists the hardware performance when simulating the four systems, showing the relative expense of the large and small system sizes.

Table 7.2: Hardware performance when simulating the four systems on a Nvidia Tesla P100 GPU.

Sphere radius [nm]	System side length [nm]	Performance [ns/day]
4.0	11.33	12
4.0	9.16	37
2.5	7.41	54
1.5	5.23	104

The largest system configuration, depicted in the top row in Figure 7.2, clearly yields the most distinctive crystal shape, though the cross-sectional snapshot of the (100) plane (Fig. 7.2b) shows that some of the facets are not fully grown after 100 ns. In the MD simulations of a silicon carbide nanocrystal of equal size to this pore by *Sveinsson et al.* [8], they found that the crystal



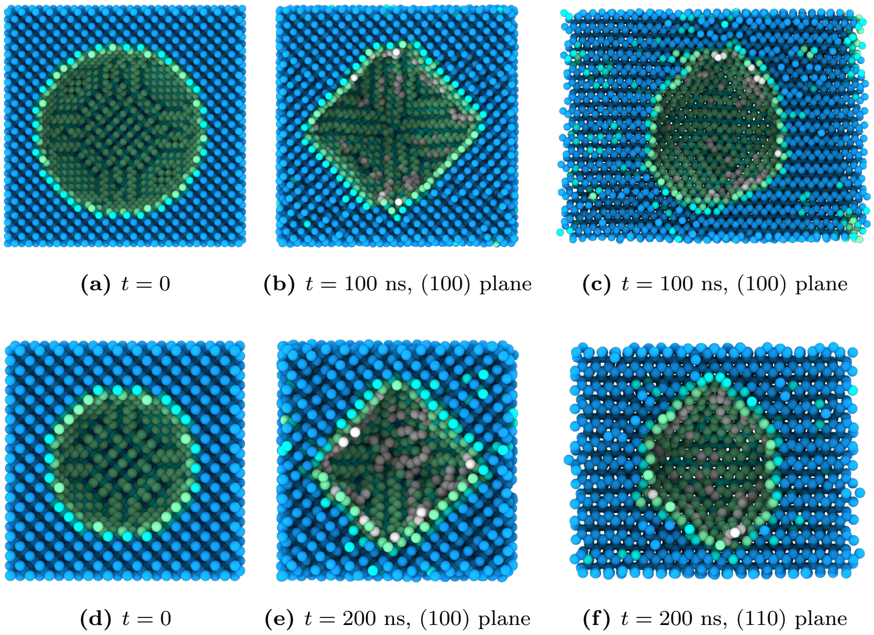
(a) $t = 0$ **(b)** t = 100 ns, (100) plane **(c)** t = 100 ns, (110) plane

(d) $t = 0$ **(e)** t = 100 ns, (100) plane **(f)** t = 100 ns, (110) plane

Figure 7.2: Cross-sectional snapshots of the initial and final atom configurations of the two larger systems. The spherical pore undergoes a faceting process, and is transformed into an "inverse" nanocrystal with 12 (110) facets in the course of the 100 ns simulation. The side lengths of the systems are (top row) 9.16 nm and 11.33 nm (bottom row), the radii of the spheres at $t = 0$ is 4.0 nm. The 11 nm system is large enough for a pore of this size to grow in isolation, while the size of the smaller system is inadequate, as the pore fuses with its periodic images in the (100) planes. The colors indicate the crystal structure of the atoms, cyan means cubic and orange means hexagonal.

was partly and fully faceted after approximately 140 ns and 400 ns. Here, the facets seem to be fully grown after only 100 ns, though this is based on visual inspection only. It may be that the inverse crystal has a growth process which is more rapid than the actual nanocrystal itself, but this speed-up may also be related to the fact that the pressure is not controlled during these simulations.

For the purposes of this thesis, 100 ns was chosen as an adequate simulation period, as the shape of the pore in this system and theirs appear to be quite similar at this point. The hexagonal diamond structures of some atoms visible in Figure 7.2c of the same system, is possibly due to the pressure not being controlled during the simulation, but this looks to have little effect on the final crystal shape of the pore. In the bottom row of the same figure (Fig. 7.2), the pore fuses with itself thought the periodic boundaries, which perhaps is an obvious result when looking at the initial configuration, as the distance between the pore wall and the boundaries in the (100) axes are barely two unit cells wide. In Figure 7.3 the size of the systems were reduced rather drastically, but their final shapes are still similar to those observed in the large ones. It is curious that the smallest of the systems required additional simulation time before its pore grew to a distinct crystal - This may be related to the possibility that the size of the initial spherical pore is bordering on the limit of how small it can be and still transform into the recognizable geometry



(a) $t = 0$ **(b)** $t = 100$ ns, (100) plane **(c)** $t = 100$ ns, (100) plane

(d) $t = 0$ **(e)** $t = 200$ ns, (100) plane **(f)** $t = 200$ ns, (110) plane

Figure 7.3: Cross-sectional snapshots of the initial and final atom configurations of two smaller systems with side lengths (top row) 7.41 nm and (bottom row) 5.23 nm, with pore radii 2.5 nm and 1.5 nm respectively. The smaller system (bottom row) required and additional 100 ns of simulation before the facets were visible and unambiguous.

seen in the larger systems. Due to the relatively low expense of simulating the smallest of the four systems, it was deemed the most practical to study further, as repeated simulations of the larger ones would require significantly more computational resources, as seen from the listing of hardware performance in Table 7.2.

7.2 Force of Crystallization

During the energy minimization of the spherical pore in the previous section, we observed that a faceting process took place, giving the pore an inverse faceted shape. There is effectively no change in the chemical potential in a pure system at constant temperature and pressure, indicating that this type of system may be a candidate for studying the *force of crystallization* where the only effect during the crystal "growth" is the reorganization of surface atoms. In this section simulations of this system are performed at constant temperature and volume, where the initial configuration is a spherically shaped pore, and the global pressure and potential energy during the relaxation towards its equilibrium faceted shape is measured. These measurements are then compared to the the *Kohlrausch-Williams-Watts* stretched exponential [18].

7.2.1 Simulation Setup

The initialization of the system follows the same procedure as in Section 7.1. Following this, the simulations are run according to this general procedure:

1. The temperature of the system is set to the desired value (see below) and relaxed with the pressure to 0 during 30 ps. A damping coefficient in the thermostat of 0.5 ps is used. The barostat is set to have a damping coefficient of 0.5 ps together with a drag coefficient of 2 ps, using the `pdrag` keyword together with the `fix npt` command. As will be shown in the results below, the faceting process begins immediately, the damping parameters for the barostat were therefore chosen[2] based on their ability to relax the pressure rapidly with limited oscillations.
2. Once the pressure and temperature are thermalized, I sample the lengths of the simulation box at every time step for an additional 30 ps. This is done in order to get a good estimate for the volume of this system at zero pressure, such that the pressure induced by the faceting process can be accurately measured.
3. The dimensions of the simulation box are fixed to the measured average from the previous step, and the simulation is run for 30 ns at constant volume and temperature.

The above steps are visualized in Figure 7.4. These simulations are performed with temperatures $2100, 2150, 2200, 2250$ and 2300 K, running 20 simulations with different initial velocities in each instance (100 simulations in total). This is necessary because the pressure fluctuations in NVT simulations of this size are very large and need to be averaged (see Figure 10.1 in the Appendix). The pressure, potential energy and mean squared displacement of the atoms are sampled every 10 time steps (0.02 ps), and their values are averaged over each of the 20 runs. The atom positions are also sampled every 100 ps, which are used to measure the surface area of the pore.

7.2.2 Results

Figure 7.5 shows the system pressure, pore surface area, potential energy (per particle) and diffusion coefficients during the faceting process. The measurements are the result of averaging over 20 full simulations at 2200 K. Subfigures 7.5ab, shows that there is a rapid build-up of pressure accompanied with the transformation of the pore shape during the first 5 ns. The increase in pressure decays quickly and looks to plateau at around $P \approx 30\,\text{MPa}$ towards the end of the simulation. In addition to averaging over 20 simulations, the pressure seen in Subfig. 7.5a is time-averaged with a window of 0.5 ps due to high fluctuations during the course of the simulation. Similarly, Subfigs. 7.5cd in the same figure show that the surface atoms on the pore walls reorganize and the system moves towards a more energy favourable configuration. The potential energy and diffusion seem highly correlated, if not proportional, and the same rapid decay of the faceting process is apparent in all the plots in this figure. The diffusion constant in Subfig. 7.5d is calculated at 10 ps intervals from the relation [66, p. 297]

$$D(t) = \frac{\langle r^2 \rangle}{2d}, \tag{7.1}$$

[2] a search over several damping and drag coefficients was conducted to determine the optimal values

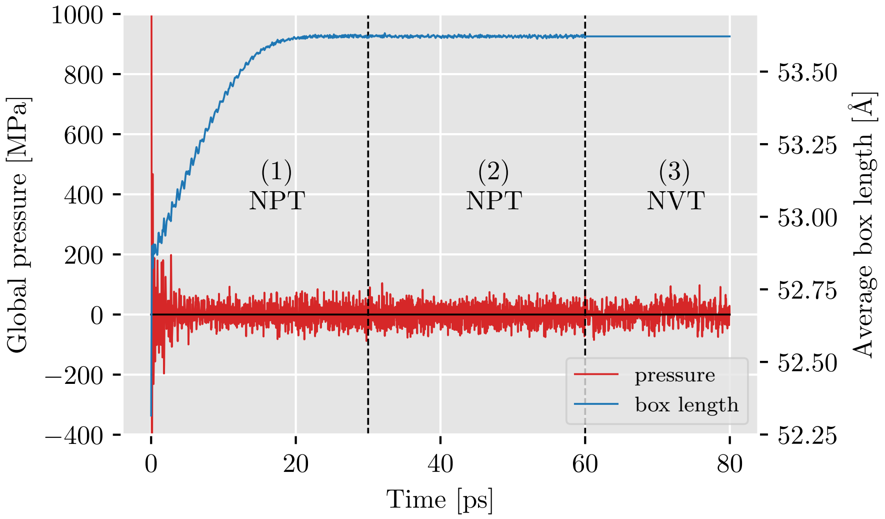


Figure 7.4: Pressure and average box length as a function of time during different stages of the thermalization process when measuring the force of crystallization of the faceted inclusion. Stage 1), indicated in the plot, is the initial thermalization of the pressure by expansion of the simulation box. When the box lengths have reached a constant value, Stage 2) begins, in which the box lengths are sampled at every time step for 30 ps (15000 time steps). Stage 3) marks the beginning of the main run of the simulation, where the box lengths are set to the calculated average from Stage 2), and the time-development of the pressure is sampled.

where $d = 2$ since the diffusing atoms are (roughly) restricted to a surface, and $\langle r^2 \rangle$ is the mean squared displacement of all atoms in the system in the given interval. The surface energy density of the faceted pore is also found to be

$$\gamma = \frac{U_1 - U_0}{A} = 1.848(2)\ \mathrm{J/m^2}. \tag{7.2}$$

Here, U_0 and U_1 is the potential energy of the system averaged over the first and last 10 ps of the simulation respectively, and A is the pore surface area averaged over the last 2.5 ns. The surface area is measured using the atom positions in Ovito's `ConstructSurfaceModifier` [67] with a probe radius of 2.8 Å.

Figure 7.6 shows the potential energy during faceting of the spherical pore for all temperatures, averaged over 20 full simulations of each temperature. Other than the offset in potential energy from thermal effects, the measured values are immediately hard to distinguish, apart from the 2300 K system, which has a bump at around 10 ns. Further, the measured values are fitted to the stretched exponential

$$U(t) = U_0 + A \exp\left[-\left(\frac{t}{\tau}\right)^{\beta}\right], \tag{7.3}$$

introduced in Section 2.3, using `scipy.optimize.curve_fit`[3]. The fitted curves seem to be in good agreement with the measured values, except when t is very close to zero. In this domain,

[3] https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.curve_fit.html

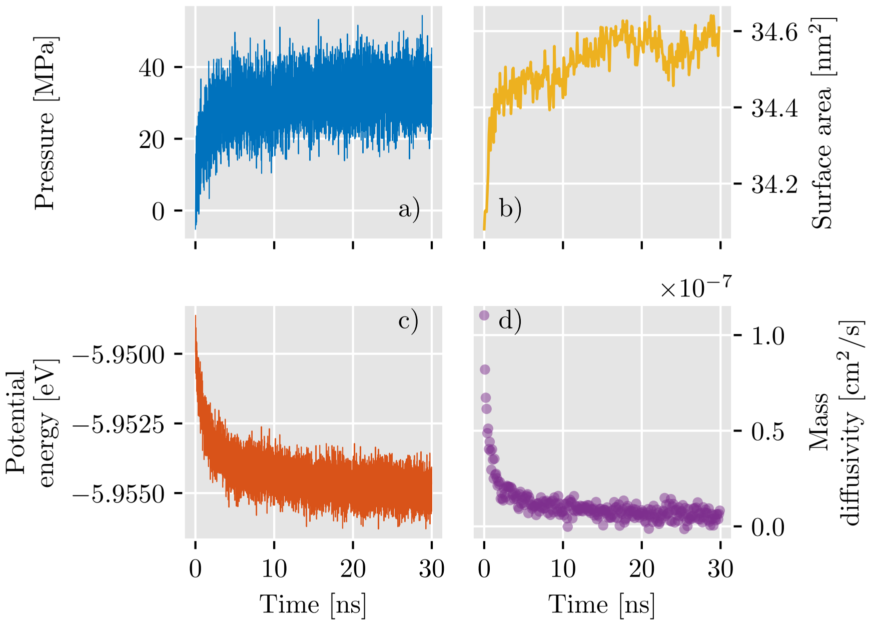


Figure 7.5: System pressure (a), pore surface area (b), potential energy per particle (c) and mass diffusivity (d) during the faceting process of the spherical inclusion at 2200 K. As atoms on the pore surface reorganize, the system moves towards a more energy favourable configuration, indicated by (c) and (d). Together with this reorganization, the shape of the pore is transformed, giving rise to a build-up of pressure in the system, as can be seen in (a) and (b).

the predicted values are clearly higher than the measured ones. The same plot also shows curves which are fitted to the first 10 ns of the simulation, in which case the predictions do deviate slightly from the measured values for $t > 10$ ns, though for temperatures 2100 and 2250 K the two fits are very similar. The exponential factor β is determined in each fitting procedure and found to be

$$\overline{\beta}_{\text{full}} = 0.39(6) \quad \text{and} \quad \overline{\beta}_{\text{partial}} = 0.58(9), \tag{7.4}$$

where $\overline{\beta}$ is the mean of the measured value across all temperatures. This is consistent with the theoretical prediction of $\beta = d/(2+d) = 1/2$ for surface diffusion [33].

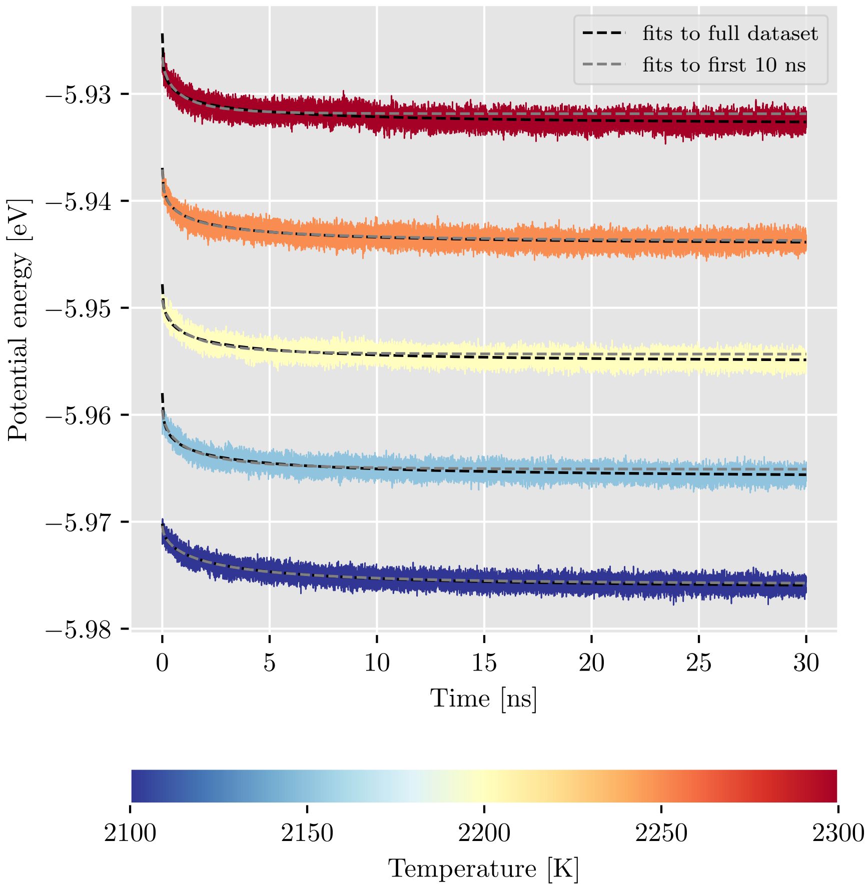


Figure 7.6: Potential energy per particle during faceting of the spherical pore for different temperatures. Each curve is the averaged value of 20 full simulations. The measurements are fitted to Equation 7.3 in order to determine the time-stretching exponent β.

Setting $\beta = 1/2$ and performing a new curve fit to all temperatures gives a mean value of the pre-exponential factor $\overline{A} = 0.0058(1)$ eV, which is assumed to be independent of temperature. Figure 7.7a shows the final curve fits to Equation 7.3 with only the waiting time τ and offset U_0 as free parameters, where $\beta = 1/2$ and $A = 0.0058$ eV are fixed. The fitted curves agree well with the measured values in the whole domain, and do not suffer the slight mismatch at $t \approx 0$ which was observed in Figure 7.6. The measured waiting times τ_i are indicated in the Arrhenius plot in same figure (Fig. 7.7b), where we see that it decreases inversely with the temperature, although the measured values are somewhat noisy. A linear fit to this plot gives the energy barrier of the faceting process of this system, here estimated to be

$$E_a = 1.11(30) \text{ eV}. \tag{7.5}$$

Finally, Figure 7.7c shows the potential energy of the five systems scaled by the pre-exponential factor A and as a function of the dimensionless time t/τ, collapsing the curves into a single line.

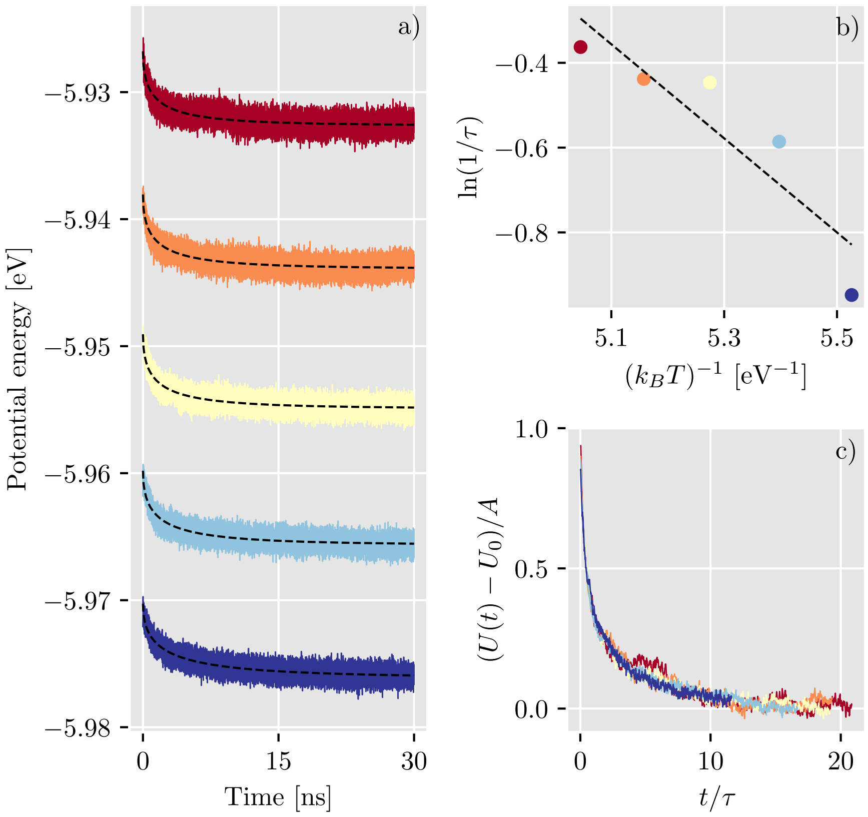


Figure 7.7: a) Potential energy per particle during faceting of the spherical pore for different temperatures. Measured values are averaged over 20 full simulations for each temperature, and fitted to the stretched exponential (Eq. 7.3) with fixed parameters $A = 0.0058$ eV and $\beta = 1/2$. Sub-Figure b) shows a linear fit to the Arrhenius plot, which determines the energy barrier for the faceting process, $E_a = 1.11(30)$ eV. In Sub-Figure c) the normalized and time-averaged potential energy for all temperatures is plotted as a function of t/τ, collapsing into a single curve. Temperatures are indicated by the same colorbar as in Figure 7.6.

For visibility, a running mean is applied in time of the measured values with a window of 0.1 ns. The time averaging reveals slight oscillations in the curves of the systems with $T > 2100$ K towards the second half of the simulation, which becomes more evident in higher temperature systems. The bump in the $T = 2300$ K system at $t/\tau \approx 5$ is especially visible in this plot.

Figure 7.8a shows the global pressure of the system at the different temperatures. Here, simple modification of the stretched exponential is fitted to the measured values,

$$P(t) = P_0 \left\{ 1 - \exp\left[-\left(\frac{t}{\tau} \right)^{\beta} \right] \right\}, \tag{7.6}$$

with $\beta = 1/2$. In order to more easily determine the quality of the fit, the plotted pressure is time-averaged with intervals of 2 ps, and the fit seems to corresponds well with the averaged

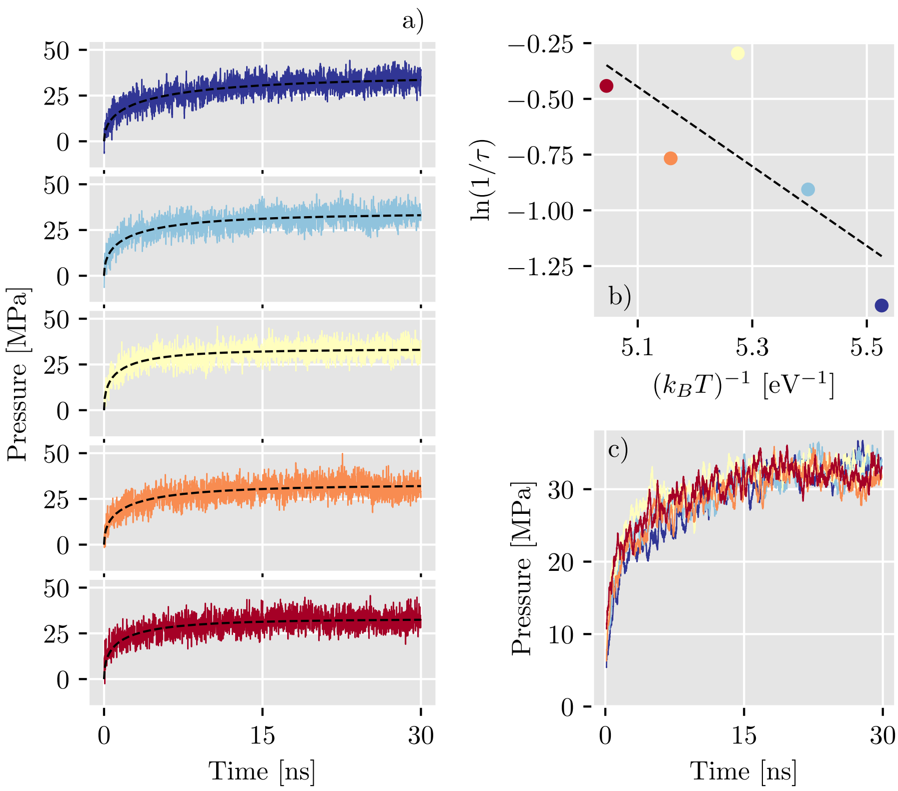


Figure 7.8: a) Global pressure during the faceting of the spherical pore for different temperatures, together with their fits to equation 7.6 (indicated by the dashed lines) with fixed $\beta = 1/2$. Subfigure b) shows the corresponding Arrhenius plot, where a linear fit results in an activation energy $E_a = 1.78(86)$ eV. In Subfigure c) the pressure is again plotted (with time-averaging). Temperatures are indicated by the same colorbar as in Figure 7.6.

values. The resulting parameters are listed in Table 7.3. Figure 7.8c again shows the time-averaged pressure with a larger averaging window of 25 ps, where we immediately see that the temperature dependence of the pressure as a function of time is hard to determine, as the curves do overlap quite heavily. This also becomes clear from Figure 7.8b, showing the corresponding Arrhenius plot with a rather unsystematic behaviour of the reaction rates. The linear fit to the Arrhenius plot gives an activation energy

$$E_a = 1.78(86) \text{ eV}, \tag{7.7}$$

which is consistent with the energy barrier measured from the potential energy.

From the above results, there seems to be strong correlations between both the pressure and pore surface area, and the potential energy and atom diffusion. This indicates that the mechanism of the faceting process is the reorganization of surface atoms. The measurements of the mass diffusivity is is done using the mean squared displacement of *all* atoms in the system, so the actual values are a little arbitrary. Still, its functional form suggests that surface diffusion is the

Table 7.3: Parameters from fitting Equation 7.6 to the measured pressure in Figure 7.8.

T [K]	P_0 [MPa]	τ [ns]
2100	35.99(8)	4.17(5)
2150	34.12(5)	2.48(3)
2200	33.27(4)	1.34(1)
2250	32.78(5)	2.15(3)
2300	32.84(4)	1.56(2)

mechanism of the relaxation process when comparing it to the potential energy of the system in Figure 7.5.

The sampling of the pore surface area in the simulations are done at intervals of 100 ps, which is quite large. Still, the surface energy density is in good agreement with the reference value for 3C-SiC of 1.830 J/m^2, found by DFT-methods [68].

The fitting procedure was also performed on the pressure measurements, where I used a simple modification of the stretched exponential. The pressure in small systems such as this one, is subject to large fluctuations, and the resulting fitted values of the relaxation times are less definitive than the ones found from fitting the potential energy. The two measurements of the energy barrier are in agreement, but I regard the fit to the potential energy as more reliable due to lesser amounts of noise in the data and a smaller uncertainty in the resulting value.

In general, the pressure build-up during the simulations are in the order of ~ 30 MPa, which is comparable to the pressure induced by salt crystallization, measured by *Desarnaud et al.* [16] to be in the order of 150 MPa. In my system, it is reasonable to imagine that the final pressure will in some way be dependent on the ratio of the pore and system sizes, though this has not been explored further. It does also seem to have a slight dependence on the temperature, when considering the fitted values of the pre-exponential factor P_0 in Table 7.3.

Determining the time-stretching factor β was done by performing full and partial fits to the potential energy. These measurements were both in agreement with the analytical value for diffusing atoms in a medium of random traps [33], and is interpreted as an effect of the growing pore which gets an increasingly lower energy state, effectively trapping atoms in place as it relaxes. This relaxation behaviour is also found in glassy systems [69, p. 318], suggesting that there are similarities in the behaviour such systems and in the faceting of this pore. Although not examined in this system directly, the growing nanocrystal in *Sveinsson et al.* [8] was observed to have atom layers which successively seized in participating in the facet growth after increasingly long times. This is a similar hierarchical behaviour to the premise of the stretched exponential derivation by *Palmer et al.* [34], in which relaxation also was divided into a sequence of processes with increasing relaxation times. The faceted pore does of course have the inverse curvature to the system of *Sveinsson et al.* [8], but their similarities are great enough for these results to suggest that the faceting process is related to a mechanism in which increasingly many atoms need to reorganize in order for the system to reach a lower energy state.

7.3 Mechanical Characteristics of the Faceted Inclusion

In the previous sections we saw that the shape of the initially spheric inclusion in the SiC system eventually transformed to the shape of a dodecahedral nanocrystal. In this section I perform further simulations of this system, where the initial geometry is the faceted inclusion. A literature search on this subject revealed little on this specific system, and so the goal of these simulations is to observe its basic mechanical behaviour when subjecting it to a compressive force. Repeated simulations of this system are performed varying temperatures and rates of compression while measuring its global stress.

7.3.1 Simulation Setup

The final atom configuration of the system of size (5.32 nm)^3 in the previous section (Sec. 7.1) is used as the initial configuration of the simulations in this section. I thermalize the temperature to the desired value and the system pressure to 0 for 200 ps, using a Nosè-Hoover thermo/barostat with damping coefficients of 1 and 10 ps respectively. Uniaxial stress is applied to the system by deforming the z-length of the system to 85% of its original length during the simulation. The x and y lengths are allowed to expand, such that the corresponding stress components are kept at 0. In total 20 simulations are performed with temperatures $T \in [2000, 2100, 2200, 2300, 2400]$ K and times $t \in [18.2, 25, 40, 100]$ ns, corresponding to strain rates $\dot{\varepsilon} \in [8.24, 6.0, 3.75, 1.5] \times 10^{-3}$ ns^{-1}.

7.3.2 Results

Figure 7.9 shows the stress-strain curves from two sets of simulations of varying temperature (a) and strain rate (b), as well as a comparison to how the bulk material behaves (c). The temperature has an expected and apparent effect on the shape of the curves, where we see that both the stiffness and strength of the system is reduced with increasing temperatures. The stress-strain relationship is linear initially, with a hyperelastic softening leading up to the yield point. The two initial yield points look to have a structured behaviour, while the proceeding ones are increasingly less predictable in terms of both when they occur and the resulting residual stress. Varying the strain rate, as seen at the bottom of Figure 7.9, suggests that it is somewhat proportional to the system strength, but its stiffness remains the same.

The initial release in stress post-yield is associated with the nucleation of a dislocation plane in the (111)-direction of the system, as indicated by Figure 7.10, showing snapshots of the system at different stages of deformation. The consecutive drops each coincide with a step of dislocation creep along the dislocations, shown in Figure 10.2 in the Appendix.

Figure 7.11 shows a data-collapse of the stress-strain curves. These are the result of attempting to find a scaling function for these curves which makes them invariant to the temperature. Here, it is assumed that the yield rate $1/\tau^y$, measured as the reciprocal of the time at the yield point, should be proportional to the Arrhenius factor, similar to a sliding grain boundary model by

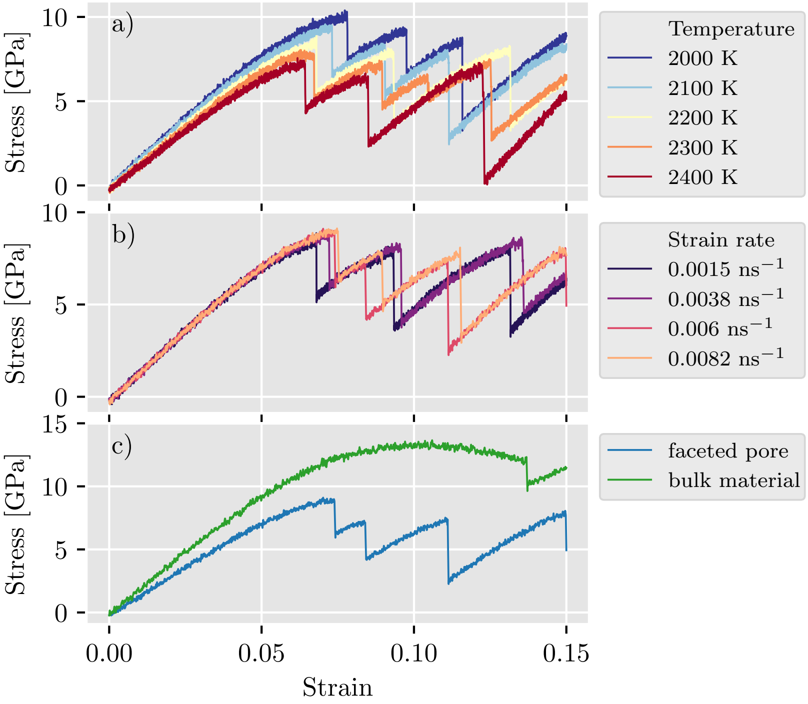


Figure 7.9: Stress-strain curves during uniaxial compression of the faceted pore. The effects of temperature are shown in a), showing that increasing the temperature reduces the stiffness and yield strength of the system. In b) the stress-strain relationship is shown for varying strain rates, where the stiffness remains the same, but the strength increases slightly with higher strain rates. Subfigure c) shows the difference between the system with and without a faceted pore. There are noticeable differences in the behaviour of the two. In addition to being stiffer, the bulk material goes through a distinct phase of hyperelastic softening, before yielding at approx. double the strain.

Borodin et al. [70]:

$$1/\tau^y \sim \exp\left(\frac{-E_a}{k_b T}\right). \tag{7.8}$$

Here, E_a is an activation energy and k_b Boltzmann's constant. The activation energy is determined by taking the logarithm of Equation 7.8, and performing a linear fit, resulting in $E_a = 0.26(1)$ eV. As can be seen in the figure (Fig. 7.11), scaling the x-axis by this exponential causes the (scaled) yield strain to line up irrespective of the temperature. Similarly, the system stress is scaled by a factor T^n, with $n = -2$, which is inspired by the modified Arrhenius equation where the temperature is made explicit in the pre-exponential factor [71].

Using `scipy.signal.find_peaks`[4] the yield stress and strain of the three initial peaks of the stress-strain curves are determined, for all the simulation configurations, shown in Figure 7.12.

[4] https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.find_peaks.html

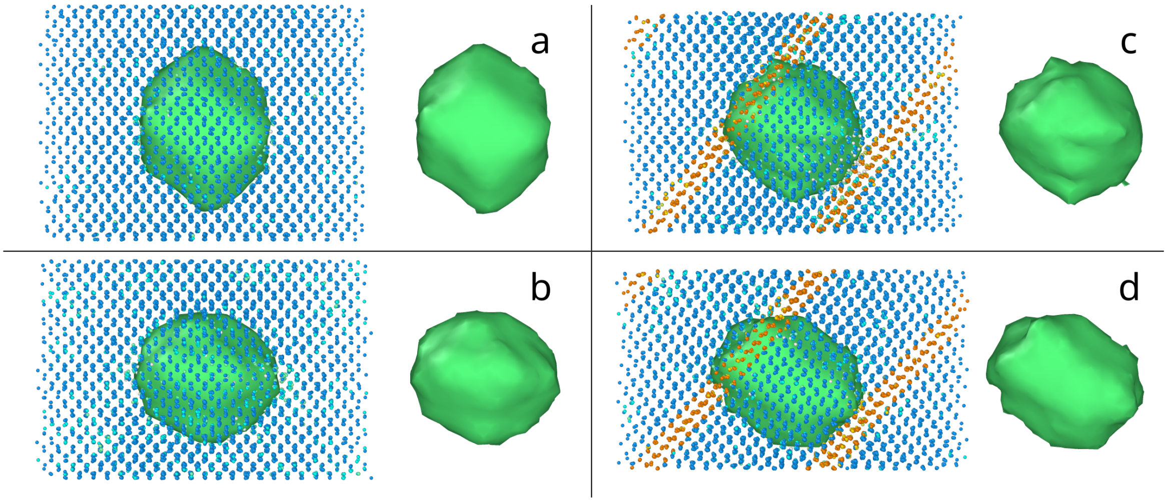


Figure 7.10: Renderings of the faceted pore under uniaxial stress at different stages of deformation, a) initial configuration, b) moment before yield, c) moment after yield, d) final configuration. The depictions are with and without atoms for visibility. The green mesh outlines the pore surface. As the system is compressed and yields, the atoms form diagonal dislocation planes, indicated in orange. The shape of the pore is also visibly changed, and its facets seen in the initial configuration are substantially less distinctive at the end of the simulation. Constructed using Ovito [54].

The initial yield point behaves as expected, with a clear dependence on the temperature. At the second point, the yield stress is slightly lower, though still ordered, and finally the third point of yield which exhibits a more chaotic behaviour. The yield stress and strain for the first point of yield is listed in Table 7.4. Included there is also the measured Young's moduli for the different temperatures, found by linear fit to the stress-strain curves in the domain $\varepsilon \in [0, 0.01]$ using `numpy.polyfit`[5]. These are also shown in Figure 7.13, showing a linear decrease with temperature of both the yield stress and Young's modulus.

Table 7.4: Measured values of the Young's moduli, yield stress and yield strain in systems of varying temperature. The listed values are the result of averaging the measurements from the simulations with different strain rate.

T [K]	E [GPa]	σ^y [GPa]	ε^y
2000	170.7(5)	10.4(1)	0.081(2)
2100	161.1(5)	9.7(1)	0.077(2)
2200	152.8(5)	9.0(2)	0.072(3)
2300	146.3(6)	8.2(2)	0.068(2)
2400	139.8(6)	7.5(1)	0.065(1)

During the compression of the system, the pore shape is transformed and its volume reduced. This is shown in Figure 7.14, where the pore volume is plotted against the system strain. There is initially a linear decrease in volume, followed by an acceleration as the system stress reaches the

[5] https://numpy.org/doc/stable/reference/generated/numpy.polyfit.html

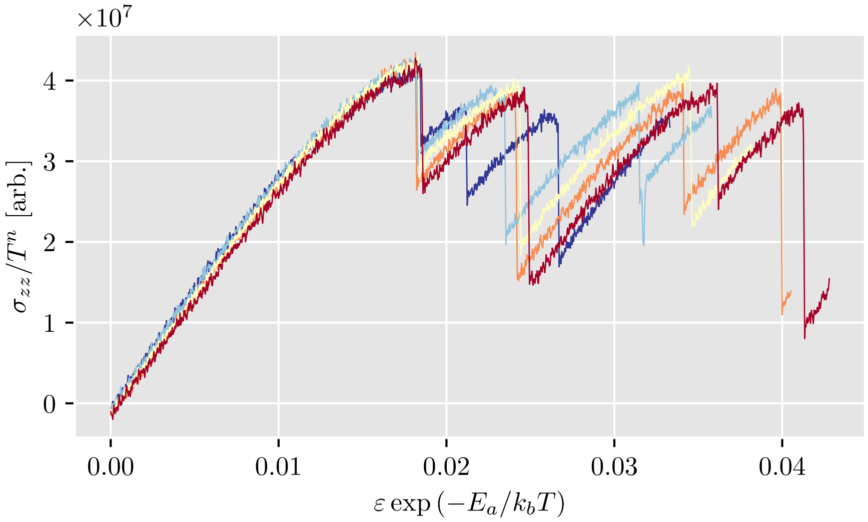


Figure 7.11: Stress versus strain scaled by the Arrhenius factor and the temperature, collapsing the measurements on to a single curve up to the yield point. The colors indicate different temperatures, and follow the same legend as in Figure 7.9. These simulations are run with strain rate $\dot{\varepsilon} = 0.00375$ ns^{-1}.

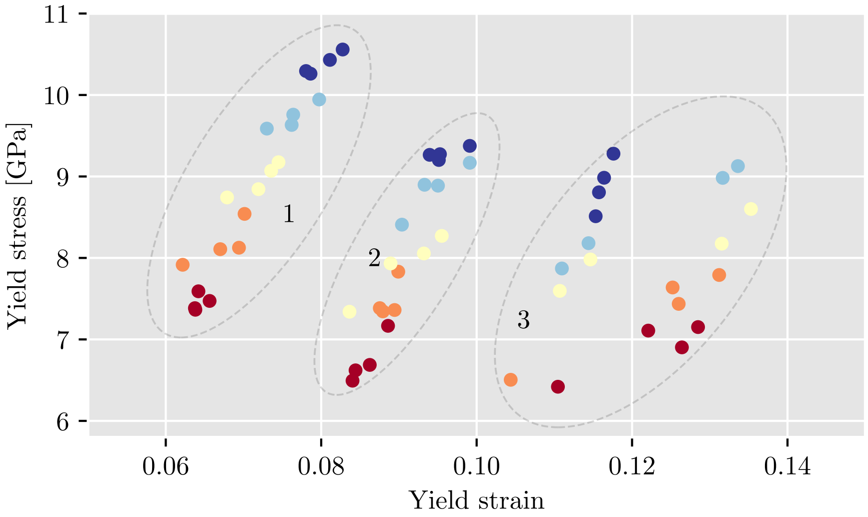


Figure 7.12: Yield stress and strain for the three initial yield points, indicated by the dashed ellipses. The measurements are from simulations of all temperatures and strain rates. System temperature is indicated by the same colors as in Figure 7.9. The yield stress and strain of the first two yield points seem to be quite systematic, while the third has a more complex behaviour.

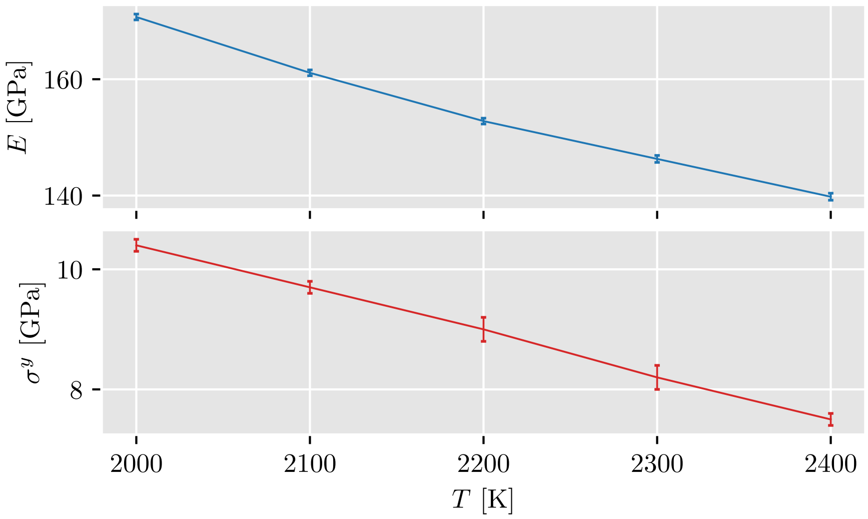


Figure 7.13: Young's modulus (top) and yield stress (bot) as a function of temperature.

critical point of yielding. This general trend is also demonstrated to the right in the same figure, which shows the piecewise change in volume and stress at ten intervals of the stress/volume-strain relationship from each of the simulations.

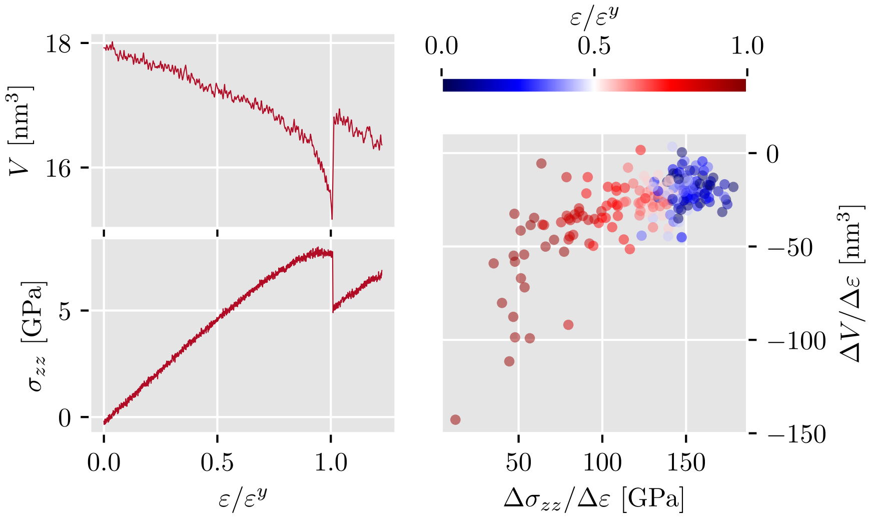


Figure 7.14: Pore volume and stress during compression of the system (left). The volume decreases linearly during the initial stages of deformation, and as the system approaches its yield strength, the shrinkage accelerates. The subfigure to the right shows the rates of change of the pore volume and stress in different intervals leading up to the point of yield (all system configurations included).

Figure 7.15 shows the time evolution of the potential energy (per particle) of the system, before and during compression. It remains constant before the simulation box is deformed, and declines initially when compression is applied, after which it begins to rise up until the point of yield. The initial decline in potential energy is a little unexpected, as we might anticipate that it increases as it gets compressed. It is not apparent whether this effect is general for the SiC-3C crystalline structure itself, or if it is specific to the porosity of this system.

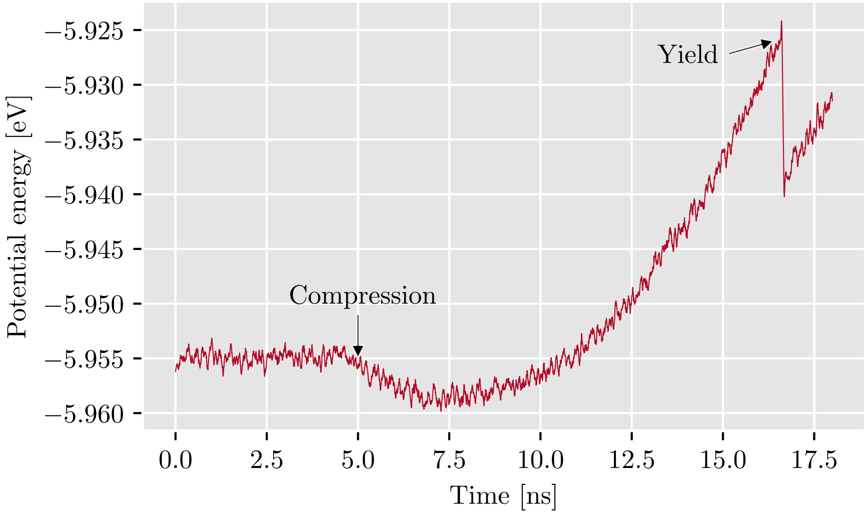


Figure 7.15: Potential energy (per particle) of the system before and during compression. Before compression is applied to the system, the potential energy remains constant. Straining it causes the potential energy to decline in a convex manner, before it drops abruptly at the point of yield.

The stress-strain curves of the system (Fig. 7.9) reveals that the system strength and stiffness decrease with increased temperatures. This behaviour is not unexpected, and has also been observed in experiments on silicon carbide materials of varying porosity by *Milman et al.* [72]. The porosity-dependence of the strength in the system in this thesis has not been studied, but the observation that the bulk material is stronger than with the pore present is also consistent with general relations of porosity and the uniaxial strength of materials [73].

The scaling factors used in Figure 7.11 succeed in collapsing the stress-strain curves, though only up to the yield point. Scaling the stress by the temperature collapses the elastic regime of the stress values on a single line. Similarly, the Arrhenius factor accounts for the temperature dependence of the yield strain, although I have no clear interpretation of the value found for the activation energy. Interestingly, it does correspond to values provided by *Borodin et al.* [70], for stress relaxation in grain boundaries of metals at room temperature. Following the initial yield point, the consecutive build-ups and releases in stress have a more chaotic behaviour, where the yield points and the resulting residual stress may be increasingly sensitive to thermal fluctuations, and are not easily explained in terms of the system temperature by itself. These scaling relations are largely experimental, without proper justification with respect to the underlying mechanisms of the observed behaviour.

The general yielding mechanism in this system is mediated by dislocations which are observed to nucleate at the interface of the pore space and bulk material, indicated in Figure 7.10. This is not entirely unexpected, as the geometric irregularities along this interface may be regarded as crack tips with asymptotic stress values. This has not been measured directly in this system, but measurements of the local stress field in cracks in silica systems by *Rimsza et al.* [22] show levels which are orders of magnitude higher near the cracks than elsewhere in the system.

The pore volume as a function of strain in Figure 7.14 demonstrates some interesting dynamics of this system. The initial linear decline of the volume is not unexpected, due to the fact that the system is being compressed, but the acceleration of the reduction towards the yield point is intriguing. It immediately looks as though the external stress is large enough for the pore to grow inwards for a short period of time, before it becomes large enough for the dislocations to nucleate. This acceleration in volume reduction also looks to be correlated with the hyperelastic softening effect observed in the same figure (Fig. 7.14). As the stress is released, the volume suddenly increases, and most interestingly in this figure, back to the same value it had before this acceleration began. This jump in volume may also suggest that there is some elastic energy stored in the pore surface, which gets released as the system yields.

7.4 Creep

Motivated by the interesting dynamics observed near the yield point of the faceted inclusion in the previous section, specifically the behaviour of the pore volume, a series of creep-type simulations of this system under constant stress is performed. I measure its time to failure under different (constant) stressed conditions, and examine how this depends on both the externally applied stress and the system temperature.

7.4.1 Implementation

The simulations of the faceted inclusion in this section are split into two parts. One for generating a number of LAMMPS restart files (containing the system state) in which different levels of external stress is applied to the system, and one for performing creep-type simulations at constant pressure, with these states as the starting point. The simulation specifications to generate the initial creep-states follow the same procedure as in 7.3.1, where the simulation box is compressed in the z-axis at a constant rate of $\dot{\varepsilon} = 6 \times 10^{-3}$ ns^{-1}. At $\varepsilon/\varepsilon^y = 0.6$, where ε^y is the point at which the system yields for a given temperature (Table 7.4 in the previous section), restart files of the system state are written at intervals of 50 ps until the system yields. Creep simulations are then run with the generated restart files as the initial conditions. No further compression is applied at this stage, and the simulations are run until the systems yield (or for a maximum for 20 ns). This procedure is performed for temperatures $T \in [2000, 2100, 2200, 2300, 2400]$ K, and a total of 194 initial creep-states are generated and simulated. The breaking time τ is measured as the period from the start of the creep-simulation to the time at which the system yields. The simulation procedure is also illustrated in Figure 7.16.

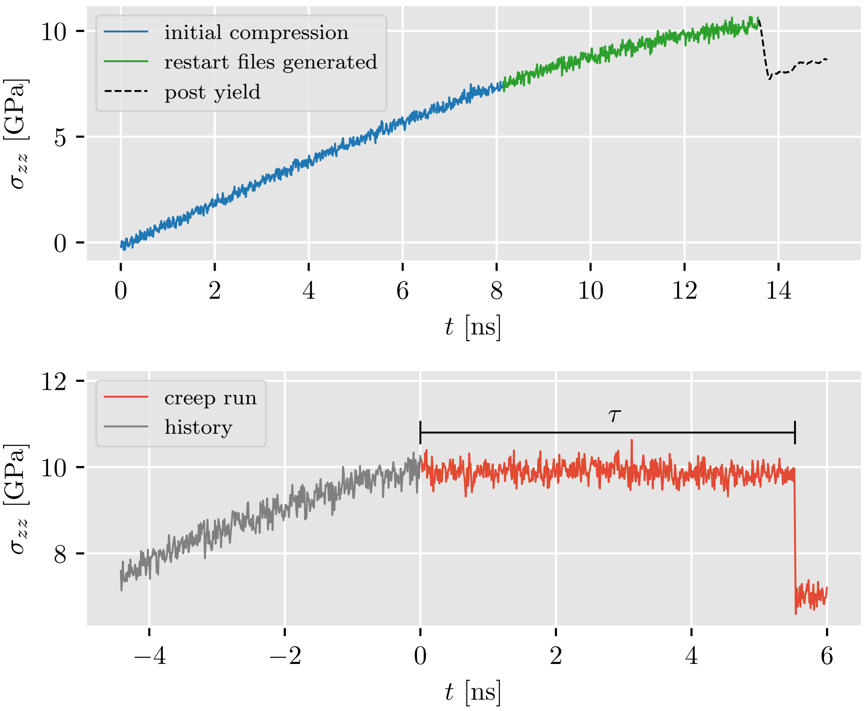


Figure 7.16: Illustrations of the initial setup and implementation of the creep simulations. The top subfigure shows the stress-curves from the simulation which generates restart files to be used in the creep simulations. The restart files are generated every 50 ps along the green curve, as indicated, until the system yields. In the bottom subfigure, the stress is shown from one of the creep simulations. The grey curve is to indicate the stress of the system prior to the simulation, and the stress during it is shown in red. Also shown in the bottom subfigure is how the breaking time τ is measured.

7.4.2 Results

In Figure 7.17 (top), breaking times τ are showed as a function of externally applied stress σ for different temperatures. The breaking times shorten exponentially with increased stress, and behave similarly to a step function at the critical stress value. The curve fit of the BBZ model of time-dependent rupture,

$$\tau = \tau_0 \exp\left(\frac{E_0 - \sigma V}{k_b T}\right), \tag{7.9}$$

introduced in Section 2.2, is shown along side in the plot, where I have used `sklearn.LinearRegression`[6] to find the energy barrier E_0, activation volume V and characteristic breaking time τ_0. The curves are the result of fitting the model to the whole data set, giving one value for each of these

[6] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

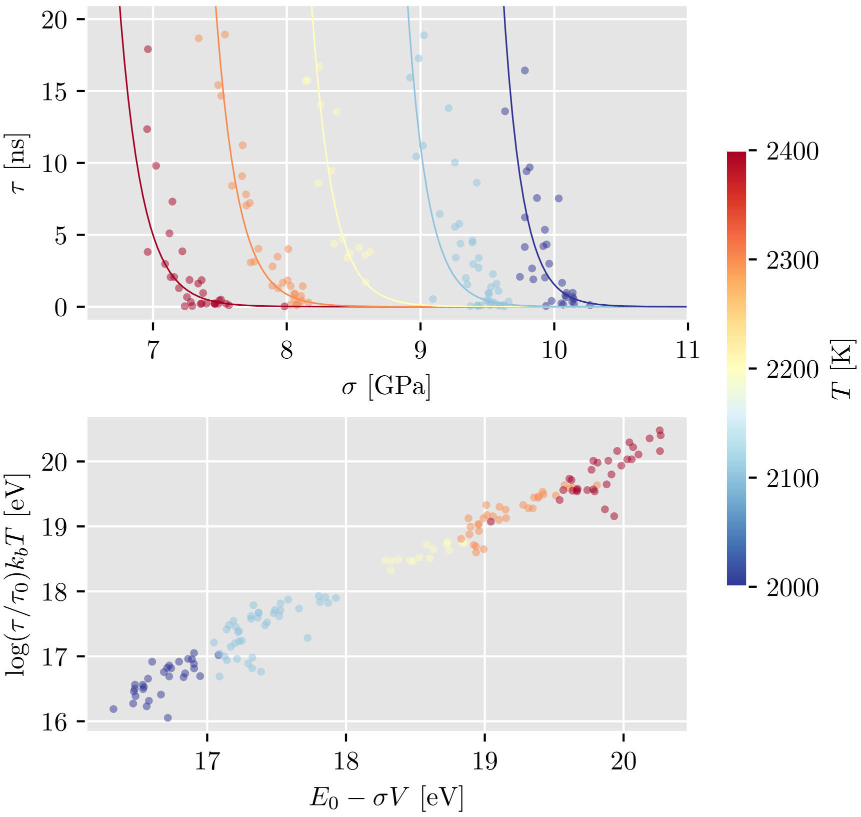


Figure 7.17: (Top) Breaking time τ as a function of externally applied stress σ for different temperatures. The BZZ model for time-dependent rupture is fitted to this data. (Bottom) Data-collapse of the breaking times.

parameters:

$$E_0 = 28.5(19)\text{ eV}, \qquad V = 7.4(5)\text{ nm}^3, \qquad \tau_0 = 1.7 \times 10^{-42}(17)\text{ ns}. \tag{7.10}$$

The corresponding data-collapse is also shown in Figure 7.17 (bottom). The reference value of the pre-exponential factor is in the order of $\tau_0 \approx 10^{-4}$ ns [23]. The model is also fitted separately to the breaking times in systems of individual temperatures, where the pre-exponential factor is fixed to the reference value. The resulting parameters are listed in Table 7.5. Here, the energy barrier decreases slightly as the temperature is increased, while the activation volume is roughly constant.

The breaking times are very sensitive to the applied stress and increase almost step-like when this critical stress is reached, within the range of this narrow time-domain. This has resulted in difficulties in gaining any reliable insights into the behaviour of this system when subjected to a constant external load. Even though the fitted curves and data-collapse in Figure 7.17 immediately look reasonable, the value found for the characteristic breaking time τ_0 is many orders of magnitude below the reference value, and is clearly unphysical. *Zkurkov* [23] treats the energy barrier and activation volume as constants specific to the material, but in order to get

Table 7.5: Fitted parameters to the BBZ model (Eq. 7.9) with $\tau_0 = 10^{-4}$ ns, showing the resulting values for the energy barrier E_0 and activation volume V.

T [K]	E_0 [eV]	V [nm^3]
2000	14.1(23)	7.8(14)
2100	13.0(19)	7.5(13)
2200	8.4(10)	4.7(7)
2300	11.2(10)	7.3(8)
2400	11.6(14)	8.3(12)

reasonable values for the physical parameters, an assumption was made that they are dependent on the temperature as well. In doing so, the resulting values for E_0 (Table 7.5) are reasonably close to the bond energy of Si-C (4.6 eV [74, p. 11]). It is possible, though, that this assumption is valid over a large range of temperatures, as we might expect the range of stresses at which the system yields to get narrower with lower temperatures due to reduced thermal fluctuations. Ultimately, the BBZ model has previously been successful in predicting the breaking times of materials under tensile stress, so it is also possible that it is not well-suited in explaining the behaviour of compressed systems. It is also clear from the above results that the simulations would have to be run for longer than 20 ns to get a good representation of the breaking times across a larger range of stresses.

Chapter 8

Multiple Pore Systems

The previous chapter was dedicated to studying the dynamics of a single pore. The objective of study in this chapter is a much larger system with several faceted inclusions within it. Of interest is how the spatial positions of these pores affect the strength of the system, and here I attempt to answer this question by combining the use of molecular dynamics simulations and machine learning.

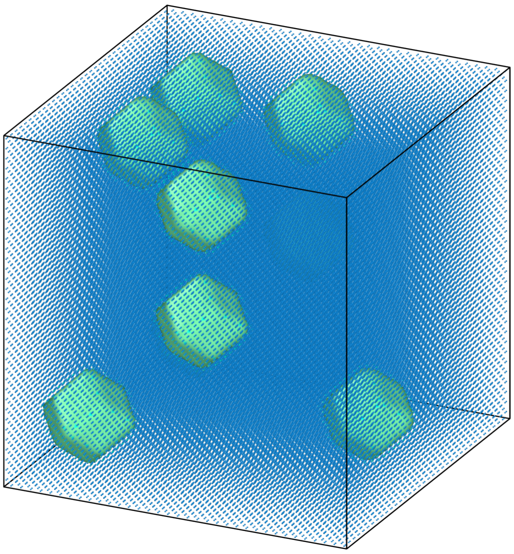


Figure 8.1: Snapshot of the silicon carbide system with faceted inclusions. The pore surfaces are here outlined in green, and the surrounding bulk material of atoms are shown as blue dots. This system is designed such that the possible locations of the pores are on a regular $3 \times 3 \times 3$ grid.

8.1 Predicting Mechanical Properties

In this section I describe my simulations of a porous system consisting of several of the faceted inclusions from Chapter 7. This system is composed of a regular $3 \times 3 \times 3$ grid, where each grid-cell is occupied by either bulk material or a faceted pore located at its center, illustrated in Figure 8.1. I perform repeated simulations of this system with different pore configurations under compressive stress. The goal is to study how the strength behaves as a function of the pore locations, and so the results of these simulations are used as a ground-truth for training a machine learning model capable of approximating the yield strength.

8.1.1 Data Acquisition

Defining the Design Space

I define this system as a $3 \times 3 \times 3$ binary grid G, where each of the 27 cells may or may not be occupied by a pore located at its center,

$$G_{xyz} = \begin{cases} 1, & \text{pore is present} \\ 0, & \text{pore is not present.} \end{cases} \tag{8.1}$$

Denoting the system length as L and the cell-length as $l = L/3$, the real space vector of a pore (center) within the system is given by

$$\boldsymbol{r} = l \cdot (x + \tfrac{1}{2}, y + \tfrac{1}{2}, z + \tfrac{1}{2})^T \tag{8.2}$$

with $x, y, z \in [0, 1, 2]$.

Since the number of possible configurations in this space is rather large ($2^{27} \approx 1.3 \times 10^8$) and the porosity of a system is often a first-degree approximation to its strength [73], I restrict myself to only considering systems with a fixed porosity of *nine* pores[1]. I also wish to avoid performing costly simulations of two or more configurations which *should* yield the same result, and I therefore choose not to consider those which are periodic and symmetric images of another. These can be identified by recognizing that the grid may be rotated 90, 180 and 270 degrees about the axis of applied stress and 180 degrees about the remaining two without breaking symmetry. Furthermore, the cell-coordinate of the pores within the grid may be shifted one step in each direction of the spatial dimensions, totalling 26 images of the original which are in fact equivalent in a periodic system.

I implement this procedure by first assigning an id to the cell-coordinates $\boldsymbol{g} = (x, y, z)^T$, giving a list of ids $\boldsymbol{c} = [1, 2, 3, \ldots 27]$. I define the mapping from $\boldsymbol{g}$ to $\boldsymbol{c}$ to be

$$c_i = x + 3y + 9z + 1. \tag{8.3}$$

The grid and corresponding cell ids are illustrated in Figure 8.3. A configuration with a given number of pores k, labeled $\boldsymbol{c}^k$, is then be represented by any k unique elements from $\boldsymbol{c}$. The number of possible ways of ordering $\boldsymbol{c}^k$ is given by the Binomial coefficient $\frac{n!}{k!(n-k)!} \approx 4.7 \times 10^6$

[1] the description of this implementation is valid for any choice of porosity, nonetheless

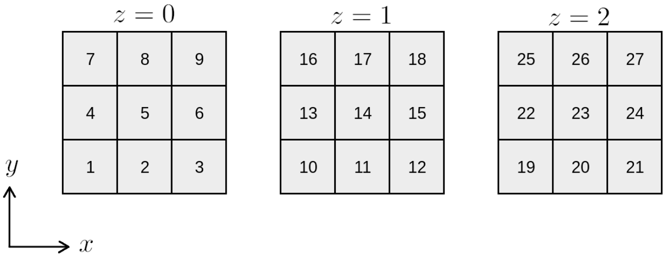


Figure 8.2: Illustration of the pore grid and corresponding grid-cell ids. The physical system is split into 27 equally sized grid-cells, the centers of which are the possible locations for the pores.

with $n = 27$ and $k = 9$. The set of possible orderings, hereby labeled $\mathcal{A}$, is generated by applying Python's `itertools.combinations`[2] to the full id list $\boldsymbol{c}$. For each of the rotational transformations of the grid, I define a mapping $c_i \to c_j$, where c_j is the id corresponding to the location in the grid were it to be rotated. The mappings under rotation about the z-axis is implemented in the following way:

```python
from numpy import array, arange, ravel, flip, rot90

mapping = {90: {}, 180: {}, 270:{}}

for z in range(3):
    #cell IDs in current (x, y)-plane
    xy_plane_IDs = flip(arange((z+1)*9, z*9, -1).reshape(3, 3), axis=1)
    for k in range(1, 4):
        #rotate IDs k*90 degrees
        rotated_IDs = rot90(xy_plane_IDs, k=k)
        for ID, rot in zip(ravel(xy_plane_IDs), ravel(rotated_IDs)):
            mapping[k*90][ID] = rot
```

Similarly, the mappings of 90 degree rotations about the x and y axes are defined as

```python
# 180 degree rotations about y-axis
dx_z0 = {1:21, 2:20, 3:19, 4:24, 5:23, 6:22, 7:27, 8:26, 9:25}
dx_z1 = {10:12, 11:11, 12:10, 13:15, 14:14, 15:13, 16:18, 17:17, 18:16}
dx_z2 = {val:key for key, val in dx_z0.items()} #dx_z2 = inverse of dx_z0

# 180 degree rotations about x-axis
dy_z0 = {1:25, 2:26, 3:27, 4:22, 5:23, 6:24, 7:19, 8:20, 9:21}
dy_z1 = {10:16, 11:17, 12:18, 13:13, 14:14, 15:15, 16:10, 17:11, 18:12}
dy_z2 = {val:key for key, val in dy_z0.items()} #dy_z2 = inverse of dy_z0

mapping = {'x': {**dx_z0, **dx_z1, **dx_z2}, 'y': {**dy_z0, **dy_z1, **dy_z2}}
```

Shifting the grid periodically is done by adding the direction of the shift to the cell-coordinate $\boldsymbol{g}$ and modulating by the grid-size,

$$g_x \leftarrow (g_x + \Delta x)\%3. \tag{8.4}$$

[2] https://docs.python.org/3/library/itertools.html#itertools.combinations

I apply these transformations on each set in (the original design-space) $\mathcal{A}$ and add the transformed sets to a new set $\mathcal{B}$. I define the final design-space as the set of configurations $\mathcal{C}$ which contain all elements of $\mathcal{A}$ not in $\mathcal{B}$, or more formally $\mathcal{C} = \left\{\{\boldsymbol{c}^k\}_i \in \mathcal{A} : \{\boldsymbol{c}^k\}_i \notin \mathcal{B}\right\}$. The steps to the procedure of identifying duplicates is also summed up by the pseudo-code in Listing 3 in the Appendix. Performing this algorithm for $k = 9$ results in 173214 distinct configurations, reducing the original design-space by a factor $\sim 1/27$.

Generating Systems

Rather than replicating the atom configuration of the faceted pore from Section 7.1 (in the previous chapter) across the system grid, I have chosen to manually carve the pores from the bulk material in the shape of a dodecahedron. This is a process which is much simpler to perform in practice, and this geometry also greatly resembles the shape of the faceted pore.

I do this with a simple wrapper-class around the `molecular-builder` package, which comes with built-in functionality for creating atomic nanostructures in the wanted shape. This class is initialized with the dimensions of the grid and grid-cells, and the length from the center of the pore to its faceted face. The atomic system can then be created by providing it with a list of grid coordinates, which will carve the pores in the locations given by Equation 8.2. A simple usage example is shown in Listing 2. The resulting output is a file containing the atom coordinates which can be read by LAMMPS directly. In the simulations I perform in this section, I have specified cell-lengths of 50 Å, pore-radii 15 Å and grid size (3, 3, 3). This generates systems of size[3] (152.6 Å)^3 and a total of 327318 atoms (with nine pores present). The source code is available at https://github.com/alexahs/DeepFacet/blob/main/deepfacet/PoreGrid.py.

```python
# Initialize the class with the dimensions of the system
grid = PoreGrid(cell_size = 50, pore_radius = 15, grid_shape = (3, 3, 3))

# Carve pores from the bulk material
grid.create_from_points([(0,0,0), (1,1,1), (2,2,2)])

# Write atom positions
grid.write_data("atoms.data")
```

Listing 2: Example usage of the PoreGrid-class for creating atomic systems with faceted pores placed on a regular grid. This code will generate a system of approx. size $(50 \times 3)^3$ Å with three pores of radius 15 Å located at $\boldsymbol{r}_0 = (25, 25, 25), \boldsymbol{r}_1 = (75, 75, 75)$ and $\boldsymbol{r}_2 = (125, 125, 125)$.

Simulation Specifications

Pore configurations are drawn at random from the set of distinct configurations (Section 8.1.1.1), and the systems are initialized in accordance with the procedure in Section 8.1.1.2. The simulations follow this general procedure:

[3] The system size is required to be a multiple of the unit cell lattice constant, which is why the side lengths are not exactly 150.0 Å. The lattice constant used here here is 4.3596 Å.

1. Equilibration is done for 100 ps, where the temperature and pressure of the system is set to $T = 2200\,\mathrm{K}$ and $P = 0$. This is controlled by Nosè-Hoover thermo/barostats with damping coefficients of 1 ps and 10 ps respectively.

2. The system is compressed by deforming the simulation box along the z-axis with a strain rate of $-0.03\,\frac{1}{\mathrm{ns}}$, while allowing it to expand in the x and y directions. This is done by using the `fix deform` and `fix npt` commands in LAMMPS.

3. The simulation is initially run for 1.7 ns, after which restart files are written.

4. Four additional simulations are run from the checkpoint in 3), initialized with different velocity seeds, and run for another 0.6 ns, totalling 2.3 ns.

The multiple simulations per configuration are run in order to get a better estimate of the yield stress, as thermal fluctuations tend to cause some variations. From the five samples, I calculate the root mean squared deviation from the mean yield stress, and estimate the MD precision as the average of these values across all simulations. During the simulations, the system stress is sampled at intervals of 20 fs. The above procedure is run for a total of 193 configurations, where each run takes approximately 5 hours on a Tesla A100 GPU.

8.1.2 Data Preprocessing

Each of the pore configurations that have been simulated has an associated strength, which is the target value to be estimated by the machine learning model. The ground-truth yield stress is computed by first applying a smoothing filter to the sampled stress values from the simulations, using `scipy.signal.savgol_filter`[4]. This effectively performs a polynomial fit to the stress values in small consecutive windows. The yield strength is then found by simply taking the maximum observed stress value in the simulation, which is sufficient in this case, as the simulations are not run long enough to have multiple yield points. This is done for each of the five simulations per configuration, and the final yield strength is calculated from their average.

The system configurations are represented as a list of coordinates, according to Equation 8.3, which correspond to the pore locations within the system, and this is what will eventually be used as an input to the model. Since we wish to provide the model with as much information about the physical system as possible, I periodically pad the input values to account for that the system in fact extends beyond the 3^3-grid. I do this by first transforming the configuration coordinates to a binary array of size (3, 3, 3), and use `numpy.pad`[5] with a width of two, resulting in a periodically padded array of size (7, 7, 7). Furthermore, a data set of only 193 samples is often insufficient when training deep neural networks, so I use the assumption that symmetric configurations exhibit the same mechanical characteristics in order to increase the size of the data set. This is done by using the algorithm which identifies symmetric configurations in Section 8.1.1.1 - for each configuration, I find its rotational symmetric images, label them with the same yield strength as was found by simulation for the original, and add them to the data set[6]. This increases the size of the data set by a factor of eight, totalling 1544 samples.

[4] https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.savgol_filter.html

[5] https://numpy.org/doc/stable/reference/generated/numpy.pad.html

[6] The periodic images of the configurations are not included in the data set, as this would indirectly cause leakage from the training set to the test set, due to the periodic padding.

8.1.3 Model Selection

Since we are dealing with data pertaining to a physical system with spatial dependencies, the logical choice is to use a convolutional neural network (CNN) for this task. I also, however, include a simpler dense neural network (DNN) for comparison.

Choosing a suitable architecture for the models can be a time-consuming and challenging process, simply due to the large number of hyperparameters which can be tuned and varied. I have therefore chosen an architecture similar to one which has been used previously on a related task by *Hanakata et al.* [4]. In their work, the CNN consists of three (2D) convolutional layers followed by a fully connected layer, with pooling layers following the convolutional layers. The size of my input data is very low, however, so the pooling layers are omitted in my architecture all together. I am also working with volumetric data, so the convolutional layers in my CNN are three-dimensional in space. This is done by using PyTorch's `Conv3d` module[7].

Hyperparameters such as the learning rate and choice of activation function are found by trial and error. The remaining architectural parameters (no. of nodes, kernel size etc.) are found by performing a brute-force grid search over the parameters listed in Table 8.1. The architectures are evaluated using k-fold cross-validation with $k = 5$. The one with the lowest average validation loss is chosen as the optimal model. The data set is split into 80% for the cross-validation process, and 20% is used for testing the performance of the model once the optimal set of parameters has been found.

I use the ADAM optimizer [60] with a learning rate $\eta = 10^{-5}$ for optimizing the mean squared error (MSE) loss (Eq. 4.4). The LeakyReLU (Eq. 4.3) with a negative slope of $\alpha = 0.01$ is chosen as the activation function for all layers. The models are trained for 300 epochs, with a minibatch size of 32. Model performance is measured using the MSE and R^2 scores.

Table 8.1: Model parameters searched over using cross-validation, in total 63 and 81 runs for the DNN and CNN, respectively.

	No. of dense nodes [$\log_2$]	No. of dense layers [$\log_2$]	No. of kernels [$\log_2$]	Kernel size
DNN	$\{2, 3, ...10\}$	$\{1, 2, ...7\}$	-	-
CNN	$\{2, 3, ...10\}$	0	{3, 4, 5}, {4, 5, 6}, {5, 6, 7}	$3^3, 4^3, 5^3$

8.1.4 Results

Figure 8.4 shows the results from the grid search of optimal parameters for the CNN. The root-mean-squared-error and R^2 scores are plotted as a function of the number of neurons in the dense layer, for different numbers of kernels in the convolutional layers. From the figure, it is clear that a larger number of kernels increase the model performance, which is generally also the case for the number of nodes in the fully connected layer. The optimum is found at

no. of dense nodes: 512, no. of kernels: $\{32, 64, 128\}$, kernel size: $4 \times 4 \times 4$,

also illustrated in Figure 8.3. This model achieves an R^2 score of 0.57 and RMSE of 0.154 GPa

[7] https://pytorch.org/docs/stable/generated/torch.nn.Conv3d.html

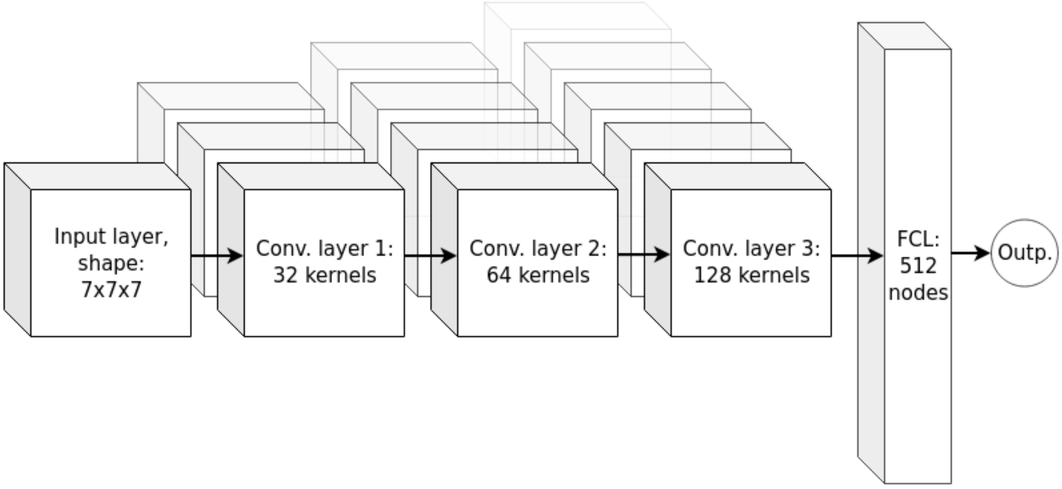


Figure 8.3: Illustration of the CNN with the optimal parameters determined by the grid-search.

on the test data, slightly worse than the MD precision approximated to be 0.111 GPa. The performance on the training and validation data are listed in Table 8.2. In Figure 8.5, the MSE loss and R^2 scores during the model training are shown. The training and validation loss are approximately equal for the first 200 epochs, after which the training loss continues to drop and validation loss looks to plateau. I also show the model predictions of the yield stress on the test set vs. the true values from the MD simulations in Figure 8.6. The results from the *dense* neural network are largely omitted, as it was not able to describe the variance in the data which we are interested in predicting, and can be summed up by reporting that most of the architectures converged towards outputting the mean yield stress of the entire training set.

Table 8.2: Root-mean-squared-error (RMSE) and R^2 scores of the optimal CNN on the training, validation and test data sets. For reference, the MD precision is approximated to 0.111 GPa.

	Train	Validation	Test
R^2 score	0.82	0.65	0.57
RMSE [GPa]	0.096	0.136	0.154

From the results of the grid-search in Figure 8.4 the immediate take-away is that deeper networks are better. However, comparing the achieved RMSE training score in Table 8.2 with the MD precision suggests that it is slightly overfitted. This could perhaps have been alleviated by stopping the training process when the error passes this threshold. It is also evident from the training history in Figure 8.5 that the validation loss hardly improves beyond 200 epochs. Over all, the model performs rather satisfactory with a RMSE which is only $\sim$ 0.04 GPa above the MD precision, when considering the range of yield stresses within the data set is approx. 1 GPa. Looking back at Figure 8.6, approximately half of the predictions fall within the indication lines of the MD precision, but the model does look slightly biased towards predicting lower values for the yield stress. This is especially the case when considering configurations of yield stress 9.4 GPa and higher.

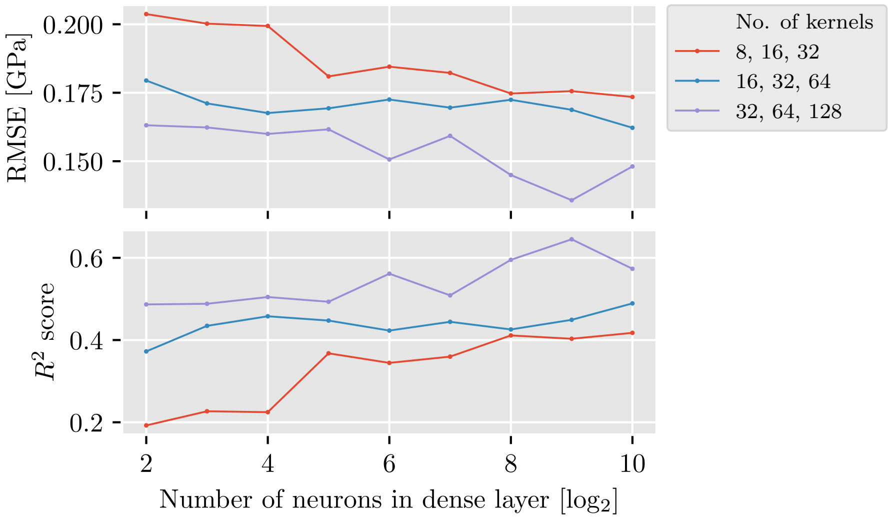


Figure 8.4: Root mean squared error (RMSE) and R^2 scores on the validation data for yield stress as a function of number of neurons in the dense layer, for different number of kernels in the convolutional layers of the CNN. Kernel sizes are here fixed at a size of 4 × 4 × 4.

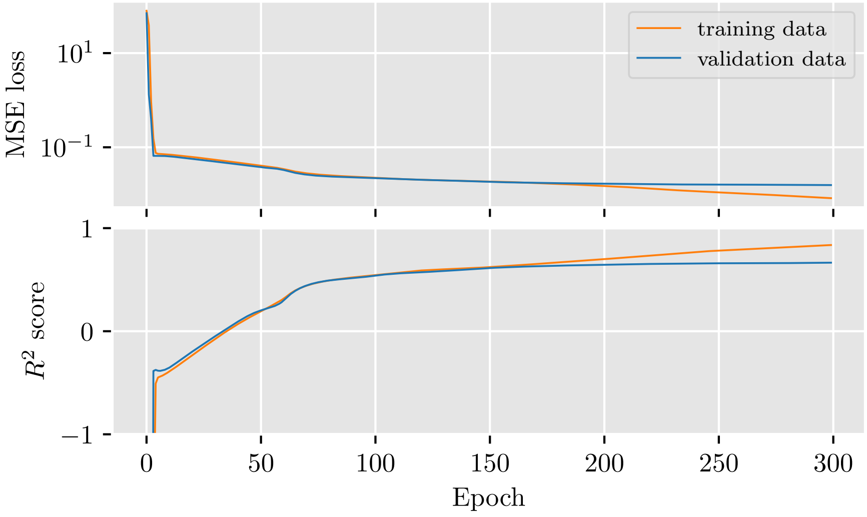


Figure 8.5: MSE loss (logarithmic) and R^2 scores as a function of number of epochs during the training process of the best performing CNN.

The network used by *Hanakata et al.*[4] on a similar task was considerably more shallow than the one I have used, and also perform better with respect to the prediction error. It is not immediately clear why my task requires a more complex architecture, but it may be related to the fact that my input size is very small, and each variable is a boolean which represents

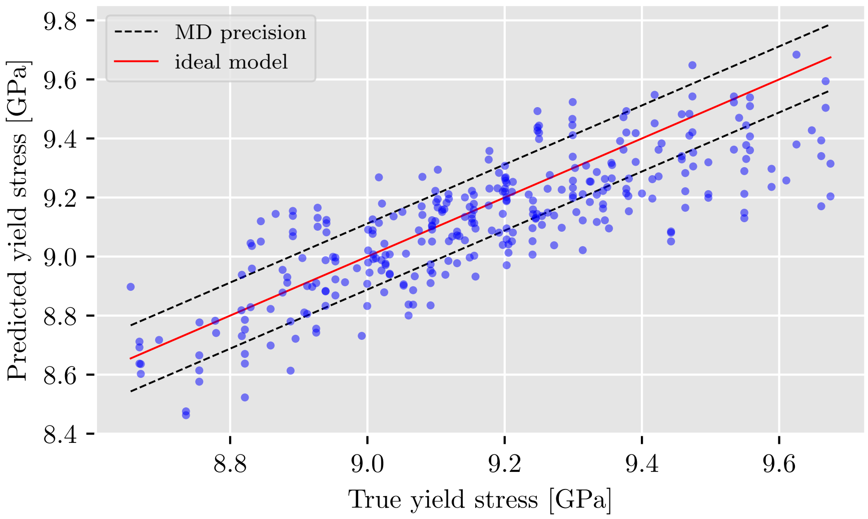


Figure 8.6: Predictions of the best performing CNN on the test data vs. the true yield stress. The dashed lines in black indicate the MD precision and in red the line $x = y$ (ideal model).

a large volume of the system. This says very little about the actual geometry of the physical system itself, other than that something exists at that location. The strength of CNNs is to learn local correlations, this becomes difficult when the resolution of the input is low. I do, however, provide it with information about the periodicity of the system, which in a sense increases the resolution slightly, and this did in the end play a large part in helping the network converge. An unfortunate side-effect of having low-resolution features is that the CNN does not produce feature maps which are easily interpreted by humans. This means that it is difficult to gain any insights into what the network deems as important to solving the actual physical problem.

The physical system itself, such as I have defined it, consists of faceted pores located at fixed positions on a regular grid. This has enabled a simple solution to removing configurations from the design space which in theory should give the same results in a MD simulation. In turn this has been used to drastically increase the size of the data set available for model training by synthetically creating new samples by rotating them. An oversight and a flaw in my implementation, though, is that the system size is not exactly a multiple of the grid-cell size - a result of it getting rounded to the nearest unit-cell multiple. This leads to a non-uniform distribution of the possible pore-locations, e.g. a pore at $x = 2$ in grid-space is centered at $r_{x=2} = 125$ in simulation-box-space instead of at $r_{x=2} = 126.9$, which is the correct location for a system of size 152.6 Å according to my definition in Equation 8.2, Section 8.1.1.1. The difference in distance between two pores at $x = [0, 1]$ and at $x = [0, 2]$ then comes out to approx. 2.6 Å, over *half* the length of a unit cell. In essence this means that the symmetry arguments for the design-space definition and sample-duplication for the model training data are not strictly valid. How much this has affected my results is unclear and not easily measured without redoing the simulations.

8.2 Accelerated Search Algorithm

In the previous section, I trained a machine learning model which approximated the yield strength of a handful of pore configurations. In this section I investigate whether this model can be used to effectively search through the complete space of configurations of this system and identify optimal geometries[8]. Due to the large number of configurations which must be explored, this is not feasible to do by brute-force MD alone. Doing so would take approximately 4 centuries using the *six* state-of-the-art Nvidia A100 GPUs I have had access to. The model can infer the yield strength of a configuration many orders of magnitude quicker than actually performing the MD simulation, and by allowing it to suggest configurations of a desired property, we can perform simulations of these and retrain the model on the results in an evolutionary manner. The idea is that the model should converge towards finding the optimal geometry, given enough generations.

8.2.1 Implementation

Since I now have a CNN which approximates the yield strength of a given pore configuration in a fraction of a second, I no longer need to restrict myself to the *reduced* design-space consisting of only symmetrically distinct configurations, which was done in the previous section. I therefore chose to search for optimal geometries in the *full* library of $\frac{27!}{9!18!} \approx 4.7 \times 10^6$ configurations, where the symmetric configurations have not been removed. I do, however, also conduct the following experiment on the reduced design-space for comparison.

The reduced library is generated as in the previous section (Sec. 8.1.1.1), and to generate the full library I use Python's `itertools.combinations`[9] function, providing it with a list of the 27 grid-cell ids:

```python
from itertools import combinations

#define grid cell ids and number of pores
grid_cell_ids = list(range(1, 28))
num_pores = 9

#create an iterator with the full set of possible combinations
full_space = combinations(grid_cell_ids, num_pores)

# number of configurations corresponds
# to the binomial coefficient (n, k) = (27, 9)
nCk = len(list(full_space))
```

I use the CNN with the optimal architecture, as found in Section 8.1.4. Data preprocessing follows the procedure in Section 8.1.2, though the samples are here split into 80% for training and 20% for testing following each generation. The algorithm to identify the optimal geometries is inspired by the work of *Hanakata et al.* [4], and it follows this general procedure:

[8] In this context, an optimal geometry is a configuration of pores which has the highest or lowest possible yield strength.

[9] https://docs.python.org/3/library/itertools.html

1. The CNN is trained on the 1544 samples from 193 randomly selected configurations from Sectrion 8.1.1
2. The library of possible pore configurations is fed through the CNN, inferring the yield stress of each configuration. This corresponds to approx. 4.7×10^6 configurations in the full design-space and 1.7×10^5 in the reduced.
3. Each configuration is now assicoated with an inferred yield strength, and the top ten performers are selected for MD simulation (and removed from the unexplored library), such that the true strength of these configurations can be established. The simulation specifications are given in Section 8.1.1.3.
4. The results from the simulations are added to the data set for training the next model generation. The data set is now composed of the original 1544 samples and the top ten performers of the current and previous generations.
5. The CNN is retrained from "scratch" on the new data set.
6. Continue from step 2) until stop criteria is met.

In the results presented below, this algorithm is applied independently for identifying configurations of the following five criteria:

- Highest yield strength (full and reduced design-space)
- Lowest yield strength (full and reduced design-space)
- Target yield strength of 9.4 GPa (reduced design-space)

The target value of 9.4 GPa is chosen somewhat arbitrarily in between the mean and max yield stress of the 193 randomly selected configurations in the initial training samples. Ideally, the stop criteria would be when the search algorithm converges to an optimal geometry. I have opted to stop the algorithm after *five* generations.

8.2.2 Results

Figure 8.7 a) and b) shows the mean yield stress of the configurations found by the search algorithm in the reduced and full design-spaces as a function of number of generations (10 configurations per generation). In the reduced library, the algorithm immediately finds systems which are stronger than ones it has previously seen, but its performance does not improve with more generations. Searching through the full design-space results in configurations which are stronger still. Here it is also the case that the strongest identified system is found in the first generation, and with diminishing strength as the generations increase. By generation 4, it looks as if the full design-space is depleted of the strongest configurations, as it here proposes ones which are substantially weaker.

Figure 8.7 c) shows three distributions of yield stresses. In blue are the initial 193 training samples (which are selected randomly), hereby referred to as the *random* search. In red and green are the configurations which have been identified by the search algorithm, in the full and

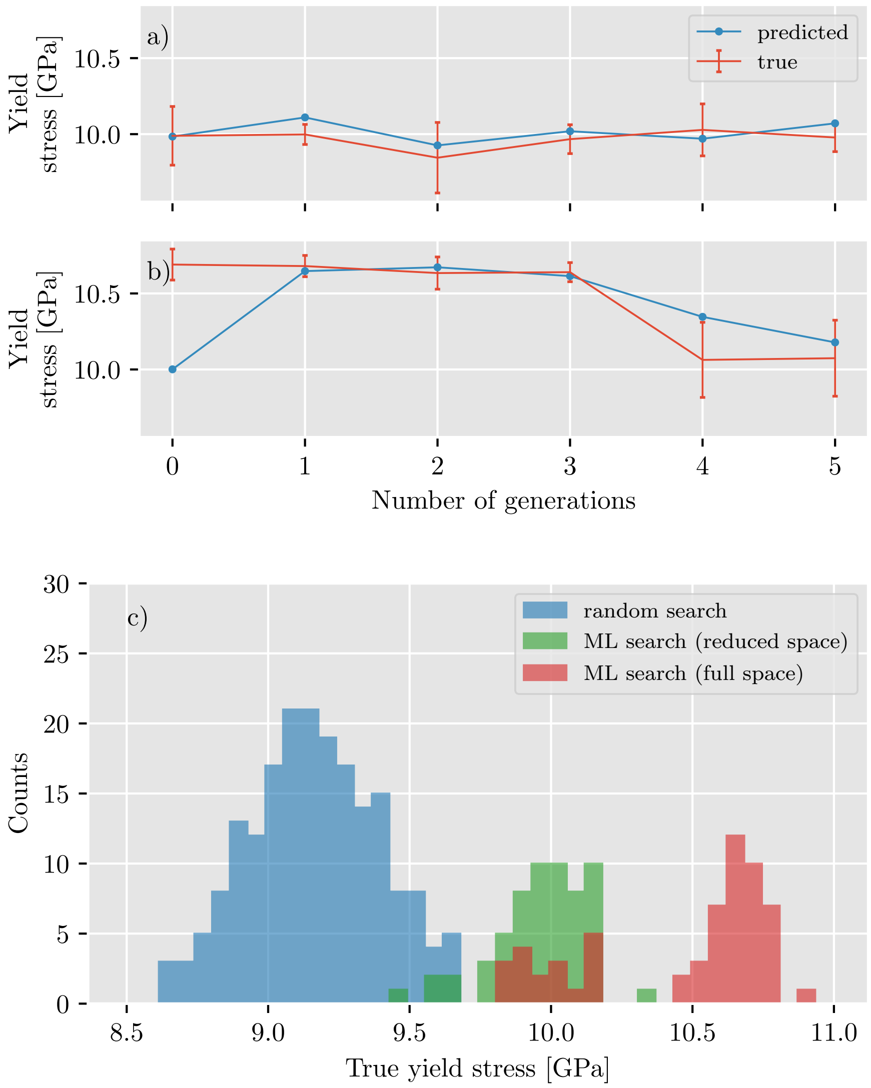


Figure 8.7: Average yield stress as a function of number of generations when searching for the strongest configurations in the reduced (a) and full (b) design-space. Errorbars indicate one standard deviation in the top ten performers. (c) Strength distribution of configurations chosen at random (initial 193 training samples), and of the configurations proposed by the network.

reduced design-spaces respectively. These distributions are the sum all of the generations in the search algorithm, and the presented values for the yield stress have been verified by simulations. The figure shows that the algorithm is generally effective in identifying configurations with higher strengths than the random distribution, where the strongest ones identified have yield strengths 10.91 GPa and 10.35 GPa, in the full and reduced design-spaces respectively. For comparison, the yield strength distribution of the random samples has a mean value of 9.16 GPa, a maximum of 9.67 GPa and a range of 1.01 GPa.

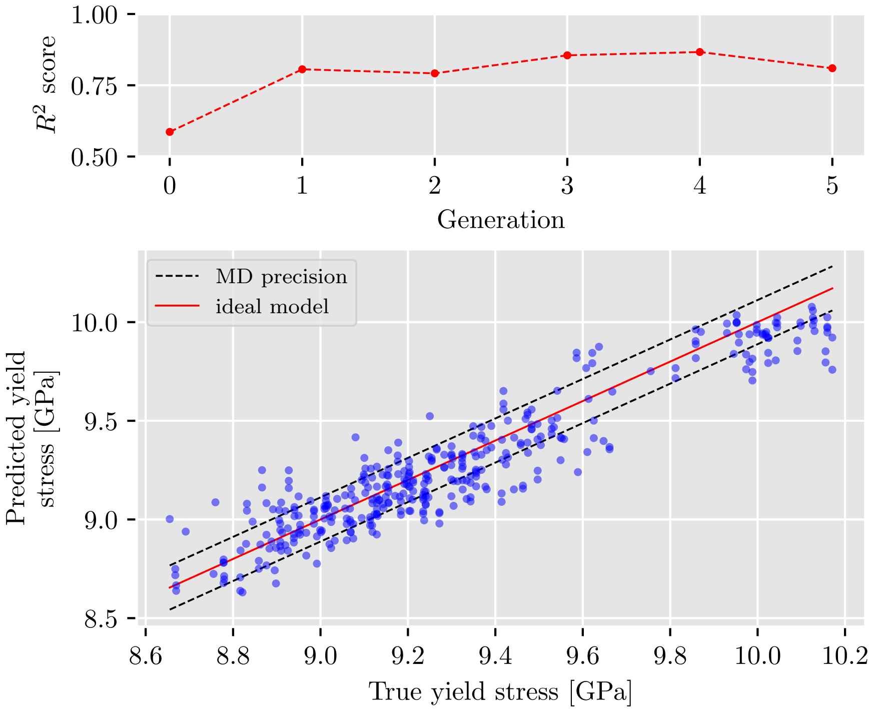


Figure 8.8: (Top) R^2 score on the test data as a function of generations when using the convolutional neural network to search for strong geometries in the reduced design-space. (Bottom) Predicted yield stress as a function of true yield stress in the final generation. The MD precision is indicated by the dashed lines.

Figure 8.8 (top) shows the performance of the model with respect to the R^2 scores on the test set, for each of the generations when searching for the strongest configurations in the reduced design-space. The model performance increases drastically after the zeroth generation, and reaches a peak score of $R^2 = 0.87$ in generation 4, before dropping to $R^2 = 0.81$ in the final generation. To the bottom of the same figure (Fig. 8.8), the predictions are shown as a function of the labels in the generation 4.

Similarly, Figure 8.9 shows the corresponding results pertaining to the minimization of the yield strength. The algorithm identifies configurations which are marginally weaker than in the random search, but it is also the case here that increasingly weaker configurations are not identified with later generations. The figure (Fig. 8.9ab) also shows the predictions of the CNN, and in early generations it tends to underestimate the yield strength with approximately 0.5 GPa. This, however, does improve as the CNN sees more low-strength configurations during training in later generations. The yield strength of the weakest configurations found by the algorithm in the full and reduced design-spaces respectively are 8.35 GPa and 8.30 GPa, compared to the random search which gave lowest yield strength 8.66 GPa.

The results of applying this search algorithm to a pre-defined target yield stress value of 9.4 GPa are shown in Figure 8.10. In the top subfigure I show mean yield stress of the configurations

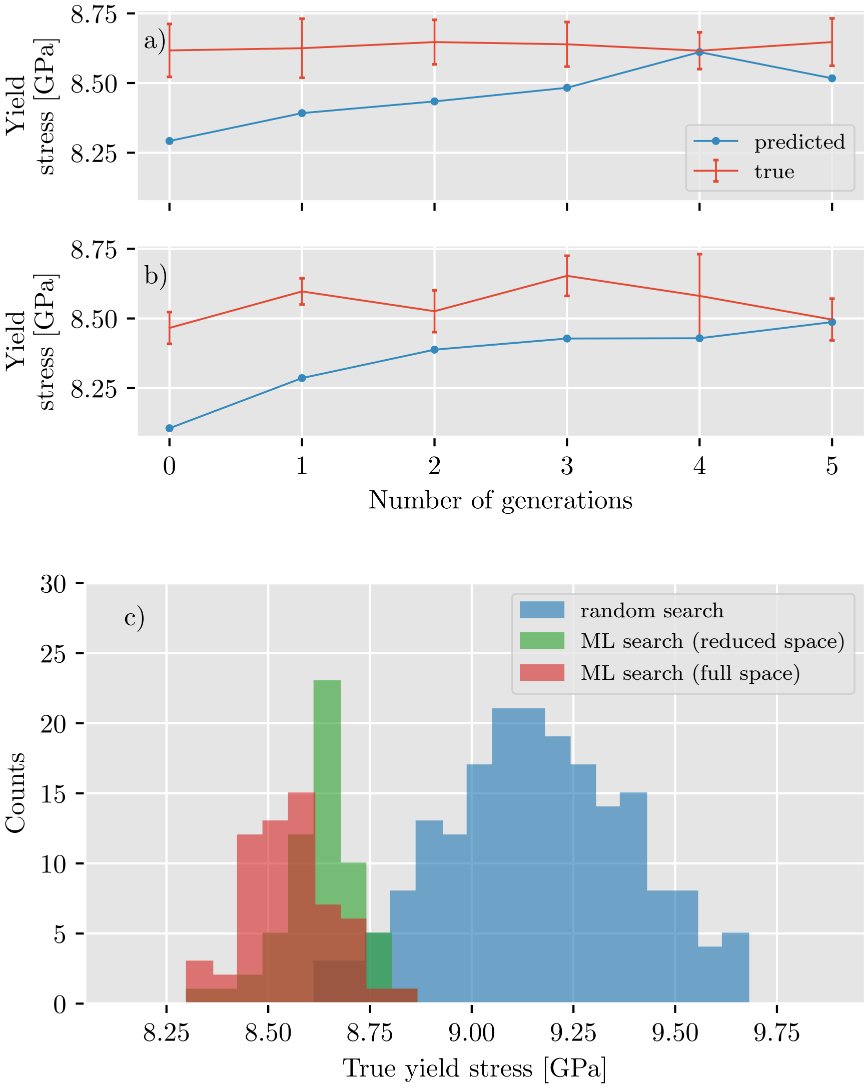


Figure 8.9: Average yield stress as a function of number of generations when searching for the weakest configurations in the reduced (a) and full (b) design-space. (c) Strength distribution of configurations chosen at random, and of the configurations proposed by the network.

proposed by the network per generation, and we see that it oscillates around the target value. The errorbars in this subfigure indicate one standard deviation, and it shows that the yield stress of the proposed configurations vary wildly, and are consistently high for all generations. The histogram to the bottom of the figure (Fig. 8.10) shows the yield strength distribution of the proposed configurations for all of the generations, which has a mean value of 9.41 GPa and a standard deviation of 0.19 GPa. The width of this distribution is in other words almost as wide as the random search distribution itself, which has a standard deviation of 0.23 GPa.

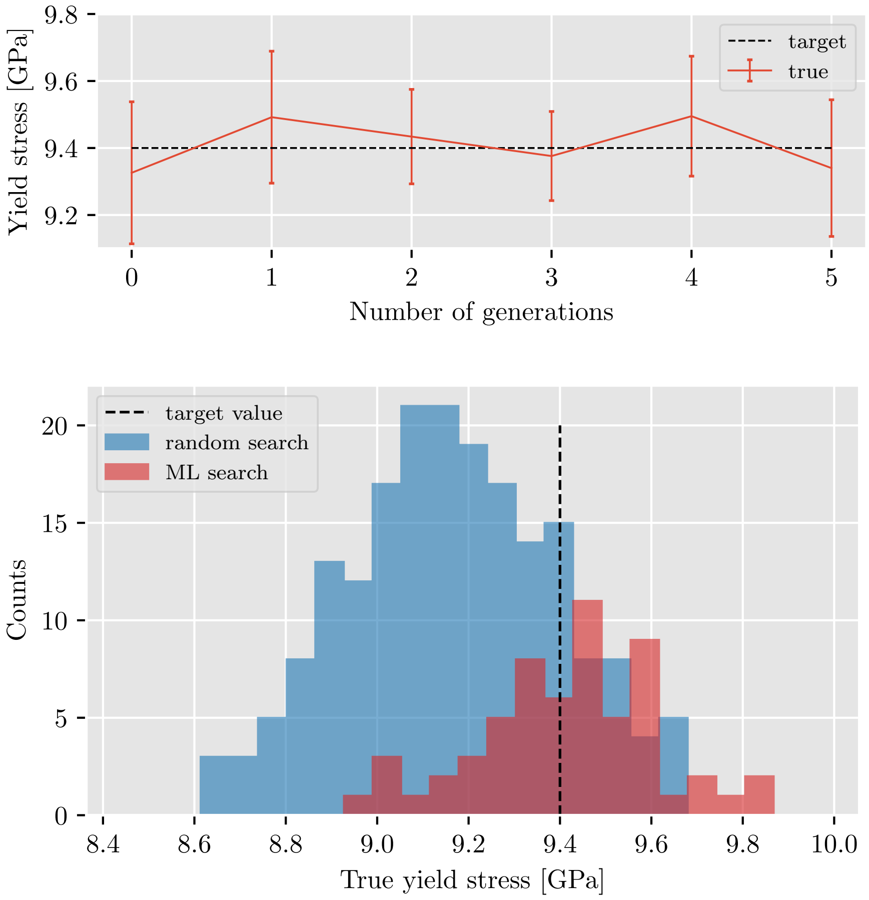


Figure 8.10: (Top) Average yield stress as a function of number of generations when searching for configurations with a yield stress of 9.4 GPa. The errorbars indicate one standard deviation. (Bottom) Strength distribution of configurations chosen at random, and of the configurations proposed by the network, from all generations. The black dashed line indicates the target value for the search algorithm.

Finally, the pore configurations of the strongest and weakest systems found by the algorithm are illustrated in Figure 8.11. In the left column of the figure, we see that the strongest system has an ordered configuration corresponding to a ”wall” of pores aligned along the axis of applied stress (z-axis), with three columns of pores placed in adjacent locations on the grid, spanning the (010)-plane. The pores in the weakest system are oriented in a diagonal fashion, where each grid location in the (011)-plane is occupied by a pore, similar to the strong configuration, but rotated 45 degrees. The figure also shows the dislocation planes which nucleate when the systems yield, throughout the (111)-plane of the entire system. The dislocations seem to propagate along the facets of the pores, as was also the case in the single pore system in Section 7.3. Other pore configurations which are identified by the algorithm for both criteria, are either symmetric or periodic permutations of the configurations shown in Figure 8.11, or they are slight variations with often only a single pore displaced from its apparent optimal grid position. The strongest and

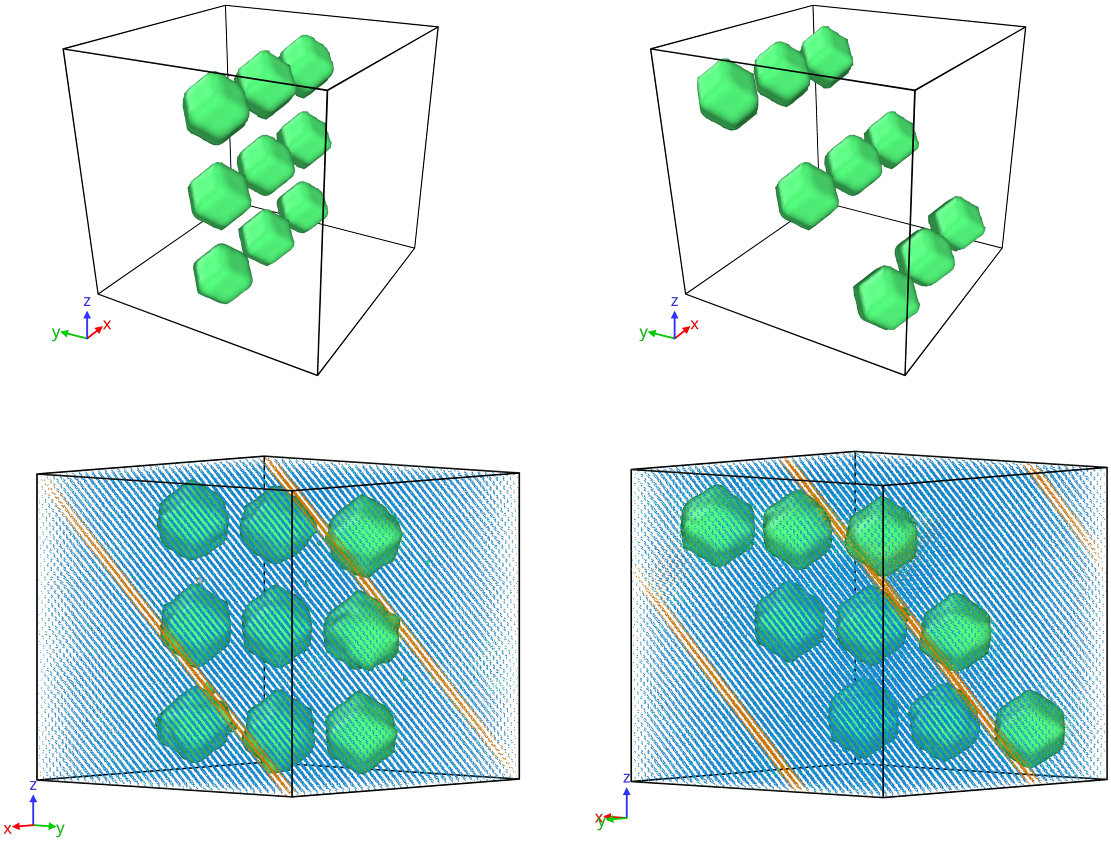


Figure 8.11: Illustrations of the pore configurations with the highest (left column) and lowest (right column) yield stresses, identified by the search algorithm in the full design-space. The green meshes outline the pores, and dislocation planes which are nucleated at the yield point are illustrated by the orange atoms.

weakest configurations from the *reduced* design-space are shown in Figure 10.5 in the Appendix.

With high yield strength as the search criteria for the algorithm, it successfully identifies configurations which lie well outside the distribution of yield stresses of the random search, especially when searching through the full design-space. Interestingly, the CNN initially predicts a considerably lower yield strength (Fig. 8.7) for the configurations in generation zero, but it has already then learned the general organization of the pores within the system which corresponds to high strength. It does not, however, improve its performance in later generations with respect to identifying stronger configurations, even though the CNN performs increasingly better on the test data. A possible explanation is that five generations is not enough when only the top ten contenders are simulated per generation, but one would perhaps still expect to see a positive trend, as was done already in generation number two in the original paper by *Hanakata et al.* [4]. My results, on the other hand, suggest that the design-space is depleted of the strongest configurations by generation 4, as seen in Figure 8.7b. Considering the fact that my CNN only suggests variations of the two ordered configurations shown in Figure 8.11, it may be that it becomes subject to overfitting and biased towards these and their variations. Conversely, it is a possibility that this system, being a regular $3 \times 3 \times 3$ grid with nine pores, is too crude to contain any other structured pore configuration which optimizes its strength, assuming for a moment that the true optimum is not a seemingly random pore distribution.

We also saw that there was a large difference in strength in the identified configurations of the reduced and full design-spaces. This suggests that the strongest configurations simply do not exist in the reduced design-space, and that removing those which are symmetrically and periodically identical from consideration was a design flaw in the method. It may also be related to the fact that the symmetry arguments for the configurations are not strictly valid for the system I have simulated, as was discussed in Section 8.1.4.

The arrangement of the pores illustrated in Figure 8.11 is rather satisfying, but it is not immediately obvious why exactly these configurations constitute the two optimums. Considering that the yield mechanism for this system is related to the formation of dislocations in the (111)-plane, it may be that it is dependent on the number of nucleation points, i.e. pore vertices or pore-bulk interfaces, which are aligned in these planes. It can be seen in Figure 8.11 that the dislocations intersect with six of the pores in both systems, but the intersection points in the weak system are in closer proximity to each other, which suggests that the distances between nucleation points (along the possible (111)-planes) are factors as well. This observation is similar to that of *Pook* [75], where the stress intensity in an array of crack tips along the axis of applied stress decreased with increasing distances between them. The plane in which the pores are arranged in the low yield stress system shares two components of the Miller-indices with those of the dislocations, whereas the strong system only has one in common. Because of this, and considering all possible (111)-planes of the system, it may be that a larger number of nucleation sites with close proximity are aligned in these planes, resulting in the system having a higher probability of yielding at an earlier point. These observations are based on a handful of visual inspections of the system only, which does not constitute a thorough analysis, and as the time resolution of the dumpfiles containing the atom positions from my simulations is very low, this is left as future work.

Chapter 9

Summary

The work presented in this thesis covers several topics. Firstly, the setup of a simple silicon carbide system where a faceting process could be observed in a molecular dynamics simulation was carried out, and subsequently a closer examination of how it behaved during faceting. The mechanical characteristics of a single pore have been studied, but also of a more complex system of numerous pores. Here, a machine learning model was used for inverse design of both high and low strength geometries. In this chapter I try to draw some conclusions and summarize the different topics in terms of what was achieved, as well as provide some prospects for possible extensions.

9.1 Summary and Conclusions

Faceting of a spherical inclusion: As the first topic of this work, I investigated whether it was possible to observe a silicon carbide material with a spherical inclusion transform into a faceted shape. Secondary to this, the goal was to find a suitable scaling of this system with respect to its size, such that it would be possible to perform repeated simulations of it effectively. This was achieved, although the configuration which was chosen was bordering on the edge of the size limit of how small it could be made, and still be characterized as a faceted system, as the length across the (110) facets were comparable to the size of the SiC-3C unit cell itself.

Force of Crystallization: Related to this was a closer examination of the faceting process itself. Here, I set out to measure its force of crystallization, in order to provide insights into the mechanisms of the stress induced by relaxing nanoparticles. The actual *force* was measured, i.e. the pressure (~ 30 MPa), and this was comparable to previous measurements in salt crystallization (~ 150 MPa). It was found to be independent of temperature. Due to the system size being relatively small (approx. 12000 atoms), the pressure was dominated by high fluctuations, resulting in a large number of simulations which had be performed in order to get a good enough average reading. This also limited the total time I was able to run the simulations, which is not ideal when studying relatively slow relaxation processes. Since this was a NVT-type simulation, it also required a parameter-sweep over a range of barostat-configurations for the initial thermalization process. In the end, the time-evolution of the potential energy during faceting proved the most reliable, and an analysis of it lead to the suggestion that the faceting process of this system relaxes similar to that of a glassy material. Here, the time-stretching factor β was found to be in agreement with the diffusion of atoms in a medium of static traps, which was interpreted as an effect of the growing pore trapping surface atoms in place for increasingly long periods as the system relaxed.

Mechanical properties of the faceted inclusion: Following this, the faceted pore was subjected to a compressive stress, and its behaviour was studied for a range of temperatures. From these simulations, we saw that the system deformed plastically, with dislocations nucleating along the bulk-space interface. The Young's modulus and yield strength of the system was measured, revealing linear dependencies on temperature. Most interestingly, by studying the pore volume as a function of stress, the system exhibited a behaviour where the pore volume decreased in an accelerated manner leading up to the point of yield. This observation motivated a series of simulations of this system at high levels of constant stress in the elastic regime of the system. An Arrhenius-type scaling relationship between the stress-strain curves of different temperatures was also proposed, successfully collapsing them, but this was mostly empirical fits and we have yet to extract useful insights from it.

The faceted pore was also studied under conditions of constant applied stress. Here, a series of creep-type simulations were performed, and the breaking time of the system was studied as a function of the applied stress and the temperature. The results showed that the breaking times increased step-like near a critical stress value, within the maximum time of the simulation. With this in mind, it is apparent that these simulations ideally should have been run for longer, as the functional form of the breaking times was hard to determine with such limited simulation times. This was not pursued further due to the high computational cost of performing a large number of these simulations. Nevertheless, the simulation results were compared with an expression for time-dependent rupture of materials under constant stress. This did not prove fruitful, as the comparison of the results with this model yielded unphyscal values for the parameters of the model, unless an assumption was made about the temperature dependence of the energy barrier of the rupture. The yielding of this system is the nucleation of dislocations, but this mechanism is inconsistent with the premise of the rupture model, which is creep-flow. In conclusion, a different theoretical prediction is needed in order to gain new insights into the time dependence of dislocation nucleation of this system at constant stress.

Predictive modelling of the yield stress: The strength of a much larger system was also examined, where the material consisted of several faceted inclusions. The system strength was here studied as a function of the locations of the pores within the material in a system, restricted to be of constant porosity. The restriction of nine pores was enforced primarily because of the large number of simulations which would have had to have been performed in order to attain enough samples within each porosity to train a machine learning model effectively. An additional reason for this choice is that the strength of a material often can be described by a simple model involving only the porosity itself [73]. A convolutional neural network was then tasked with predicting the strength of this system. It was able to predict the strength of the different materials rather successfully, using only the results from 193 simulations. It achieved a RMSE of 0.154 GPa on the test data, comparable to the molecular dynamics precision of 0.111 GPa, and an R^2 score of 0.57.

Inverse design of a multiple pore system: This model was then used to search through the complete space of materials with nine pores present, in order to identify the high and low strength configurations within it, using the methodology of *Hanakata et al.* [4]. Compared to the yield strengths of the 193 systems which were randomly chosen for simulation and used to train the model initially, this method resulted in the identification of configurations which were approximately 1.2 GPa, or 13% stronger. Similarly, weak strength configurations were also identified, though this only resulted in a slight performance increase compared to the random search, with a difference in strength of 0.36 GPa (4% weaker). The model was also used to target configurations with the specific yield stress of 9.4 GPa. This resulted in a distribution of strengths with a mean value of 9.41 GPa, however its width was almost as wide as the distribution of strengths in the original 193 simulations - Though the target value was met, the model was not as effective at identifying configurations with specific strengths, compared to when it was tasked with finding systems of high strength.

The performance of the model increased in later generations in terms of its predictive capabilities, and went from an R^2 score of below 0.6 in the zeroth generations to a score of $R^2 = 0.81$ in generation one, and peaked out at $R^2 = 0.87$ in generation four. When working with synthetic data such as in this thesis, obtaining ground truths for samples outside of the domain of what the predictive model has seen is possible. Generatively training models in such a manner therefore seems to be an effective method for improving their predictive performances.

The strongest and weakest geometries which were identified by the search algorithm were visually inspected. This revealed that the pore configurations of these two extremes were organized in an ordered fashion. Here, the strongest geometry had its pores stacked vertically along the axis of the applied stress, effectively forming a wall of pores. The pores in the weakest configuration were organized similarly, but rotated 45° relative to this axis. Little analysis was done on why exactly these configurations constituted the strongest and weakest geometries, but it was hypothesized that the differing strengths were related to the proximity of dislocation nucleation points in the (111)-planes of the system.

Even though the predictive model was effective in identifying systems with high strengths which were well outside the domain of what it had seen previously, it did not improve its performance in doing so with later generations. This was also the case for low and target yield strength systems. This, combined with the fact that the high and low strength structures it discovered were distinct and well ordered, leads us to the conclusion that the search algorithm converged to the two optimums of this system - Though without a baseline to compare it to, this is not verifiable.

9.2 Outlook

During the course of this study, a number of topics have been touched upon, however within the scope of this thesis, it has not been possible to extensively pursue them all. The following presents possible extensions of some topics which seem promising for future research.

Multiple pore systems: The most straight-forward extension of this study, is to perform the search algorithm for optimized designs of the multiple pore system for other porosities than just nine pores. The pipeline for doing this has largely been developed during this work, and would "only" require more GPU-hours for performing the simulations. An important note, though, is the necessity of a baseline for the search algorithm, as was done by *Hanakata et al.* [4], such that it can be established whether it is capable of identifying the *true* optimums of this system.

This can straight-forwardly be done by constructing a smaller system, i.e. a smaller pore grid, for instance of size $2 \times 2 \times 3$, where the number of configurations is small enough to be simulated directly. Increasing the dimensionality of the system, e.g. $4 \times 4 \times 4$ grids, is also possible, with the added benefit of possibilities for more complex geometries constituting the optimums than was observed in the $3 \times 3 \times 3$ system - Though increasing the system size quickly gets constrained by the computational cost of performing the simulations. An extension which would require some further development of the method, but which may be more closely related to natural systems, is to examine systems where the pores are not strictly constrained to a regular grid, or not of equal size. This may include a definition of the system where the pore locations have some leeway in their placement within their respective grid cells. This may, however, introduce more complex dynamics, in which pores may fuse together into larger voids if they are too close, as was observed in the first part of this thesis.

Due to the surprisingly simple geometries which were observed to constitute the strongest and weakest pore configurations by the search algorithm, it may be possible to deduce a theoretical prediction for the optimal pore locations of this system by geometric considerations. Firstly, this may require a statistical analysis of *where* along the pore-bulk interface the dislocations are probable to nucleate, followed by how these sites are situated in relation to neighbouring pores. Although this analysis can be done by a series of molecular dynamics simulations, possibly using the data from the ones I have performed, it is not straight forward as to exactly how it should be implemented.

Dynamics of the isolate inclusion: Compression of the isolate inclusion revealed a dynamic near the point of yield, where its volume was reduced at an accelerated rate leading up towards the yielding of the system. This seemed to be correlated with the hyperelastic softening. A more thorough statistical analysis needs to be done in order to establish whether these observations are general for this system or not. This can be done by performing simulations of the system in a state where it is arbitrarily close to yielding, though this may prove challenging due to the seemingly spontaneous dislocation nucleation at high stress. Of interest is also an examination of which regions of the pore surface contribute to the "inwards growing", possibly revealing a healing effect in areas of locally high stresses.

Force of Crystallization: The pressure buildup during the relaxation of the pore from its spherical to faceted shape is quite significant, and was found to not be temperature dependent. A continuation of this study would be to investigate whether it has any dependencies at all, such as the porosity or size of the pore. A detailed examination of the faceting mechanisms of silicon carbide has been provided by *Sveinsson et al.* [26]. Further studies of this process in the porous system may also reveal insights relating to how the grows mechanisms are affected when subjected to external pressure.

Part III

Appendix

Chapter 10

Supplementary material

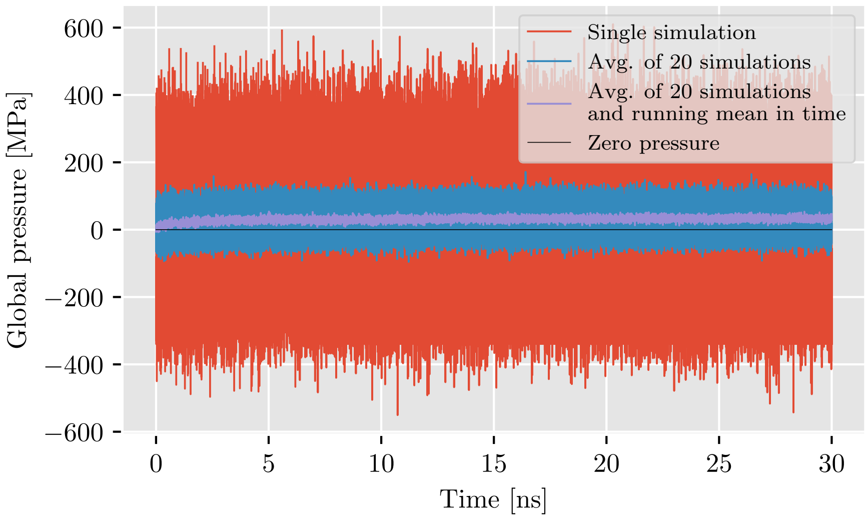


Figure 10.1: Pressure as a function of time when measuring the Force of Crystallization of the faceting process of the spherical inclusion. The figure shows the measurements in both a single simulation and from the average of 20 simulations for comparison. The black line indicates $P = 0$ and is meant to guide the eye. The plot illustrates the difficulties of pressure measurements in small system NVT simulations, and the need for averaging over multiple simulations in order to detect a weak signal.

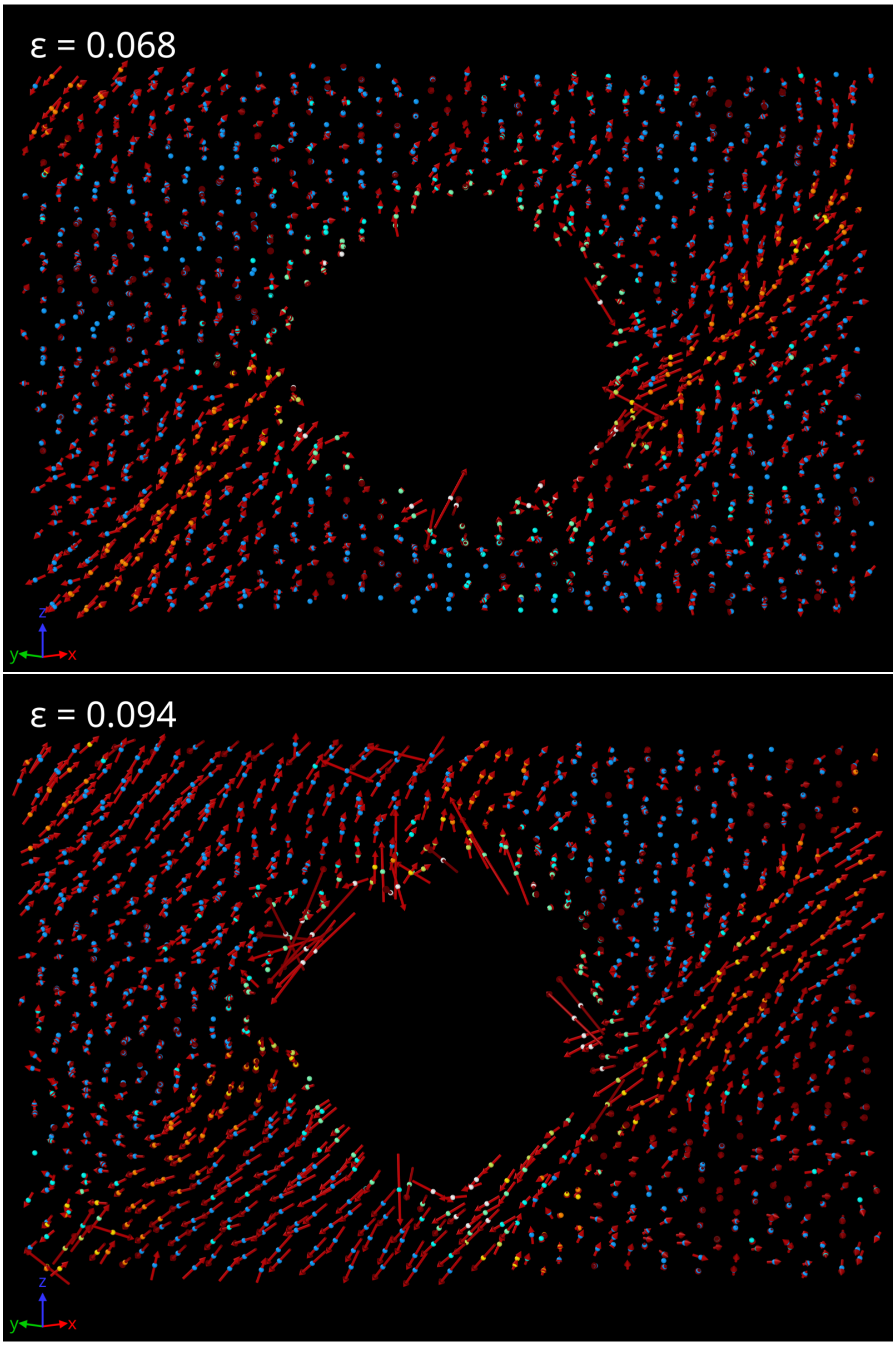


Figure 10.2: Cross-sectional snapshots of the faceted pore under uniaxial compression in the z-direction, showing the atoms (colored dots) and their displacement vectors (red arrows) during the preceding 50 ps. Stress builds up in the system, and is released with the nucleation of dislocation planes, indicated by the orange atoms in the top subfigure. The dislocations are located along the interface of the bulk material and the (111) facets of the pore. As further compression is applied, the stress is again increased, followed by a step of dislocation creep along the slip planes, seen in the bottom subfigure. The magnitudes of the displacement vectors are scaled by a factor of two for visibility. Renderings are constructed using Ovito [54].

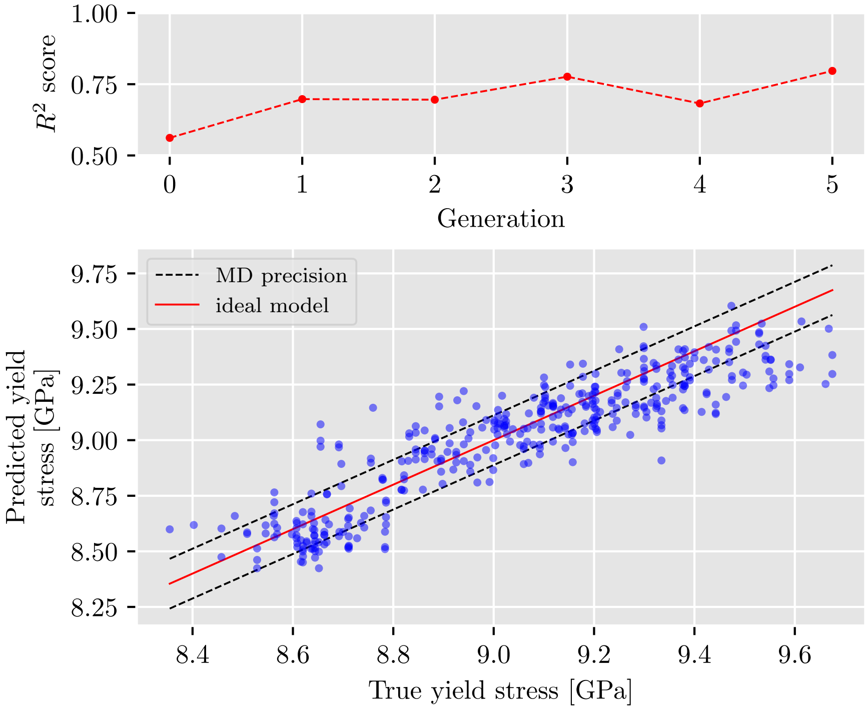


Figure 10.3: (Top) R^2 score on the test data as a function of generations when using the convolutional neural network to search for weak geometries in the reduced design space. (Bottom) Predicted yield stress as a function of true yield stress in the final generation.

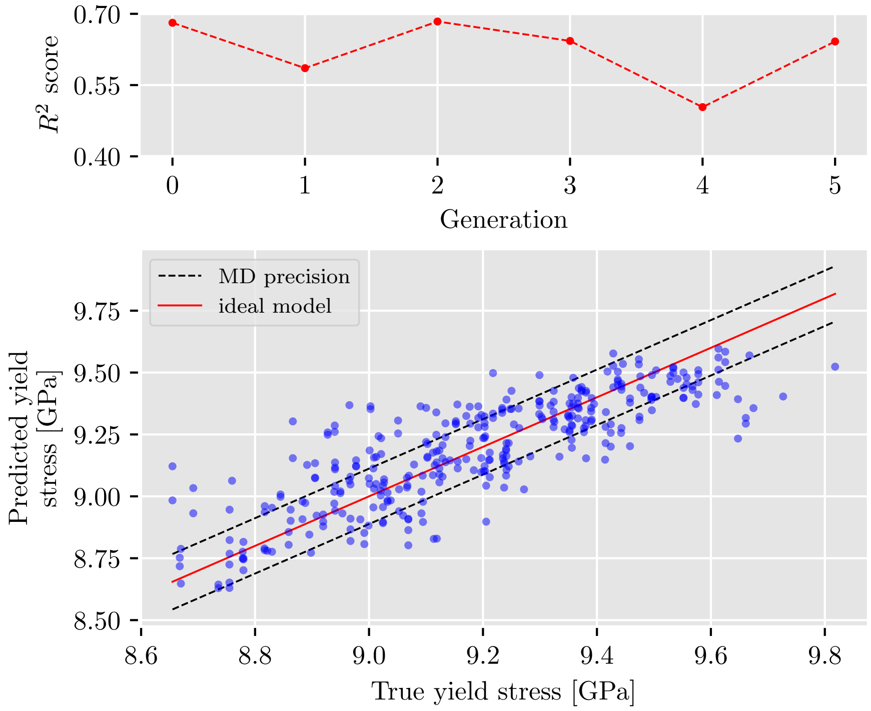


Figure 10.4: (Top) R^2 score on the test data as a function of generations when using the convolutional neural network to search for geometries with a target yield strength of 9.4 GPa. (Bottom) Predicted yield stress as a function of true yield stress in the 2nd generation.

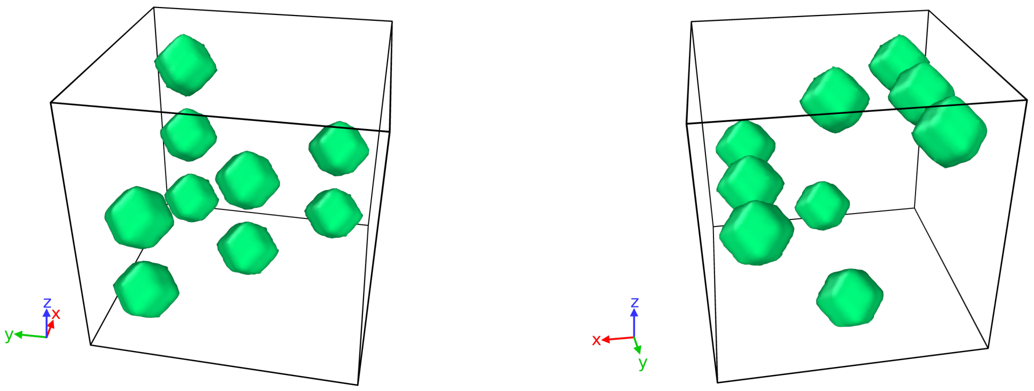


Figure 10.5: Illustrations of the pore configurations with the highest (left) and lowest (right) yield stresses, identified by the search algorithm in the reduced design space. They exhibit similar characteristics to those from the full design space (Figure 8.11), where the strong configuration favours pores which are stacked in the direction of the applied stress (z-axis), and the pores in the weak system are oriented diagonally at different levels of the z-axis.

```python
from itertools import combinations

num_pores = 9

#assign ids to each cell-coordinate
grid_ids = range(1, 28)

#generate all C(n,k) possible combinations
all_possible_configs = combinations(grid_ids, num_pores)

#set of discarded configurations identified as a duplicate
duplicate_configs = set()

#loop over all possible configurations and remove duplicates
for config in all_possible_configs:

    #skip if already checked
    if config in duplicate_configs:
        continue

    #shift coordinates and check periodic images
    for direction in it.product((-1, 0, 1), repeat=3):
        if direction == (0, 0, 0):
            continue
        periodic_config = check_periodic(direction, config)
        if periodic_config not in duplicate_configs:
            duplicate_configs.add(periodic_config)

    #rotate system horizontally
    for degrees in (0, 90, 180, 270):
        rotated_config = rotate_system(config, degrees)
        if rotated_config not in duplicate_configs and degrees != 0:
            duplicate_configs.add(rotated_config)

        #flip system vertically
        for axis in ('x', 'y'):
            flipped_config = flip_system(rotated_config, axis)
            if flipped_config not in duplicate_configs:
                duplicate_configs.add(flipped_config)

distinct_configs = all_possible_configs - duplicate_configs
```

Listing 3: Pseudo-code for generating distinct configurations of k pores in the binary 3^3 system of pores.

Bibliography

[1] Brian Gallagher, Matthew Rever, Donald Loveland, T Nathan Mundhenk, Brock Beauchamp, Emily Robertson, Golam G Jaman, Anna M Hiszpanski, and T Yong-Jin Han. Predicting compressive strength of consolidated molecular solids using computer vision and deep learning. *Materials & Design*, 190:108541, 2020.

[2] Ghanshyam Pilania, Chenchen Wang, Xun Jiang, Sanguthevar Rajasekaran, and Ramamurthy Ramprasad. Accelerating materials property predictions using machine learning. *Scientific reports*, 3(1):1–6, 2013.

[3] Chengxiong Zou, Jinshan Li, William Yi Wang, Ying Zhang, Deye Lin, Ruihao Yuan, Xiaodan Wang, Bin Tang, Jun Wang, Xingyu Gao, et al. Integrating data mining and machine learning to discover high-strength ductile titanium alloys. *Acta Materialia*, 202:211–221, 2021.

[4] Paul Z Hanakata, Ekin D Cubuk, David K Campbell, and Harold S Park. Accelerated search and design of stretchable graphene kirigami using machine learning. *Physical review letters*, 121(25):255304, 2018.

[5] Josiah W Gibbs. *The collected works of J. Willard Gibbs.* Yale University Press, 1948. Google-Books-ID: SvQyAQAAIAAJ.

[6] William W Mullins and Gregory S Rohrer. Nucleation barrier for volume-conserving shape changes of faceted crystals. *Journal of the American Ceramic Society*, 83(1):214–16, 2000.

[7] Clemens Burda, Xiaobo Chen, Radha Narayanan, and Mostafa A El-Sayed. Chemistry and properties of nanocrystals of different shapes. *Chemical reviews*, 105(4):1025–1102, 2005.

[8] Henrik A Sveinsson, Anders Hafreager, Rajiv K Kalia, Aiichiro Nakano, Priya Vashishta, and Anders Malthe-Sørenssen. Direct atomic simulations of facet formation and equilibrium shapes of sic nanoparticles. *Crystal Growth & Design*, 20(4):2147–2152, 2020.

[9] Minoru Yoshida, Akifumi Onodera, Masaki Ueno, Kenichi Takemura, and Osamu Shimomura. Pressure-induced phase transition in sic. *Physical Review B*, 48(14):10587, 1993.

[10] RI Scace and GA Slack. Silicon carbide - a high temperature semiconductor. *Ed. JR O'Connor and J. Smiltens, Pergamon Press, NY, p24*, 1960.

[11] Jaewoon Jung, Wataru Nishima, Marcus Daniels, Gavin Bascom, Chigusa Kobayashi, Adetokunbo Adedoyin, Michael Wall, Anna Lappala, Dominic Phillips, William Fischer,

Chang-Shung Tung, Tamar Schlick, Yuji Sugita, and Karissa Sanbonmatsu. Scaling molecular dynamics beyond 100,000 processor cores for large-scale biophysical simulations. *Journal of Computational Chemistry*, 40, 04 2019.

[12] George F. Becker and Arthur L. Day. The linear force of growing crystals. *Proceedings of the Washington Academy of Sciences*, 7:283–288, 1905.

[13] Luciano Oldecop and Eduardo Alonso. Modelling the degradation and swelling of clayey rocks bearing calcium-sulphate. *International Journal of Rock Mechanics and Mining Sciences*, 54:90–102, 2012.

[14] Rosa Espinosa-Marzal and George Scherer. Advances in understanding damage by salt crystallization. *Accounts of chemical research*, 43:897–905, 03 2010.

[15] Anja Røyne, Paul Meakin, Anders Malthe-Sørenssen, Bjørn Jamtveit, and Dag K Dysthe. Crack propagation driven by crystal growth. *EPL (Europhysics Letters)*, 96(2):24003, oct 2011.

[16] Julie Desarnaud, Daniel Bonn, and Noushine Shahidzadeh. The pressure induced by salt crystallization in confinement. *Scientific reports*, 6(1):1–8, 2016.

[17] Olivier Coussy. Deformation and stress from in-pore drying-induced crystallization of salt. *Journal of the Mechanics and Physics of Solids*, 54(8):1517–1547, 2006.

[18] Graham Williams and David C Watts. Non-symmetrical dielectric relaxation behaviour arising from a simple empirical decay function. *Transactions of the Faraday society*, 66:80–85, 1970.

[19] Geoffrey I Taylor. The mechanism of plastic deformation of crystals. part i.—theoretical. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 145(855):362–387, 1934.

[20] Markus J Buehler. *Atomistic modeling of materials failure.* Springer Science & Business Media, 2008.

[21] James R Rice and Robb Thomson. Ductile versus brittle behaviour of crystals. *The Philosophical Magazine: A Journal of Theoretical Experimental and Applied Physics*, 29(1):73–97, 1974.

[22] Jessica M Rimsza, Reese E Jones, and Louise J Criscenti. Crack propagation in silica from reactive classical molecular dynamics simulations. *Journal of the American Ceramic Society*, 101(4):1488–1499, 2018.

[23] S N Zhurkov. Kinetic concept of the strength of solids. *International Journal of Fracture Mechanics*, 1(4):311–323, 1965.

[24] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[25] Wikimedia Commons. Cauchy tensor, 2008. `https://commons.wikimedia.org/wiki/File:Tensore_Cauchy.png`.

[26] Henrik A Sveinsson and Anders Malthe-Sørenssen. Molecular-scale thermally activated fractures in methane hydrates: a molecular dynamics study. *Physical Chemistry Chemical Physics*, 21(25):13539–13544, 2019.

[27] Loïc Vanel, Sergio Ciliberto, Pierre-Philippe Cortet, and Stéphane Santucci. Time-dependent rupture and slow crack growth: elastic and viscoplastic dynamics. *Journal of Physics D: Applied Physics*, 42(21):214007, 2009.

[28] WF Busse, ET Lessig, DL Loughborough, and L Larrick. Fatigue of fabrics. *Journal of Applied Physics*, 13(11):715–724, 1942.

[29] Henry Eyring. Viscosity, plasticity, and diffusion as examples of absolute reaction rates. *The Journal of chemical physics*, 4(4):283–291, 1936.

[30] Frederick J Bueche. Tensile strength of plastics below the glass temperature. *Journal of Applied Physics*, 28(7):784–787, 1957.

[31] Mark D Ediger, C Austen Angell, and Sidney R Nagel. Supercooled liquids and glasses. *The journal of physical chemistry*, 100(31):13200–13212, 1996.

[32] Daniel Wüstner, Lukasz M Solanko, Frederik W Lund, Daniel Sage, Hans J Schroll, and Michael A Lomholt. Quantitative fluorescence loss in photobleaching for analysis of protein transport and aggregation. *BMC bioinformatics*, 13(1):1–21, 2012.

[33] Peter Grassberger and Itamar Procaccia. The long time properties of diffusion in a medium with static traps. *The Journal of Chemical Physics*, 77(12):6281–6284, 1982.

[34] Richard G Palmer, Daniel L Stein, Elihu Abrahams, and Philip Warren Anderson. Models of hierarchically constrained dynamics for glassy relaxation. *Physical Review Letters*, 53(10):958, 1984.

[35] John E Jones. On the determination of molecular fields.—ii. from the equation of state of a gas. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 106(738):463–477, 1924.

[36] Simon Stephan, Monika Thol, Jadran Vrabec, and Hans Hasse. Thermophysical properties of the lennard-jones fluid: Database and data assessment. *Journal of chemical information and modeling*, 59(10):4248–4265, 2019.

[37] PN Keating. Effect of invariance requirements on the elastic strain energy of crystals with application to the diamond structure. *Physical Review*, 145(2):637, 1966.

[38] Frank H Stillinger and Thomas A Weber. Computer simulation of local order in condensed phases of silicon. *Physical review B*, 31(8):5262, 1985.

[39] Priya Vashishta, Rajiv K Kalia, Aiichiro Nakano, and José Pedro 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.

[40] P Vashishta, Rajiv Kalia, José Rino, and Ingvar Ebbsjö. Interaction potential for sio-2: A molecular-dynamics study of structural correlations. *Physical review. B, Condensed matter*, 41:12197–12209, 07 1990.

[41] Paulo Sergio Branicio, José Pedro Rino, Chee Kwan Gan, and Hélio Tsuzuki. Interaction potential for indium phosphide: a molecular dynamics and first-principles study of the elastic constants, generalized stacking fault and surface energies. *Journal of Physics: Condensed Matter*, 21(9):095002, 2009.

[42] Loup Verlet. Computer "experiments" on classical fluids. i. thermodynamical properties of lennard-jones molecules. *Physical review*, 159(1):98, 1967.

[43] William C Swope, Hans C Andersen, Peter H Berens, and Kent 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.

[44] Aidan P Thompson, Steven J Plimpton, and William Mattson. General formulation of pressure and stress tensor for arbitrary many-body interaction potentials under periodic boundary conditions. *The Journal of chemical physics*, 131(15):154107, 2009.

[45] Hans C Andersen. Molecular dynamics simulations at constant pressure and/or temperature. *The Journal of chemical physics*, 72(4):2384–2393, 1980.

[46] CP Lowe. An alternative approach to dissipative particle dynamics. *EPL (Europhysics Letters)*, 47(2):145, 1999.

[47] Petri Nikunen, Mikko Karttunen, and Ilpo Vattulainen. How would you integrate the equations of motion in dissipative particle dynamics simulations? *Computer physics communications*, 153(3):407–423, 2003.

[48] David S Cerutti, Robert Duke, Peter L Freddolino, Hao Fan, and Terry P Lybrand. A vulnerability in popular molecular dynamics packages concerning langevin and andersen dynamics. *Journal of chemical theory and computation*, 4(10):1669–1680, 2008.

[49] Shuichi Nosé. A unified formulation of the constant temperature molecular dynamics methods. *The Journal of chemical physics*, 81(1):511–519, 1984.

[50] Philippe H Hünenberger. Thermostat algorithms for molecular dynamics simulations. *Advanced computer simulation*, pages 105–149, 2005.

[51] William G Hoover. Canonical dynamics: Equilibrium phase-space distributions. *Physical review A*, 31(3):1695, 1985.

[52] MJ Mandell. On the properties of a periodic fluid. *Journal of statistical physics*, 15(4):299–305, 1976.

[53] Daan Frenkel and Berend Smit. *Understanding molecular simulation: from algorithms to applications*, volume 1. Elsevier, 2001.

[54] Alexander Stukowski. Visualization and analysis of atomistic simulation data with OVITO-the Open Visualization Tool. *Modelling and simulation in materials science and engineering*, 18(1), 2010.

[55] H Morkoc, S Strite, GB Gao, ME Lin, B Sverdlov, and M Burns. Large-band-gap sic, iii-v nitride, and ii-vi znse-based semiconductor device technologies. *Journal of Applied physics*, 76(3):1363–1398, 1994.

[56] Alexander LeNail. Nn-svg: Publication-ready neural network architecture schematics. *Journal of Open Source Software*, 4(33):747, 2019.

[57] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.

[58] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[59] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[60] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[61] John O Rawlings, Sastry G Pantula, and David A Dickey. *Applied regression analysis*. Springer, 1998.

[62] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: Data mining, inference, and prediction*. Springer, 2017.

[63] Aurélien Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, 2019.

[64] Steve Plimpton. Fast parallel algorithms for short-range molecular dynamics. *Journal of computational physics*, 117(1):1–19, 1995. `https://www.lammps.org`.

[65] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[66] James M Haile. *Molecular dynamics simulation: Elementary methods*. Wiley, 1992.

[67] Alexander Stukowski. Computational analysis methods in atomistic modeling of crystals. *The Journal of The Minerals, Metals & Materials Society*, 66:399–407, 2014.

[68] Eric KK Abavare, Jun-Ichi Iwata, Abu Yaya, and Atsushi Oshiyama. Surface energy of si(110)- and 3c-sic(111)- terminated surfaces. *physica status solidi (b)*, 251(7):1408–1415, 2014.

[69] Kurt Binder and Walter Kob. *Glassy materials and disordered solids: An introduction to their statistical mechanics*. World scientific, 2005.

[70] Elijah N Borodin and Alexander E Mayer. A simple mechanical model for grain boundary sliding in nanocrystalline metals. *Materials Science and Engineering: A*, 532:245–248, 2012.

[71] Keith J Laidler. A glossary of terms used in chemical kinetics, including reaction dynamics (iupac recommendations 1996). *Pure and applied chemistry*, 68(1):149–192, 1996.

[72] Yuly V Milman, Svetlana I Chugunova, Irina V Goncharova, Tadeusz Chudoba, Witold Lojkowski, and William A Gooch. Temperature dependence of hardness in silicon–carbide ceramics with different porosity. *International Journal of Refractory Metals and Hard Materials*, 17(5):361–368, 1999.

[73] Li Li and Michel Aubertin. A general relationship between porosity and uniaxial strength of engineering materials. *Canadian Journal of Civil Engineering*, 30(4):644–658, 2003.

[74] Tsunenobu Kimoto and James A Cooper. *Fundamentals of silicon carbide technology: growth, characterization, devices and applications.* John Wiley & Sons, 2014.

[75] Les P Pook. Stress intensity factor expressions for regular crack arrays in pressurised cylinders. *Fatigue & Fracture of Engineering Materials & Structures*, 13(2):135–143, 1990.