An MD-continuum multi-scale model to simulate the ramp compression of an elastic-plastic material

Liam Douglas-Mann

Doctor of Philosophy

University of York
Physics
October 2022

Abstract

The simulation of ramp compression is a multi-scale problem; wave propagation occurs on significantly greater spatial and temporal scales than is tractable for atomistic simulation, but material behaviour - which is most effectively simulated atomistically - can play a significant role in wave propagation.

This thesis describes the development and use of a novel multi-scale methodology for simulating the ramp compression of an elastic-plastic crystalline solid. The first piece of work investigated the following assumption: for sufficiently low strain rates, the local ramp compression of a single Lagrangian element can be approximated in molecular dynamics (MD) by uniform, uniaxial compression (UUC). This assumption was analysed by comparing a piston-driven non-equilibrium MD simulation with a UUC MD simulation, both seeded with initial voids. For the tested ramp rise time of 100 ps, the assumption was found to be sound: the thermodynamic evolution of a Lagrangian element was effectively reproduced.

The second piece of work, inspired by the first, developed a multi-scale model to enable the simulation of ramp propagation for an arbitrary piston velocity profile. Three major components and their development are presented: 1) a one-dimensional hydrocode, 2) an MD data set of UUC simulations, and 3) a neural network, trained on the MD data set, to act as the equation of state/constitutive model for the hydrocode. The developed simulation tool was used to simulate a sample driven by a piston, ramped from rest to 1100 ms^{-1} in 100ps. Using voids as a plasticity parameter, the tool was found to agree well with an NEMD simulation of the same scenario but at a significantly reduced computational cost.

Contents

List of Figures

Acknowledgements

This work was supported by the Engineering and Physical Sciences Research Council [EP/L01663X/1.] and AWE Plc.

I would like to express utmost gratitude to my supervisor, Andrew Higginbotham, for his support and expertise throughout the oscillating experience of completing a PhD. Thanks are also due to Luca Antonelli for his friendship and support of this work. Thanks also to Caroline Lumsdon, Ashley Poole, and Andrew Comley. Thanks also to the members of the highly-exclusive Thesis Writing Club, Bob and Michel; writing would have been a far lonelier process without their solidarity and astute recommendations.

Enormous thanks to my family and friends for their loving attention and connection throughout the years. Erin, thanks for being in our lockdown bubble; tell Cleo I forgive her. Monica, merci pour ton amitié.

Finally, thank you Rose: for reasons manifold, overwhelming, and obvious.

Declaration

I declare that this thesis is a presentation of original work and I am the sole author. This work has not previously been presented for an award at this, or any other, University. All sources are acknowledged as References.

Chapter 1

Introduction

1.1 The study of materials in extreme conditions

Materials shaped by extreme conditions may seem like a niche area of study; in our everyday lives we are hopefully unlikely to encounter the extreme pressures and temperatures of a planetary core, or the strain rates resulting from a detonation or asteroid impact. And yet our understanding of material behaviour in these contexts underpins our understanding of exoplanet geophysics [5] [6] [7], the origin of celestial formations such as the Earth and Moon [8] [9] [10] [11], and could inform new avenues for material development and technology for a host of different applications. [12] [13] [14] [15] [16]

Our modern understanding of materials in extreme conditions owes thanks to a century of formal study. The earliest high pressure experiments, pioneered by Percy Bridgman using early versions of the 'Bridgman seal', attained pressures of 70 mPa [17]. He later developed the opposed anvil device, which was adopted and modified by General Electric to produce man-made diamonds in 1955 under pressures of ∼5500MPa. The modern version of these devices can almost reach pressures of a terapascal. [18]

Within modern high pressure physics, methods of compression can be categorised as static and dynamic. Diamond anvil cells (DACs) are the modern versions of opposed anvil devices mentioned above and are known as static compression techniques. They work by compressing material samples between two opposing diamonds, sufficiently slowly that the process is isothermal. DACs have aided

discoveries such as the insulating phase of sodium at 200 GPa [19] and the single-bonded cubic-gauche structure of nitrogen at 110 GPa. [20] Currently, the upper limit of pressure achieved during cold compression by a DAC is ∼1 TPa. [21] This is sufficient to reach the pressures of large planetary cores - Jupiter's core, for reference, is expected to be at approximately 700 GPa. [22]. However, DACs are also fundamentally limited by the stresses which the diamonds themselves can withstand. This upper limit is suggested to be ∼1.8 TPa by O'Bannon et al. [23] though they warn that this value is determined by extrapolation.

Reaching pressures far above what is achievable by DACs is one of the motivations behind the rise of dynamic compression techniques. [24] Dynamic compression refers to extremely fast, adiabatic compression. While a static compression experiment with a diamond anvil cell may take several minutes or hours to compress a piece of material, dynamic compression takes place on the order of picoseconds to microseconds depending on the method used. The material can be compressed with gas guns [25] [26] [27], lasers [28] [29] [30], or pulsed power devices [31] [32] [33]. The benefit of the latter two techniques is the availability of precise pulse shaping (though gas gun experiments can also achieve some degree of pulse-shaping through the design of the density gradients and surface geometries of the flyer plates) [34] [35] [36]. The ability to control and change the shape of a laser pulse facilitates the application of different dynamic loading techniques. Of particular interest for studying material structure is ramp compression, also called quasi-isentropic compression, which can reach the extreme pressures of planetary cores with limited heating of the material. [37] Unlike shock compression, which can cause swift melting, ramp compressed materials can maintain their solid state, facilitating the study of their structure and dynamics in extreme conditions.

Dynamic compression also opens up the exploration of another extreme beyond pressure: strain rate. This is perhaps its greatest contrast with static compression. Strain rate describes how quickly a material is compressed or deformed. At high strain rates, the material can be compressed at rates comparable with and exceeding

the speed of material dynamics, such as the onset of plasticity, the mechanics of phase transformations, and even the period of a phonon. This speed of compression can cause the material to enter transient states which are inaccessible at low strain rates.

The advancement of dynamic compression experiments means that the material response to dynamic events - such as a turbine engine failure, armour performance, or an interplanetary collision - can be modelled in the laboratory. Strain rate has been demonstrated to be a highly important parameter in studying phase changes [38] [39] and is evidently an important parameter to explore from both an engineering and a pure materials science perspective.

The structure of materials and the dynamics of material response to high strain rate can be captured by x-ray diffraction. [40] [41] In particular, experiments can be conducted at facilities which couple high power lasers with x-ray free electron lasers such as the Linac Coherent Light Source in California [42] and the European XFEL in Hamburg [43]. These types of facilities offer unprecedented degrees of time resolution, allowing the material response to high strain rates to be observed on the time scales on which they occur. Experiments of this type have observed the lattice response of shocked copper on picosecond time scales [44], the growth of crystalline stishovite from amorphous silica [45], the displacive $\alpha \rightarrow \omega$ phase transition of zirconium in real time [46], and a new metastable phase of Bismuth upon shock release [47].

It is therefore evident that dynamic compression is a rich area of study. However, the behaviour of most materials in these extreme conditions remains largely unexplored. In a practical sense, this is partially explained by the relatively recent advancement of laser pulse shaping capabilities coupled with high time-resolution x-ray diffraction diagnostics. Due to the complexity and scale of the necessary equipment, the experiments themselves are also rarely trivial to perform. In a scientific sense, the characterisation of a vast catalogue of material responses is of less priority than understanding the fundamental material processes from otherwise

well-studied materials. Since studying these high strain rate fundamentals continues to provide new insight, there is much left to explore.

For example, the study of compression pathway dependence of particular material structures could promote the discovery and retrieval of new materials. [48] Diamond is the classic example of a high-pressure structure that is metastable at room temperature, but other materials could also be developed in this manner. [49] For example, particular fervour surrounds the search for a room-temperature superconductor. [50] [51].

Further experiments taking advantage of laser pulse shaping technology and the high repetition rates of new facilities will assist a greater level of study and understanding of material behaviour in response to high strain rates and diverse compression pathways. To support this experimental effort, researchers must have access to a wide range of tools for simulation and computational modelling.

1.2 The role of simulations and multi-scale modelling

The experiments described above could not take place without rigorous planning; nor could their results be easily interpreted without some comparison to a model. For example, understanding wave propagation through material is highly important when making fundamental decisions about an experiment's design, such as the pulse shape, intensity, and duration, as well as the temporal and spatial regions of interest at which to direct diagnostics. For this reason, hydrocodes are used to simulate the propagation of the wave. These are continuum models which reference a particular equation of state for the material and calculate the evolution of its state variables and sound speed.

The accuracy of a hydrocode simulation depends on the suitability of a particular equation of state and strength model to a particular compression scenario. These can be analytically derived or empirically developed (and often are a combination) and can be successful at reproducing a particular wave profile [52] [53]. However, for

the sake of predictive capability, the effects of kinetics, strain rate and compression history should be included in simulation techniques from the beginning, not as a retrospective correction.

Molecular dynamics (MD) simulates the motion and interaction of individual atoms; it therefore intrinsically captures these effects. MD can be made more accurate still by coupling to density functional theory (DFT) in the form of *ab initio* MD. However, in most contexts it is not feasible to simply replace hydrocodes using direct simulation from first principles; due to the computational cost of simulating several million atoms, the spatial limit of large MD simulations is roughly a micron. DFT is limited to system sizes on the order of a thousand atoms. Meanwhile, ramp compression experiments are typically at least several microns in length with rise times of at least several nanoseconds. [54] [55] [56] [32] [57] Therefore some approximations and abstractions are necessary to make the problem tractable.

This is the problem that multi-scale models aim to address. As their name suggests, they transfer relevant information from small-scale simulations to large-scale simulations. This is a common strategy in the modelling of materials to connect the atomistic scale to the continuum scale, thereby achieving a balance between accuracy and efficiency. [58] An example in the area of dynamic compression at high strain rates is the Lawrence Livermore multi-scale strength (LLMS) model, which transfers information from DFT to MD to dislocation dynamics to continuum hydrodynamics. [59]

Much like purely analytic models, the possible forms of multi-scale models are diverse. The LLMS model uses information from atomistic simulations to inform the physically-derived analytic functions which form the constitutive model. Other multi-scale models take advantage of machine learning techniques, such as neural networks, which train on data retrieved from atomistic simulations and then are used concurrently with continuum simulations. [60] An example of this strategy in the context of dynamic compression is the work by Mayer et al. [61] to model the shock compression of elastic-plastic solids, using an analytic plasticity model

informed by a neural network trained on data from MD simulations. [62]

The work presented in this thesis contributes another methodology following similar principles, and is distinct in its more 'equation-free' approach: there is no intermediary analytic plasticity model. The trained neural network acts as both equation of state and constitutive model; hydrostatic and strength effects are not treated separately. The particular application of interest is ramp compression; since ramp compression typically takes place on greater temporal and spatial scales than shock compression, there is all more reason to employ multi-scale modelling. We demonstrate the potential of this technique in the case of the elastic-plastic ramp compression of single crystal copper.

1.3 Thesis structure and role of the author

Chapter 2 contains the relevant background theory to this thesis. This includes (a) elementary materials science, (b) continuum modelling, (c) molecular dynamics and (d) artificial neural networks. Chapter 3 and 4 contain the original work produced by the author, though may reference some external work from the wider field to provide context; where this is the case, it will be clearly noted. All molecular dynamics simulations were conducted by the author using the code LAMMPS [63] and run on the Viking computer cluster at the University of York. A code written by Dr. Andrew Higginbotham was used to generate spatial profiles of molecular dynamics simulations. Chapter 3 contains dislocation analysis conducted by the author using the dislocation extraction algorithm (DXA) by Stukowski et al. [64] using the software OVITO [65]. Chapter 4 contains simulations using a hydrocode. This hydrocode was originally written in C++ by Dr. Andrew Higginbotham following the Wilkins' formulation [66] as detailed in section 2.3.7. The author is responsible for editing the hydrocode to follow the new formulation as detailed in Chapter 4 and to permit the simulation of ramp waves. The neural network was developed by the author using the Python library Keras [67] and was integrated with the hydrocode using the software frugally-deep. [68]

The work presented in chapters 3 and 4 is currently in preparation for submission to the Journal of Computational Physics. Additional work carried out during the PhD, but not detailed in this thesis, can be found in:

"Exploring the Impact of Driver Adherence to Speed Limits and the Interdependence of Roadside Collisions in an Urban Environment: An Agent-Based Modelling Approach" by Sedar Olmez, Liam Douglas-Mann, Ed Manley, Keiran Suchak, Alison Heppenstall, Dan Birks, and Annabel Whipp. Applied Sciences (2021)

Chapter 2

Background

2.1 Crystalline solids

A crystal is a periodic arrangement of atoms displaying long-range order. Its structure is usually described in terms of a lattice and a basis (or a motif.) We will first discuss the mathematical concept of a lattice, and how different lattice types can be described through particular unit cells. We will then introduce the basis and show how to notationally construct a crystal by combining it with a lattice.

2.1.1 The lattice

A lattice is an infinite array of regularly spaced points which conserves translational symmetry. The local environment of each point will therefore be identical. The distance between neighbouring points is called a lattice constant. A one dimensional lattice will have a single lattice constant a; a three-dimensional lattice will have three lattice constants, which are the lengths of its basis vectors: $a_1 = |\boldsymbol{a_1}|$, $a_2 = |\boldsymbol{a_2}|$, and $a_3 = |\boldsymbol{a_3}|$. A three-dimensional lattice can be represented as

$$\boldsymbol{T} = u\boldsymbol{a_1} + v\boldsymbol{a_2} + w\boldsymbol{a_3} \tag{2.1}$$

where u, v and w are integers. The angles between the basis vectors are denoted α_{12}, α_{23} and α_{31} for the angles between $\boldsymbol{a_1}$ and $\boldsymbol{a_2}$, $\boldsymbol{a_2}$ and $\boldsymbol{a_3}$, and $\boldsymbol{a_3}$ and $\boldsymbol{a_1}$, respectively. In three dimensions, only certain relationships between the lattice constants and angles can preserve translational symmetry. There are seven of these relationships and each one denotes a particular crystal system. Within a crystal

system, there may be several possible lattice types; for example, the cubic crystal system, in which $a_1 = a_2 = a_3$ and $a_{12} = a_{23} = a_{31} = 90°$, contains the simple cubic, the face-centred cubic, and the body-centred cubic. In total, across the seven classes, there are 14 possible lattice types. These are referred to as Bravais lattices.

2.1.2 The reciprocal lattice

Of equal physical importance to the concept of the lattice is the reciprocal lattice, for reasons given in section 2.1.7. It can be defined similarly to the non-reciprocal lattice:

$$\boldsymbol{G} = h\boldsymbol{b_1} + k\boldsymbol{b_2} + l\boldsymbol{b_3} \tag{2.2}$$

where the h, k and l are integers, and where the basis vectors $\boldsymbol{b_1}$, $\boldsymbol{b_2}$ and $\boldsymbol{b_3}$ are defined as:

$$\boldsymbol{b_1} = 2\pi\frac{\boldsymbol{a_2} \times \boldsymbol{a_3}}{\boldsymbol{a_1} \cdot \boldsymbol{a_2} \times \boldsymbol{a_3}} \qquad \boldsymbol{b_2} = 2\pi\frac{\boldsymbol{a_3} \times \boldsymbol{a_1}}{\boldsymbol{a_1} \cdot \boldsymbol{a_2} \times \boldsymbol{a_3}} \qquad \boldsymbol{b_3} = 2\pi\frac{\boldsymbol{a_1} \times \boldsymbol{a_2}}{\boldsymbol{a_1} \cdot \boldsymbol{a_2} \times \boldsymbol{a_3}} \tag{2.3}$$

where $\boldsymbol{a_1} \cdot \boldsymbol{a_2} \times \boldsymbol{a_3}$ is the volume of the unit cell defined by these vectors. If the basis vectors are orthogonal, as is the case for the cubic lattice, then we can see that the dot product between the associated basis vector of the direct lattice and the reciprocal lattice are $\boldsymbol{a_i} \cdot \boldsymbol{b_i} = a_i b_i = 2\pi$. This means that we can define the length of the reciprocal lattice vector $b_i = \frac{2\pi}{a_i}$.

2.1.3 Unit cells

To describe different lattices, we use the concept of a unit cell. A unit cell is a small structure that, when repeated via translation in integer multiples of the lattice's basis vectors, recreates the entire structure. Three particular unit cell types are predominantly used. A primitive unit cell refers to the smallest repeating unit and contains only one lattice point; commonly it is constructed by touching lattice points at its vertices. The sections of these lattice points sum up to one lattice point in total. A particular type of primitive cell is the Wigner-Seitz cell. This also contains only one lattice point, but at the centre of the cell. The edges of the cell are located at the midpoints between the central lattice point and its neighbours. Finally, there

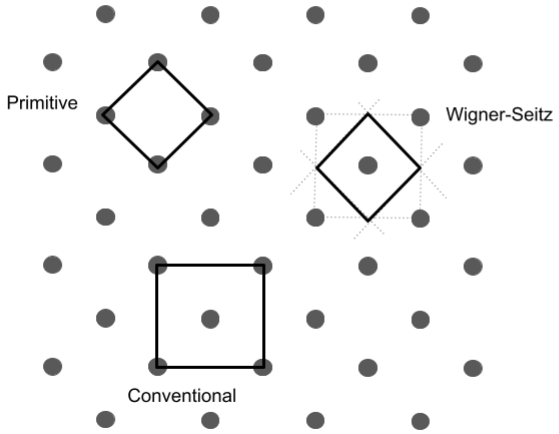

Figure 2.1: 2d lattice with a primitive cell, the Wigner-Seitz cell, and a conventional unit cell.

is the conventional unit cell. A conventional unit cell typically contains more than one lattice point in such a way that the symmetry of the lattice is made more visually apparent. As the name suggests, conventional unit cells are determined by convention rather than because they obey strict criteria. Examples of all three cell types for a 2d lattice are shown in figure 2.1. The conventional unit cells for the cubic lattice types - simple cubic, face-centered cubic and body-centered cubic - are shown in figure 2.4. Note that the conventional unit cell for a simple cubic is primitive.

The reciprocal lattice can also be described by a unit cell. This is referred to as the Brillouin zone, and it is constructed in the same way as the Wigner-Seitz cell described above; the only difference is that a reciprocal lattice point, rather than a direct lattice point, is centred. Specifically this cell is referred to as the first Brillouin zone, and further Brillouin zones can be defined outwards in the areas which border the first zone. All of these zones have the same volume. A two-dimensional lattice with the first, second and third Brillouin zones constructed in shown in figure 2.2.

2.1.4 The basis

The basis is simply the atom, or collection of atoms, associated with each lattice point. They are placed at a certain offset (or none at all) from the lattice point.

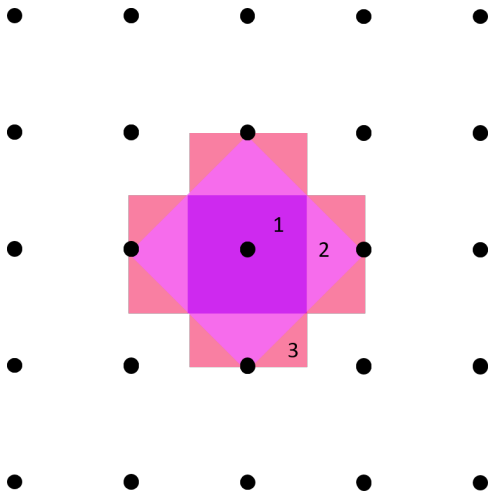

Figure 2.2: The first, second and third Brillouin zones of a 2d reciprocal lattice.

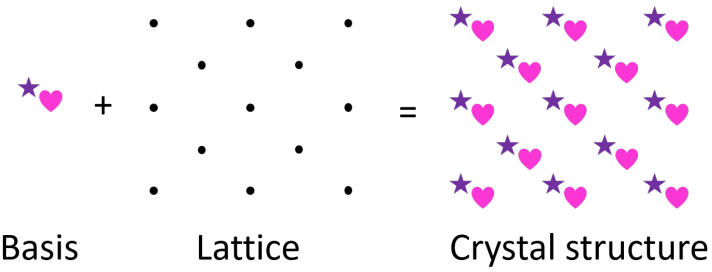

Figure 2.3: Construction of a crystal structure from a basis and lattice.

The lattice and basis together form a crystal structure, as shown in figure 2.3.

2.1.5 Crystallographic directions and planes

Shorthand notation composed of three numbers is used to describe crystallographic directions and planes in three dimensions. We can represent a particular direction as a straight line within the cell, whose projection on each axis (where the axes are aligned with the basis vectors; they needn't be Cartesian) is of a certain length x_0a_1, y_0a_2, and z_0a_3, where a_1, a_2 and a_3 are the lattice constants and x_0, y_0 and z_0 might be fractions or integers. A representation of this for a cubic cell is shown in figure 2.5. If x_0, y_0 and z_0 are fractions it is customary to rewrite them as integers by simply choosing the smallest combination of integers with the same ratio as these values. To indicate that it is a direction, these three numbers are written in square brackets. For example, lengths $x_0 = 0.75a_x$, $y_0 = 0.5a_y$ and $z_0 = a_z$ would be denoted by [3 2 4]. To denote a direction that is negative in one of the axis dimensions, a bar

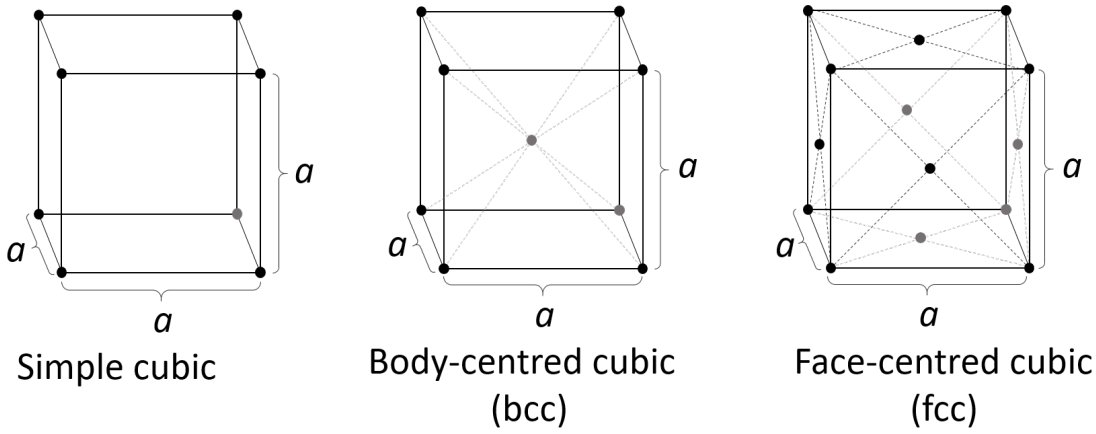

Figure 2.4: Simple cubic, body-centred cubic, and face-centred cubic lattices.

is written overhead for that value. For example, if y_0 and x_0 remained the same but $z_0 = -a_z$, the direction would be denoted $[3\,2\,\overline{4}]$. Due to crystal symmetry, certain groups of directions are identical. These groups are called 'families' and can be denoted by angular brackets: $\langle h\,k\,l\rangle$. For a cubic lattice, $\langle 1\,0\,0\rangle$ refers to $[1\,0\,0]$ $[0\,1\,0]$, $[0\,0\,1]$, $[\overline{1}\,0\,0]$, $[0\,\overline{1}\,0]$, and $[0\,0\,\overline{1}]$. Crystallographic planes are also represented by three numbers, which are referred to as Miller indices. If a plane intersects the axes at $\boldsymbol{a_1}/h$, $\boldsymbol{a_2}/k$, $\boldsymbol{a_3}/l$, the Miller indices are $(h\,k\,l)$. If a plane does not intersect a particular axis, its corresponding index is simply zero. Examples of planes with their associated Miller indices is shown in figure 2.6. Similarly to directions, families of symmetrically equivalent panes are represented using curly brackets: $\{h\,k\,l\}$.

Note that Miller indices repeat the $\{h\,k\,l\}$ notation used in the definition of the reciprocal lattice vector in equation 2.2. We can understand the link between planes and the reciprocal lattice by finding an expression for the plane normal. This is done by taking the cross product of two vectors that lie in the plane (themselves found by simply taking differences of two intercepts):

$$\begin{aligned}\hat{\boldsymbol{n}} &= (\boldsymbol{a_1}/h - \boldsymbol{a_2}/k) \times (\boldsymbol{a_1}/h - \boldsymbol{a_3}/l)) \\ &= \frac{\boldsymbol{a_1}\times\boldsymbol{a_1}}{h^2} - \frac{\boldsymbol{a_1}\times\boldsymbol{a_3}}{hl} - \frac{\boldsymbol{a_2}\times\boldsymbol{a_1}}{kh} + \frac{\boldsymbol{a_2}\times\boldsymbol{a_3}}{kl} \\ &= \frac{\boldsymbol{a_1}\times\boldsymbol{a_2}}{hk} + \frac{\boldsymbol{a_2}\times\boldsymbol{a_3}}{kl} + \frac{\boldsymbol{a_3}\times\boldsymbol{a_1}}{hl}\end{aligned}$$

It can be seen that the direction of this normal vector is parallel to the reciprocal

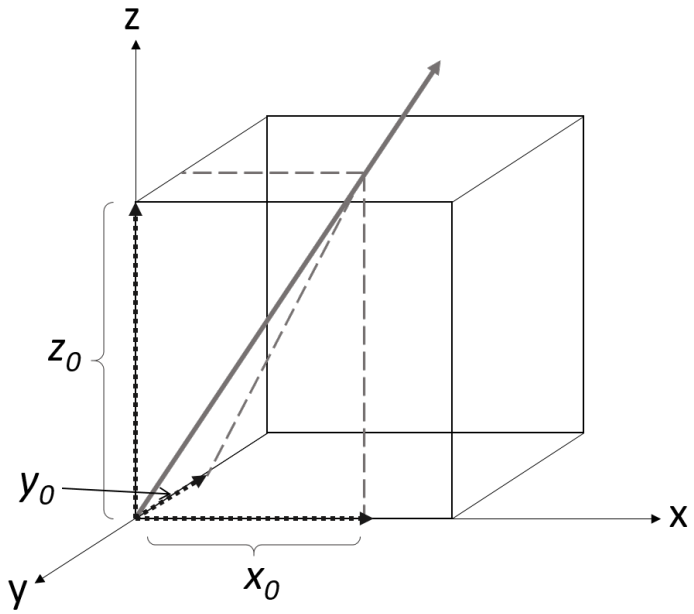

Figure 2.5: Graphical construction to determine a crystallographic direction from its projections on the x, y and z axes.

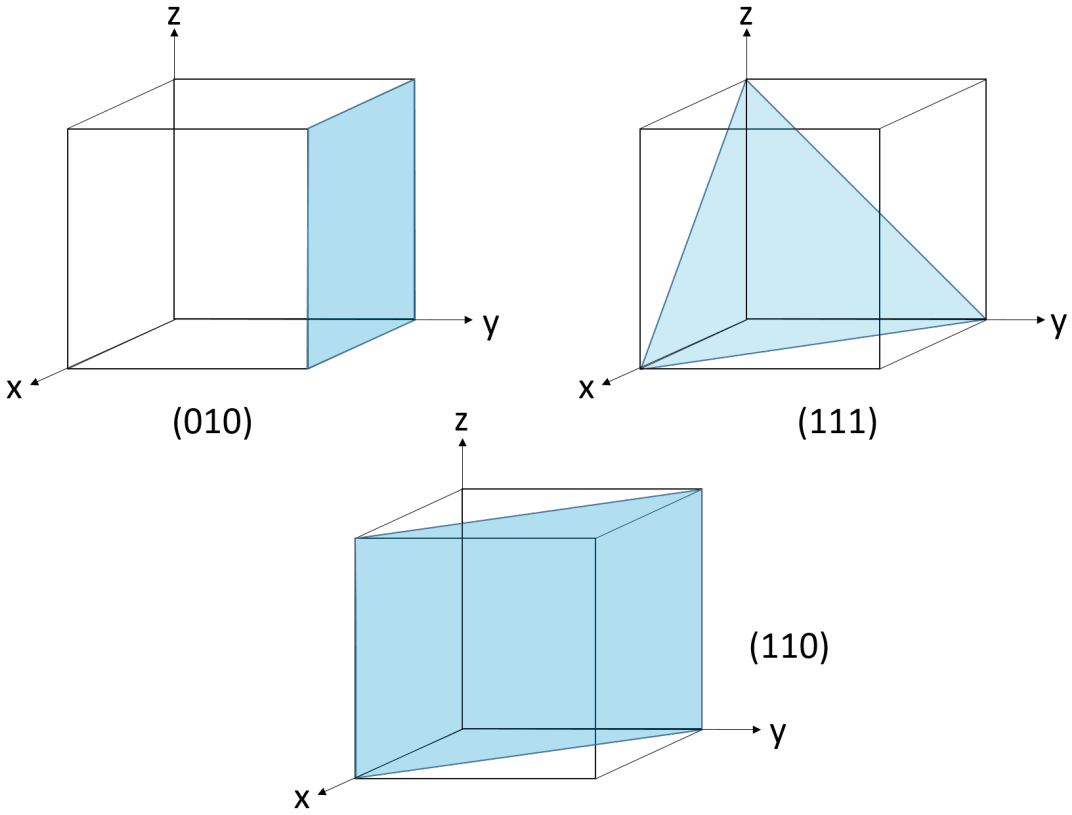

Figure 2.6: The (0 1 0) plane, the (1 1 1) plane, and the ((1 1 0) plane in a cubic lattice.

lattice vector $\boldsymbol{G}$ in equation 2.2 (whose components are given in equation 2.3. The length of the reciprocal lattice vector can be found to be equal to the distance between two planes.

Crystallographic directions and planes can influence deformation mechanisms. For example, the Bain distortion relates the fcc lattice to the bcc lattice when it is compressed along $\langle 1\,0\,0 \rangle$. [69] This relationship can be seen in figure 2.7; within the fcc lattice, there exists a body-centred tetragonal lattice (bct). Tetragonal lattices are another crystal system which are simply square cuboids: $a_1 = a_2 \neq a_3$ and $\alpha_{12} = \alpha_{23} = \alpha_{32}$. If the fcc lattice is compressed in the $\hat{z}$ direction by a factor of $\frac{1}{\sqrt{2}}$, with no corresponding expansion in the $\hat{x}$ and $\hat{y}$ dimensions, the bct lattice becomes bcc. In a volume conserving transformation from fcc to bcc, compression in the $\hat{z}$ direction would be accompanied by expansion in the other dimensions such that the new bcc lattice constant would be related to the previous fcc lattice constant in the following way:

$$a_{bcc} = \frac{a_{fcc}}{\sqrt[3]{2}} \tag{2.4}$$

This relationship is responsible for the production of martensite, of body-centred tetragonal structure, from austenite, an fcc allotrope of iron. [70] Crystal orientation during compression can also affect the onset of diffusional phase transformations, such as melting [71] and crystal growth. [72] The defect types that are produced under plastic deformation are also governed by crystal orientation during compression [73].

2.1.6 Packing efficiency

Packing density refers to how densely different crystal lattices can pack atoms. We can quantify this by using a concept called packing efficiency, or packing fraction, where we assume the lattice points are occupied by single rigid spheres of equal sizes, with the size of each sphere limited only by contact with neighbouring spheres. The packing fraction is then defined as the ratio of the total volume of the spheres in a

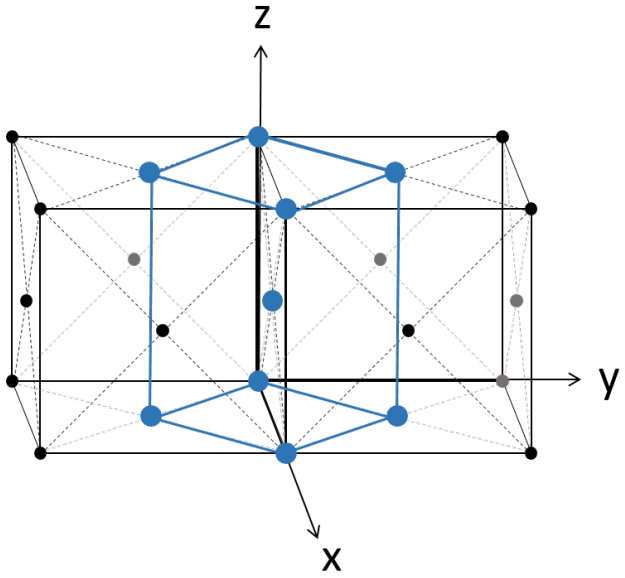

Figure 2.7: Fcc lattice with relationship to bct lattice highlighted. If compressed by a factor of $\frac{1}{\sqrt[3]{2}}$ along [1 0 0], with expansion in the other two directions to conserve volume, this becomes bcc.

unit cell to the volume of the unit cell itself.

$$P = \frac{V_{\text{Sphere}} N}{V_{\text{Unit cell}}} \tag{2.5}$$

where V_{Sphere} is the volume of a single sphere, N_{Spheres} is the number of spheres contained by the unit cell, and $V_{\text{Unit cell}}$ is the volume of the unit cell. For a cubic lattice, $V_{\text{Unit cell}} = a^3$, where a is the lattice constant. The volume of a sphere is $\frac{4}{3}\pi r^3$, where the relationship between r and a will be determined by the particular lattice type. For a simple cubic, $r_{sc} = \frac{a}{2}$, and $N_{Spheres} = 1$. The packing efficiency is therefore

$$P_{sc} = \frac{4\pi}{3}\frac{a^3}{8a^3} = \frac{\pi}{6} \approx 0.52 \tag{2.6}$$

For a body-centred cubic, $r_{bcc} = \frac{\sqrt{3}}{4}a$ and $N_{Spheres} = 2$. Therefore the packing efficiency is

$$P_{bcc} = \frac{4\pi}{3}\frac{2(\frac{\sqrt{3}}{4}a)^3}{a^3} = \frac{\sqrt{3}\pi}{8} \approx 0.68 \tag{2.7}$$

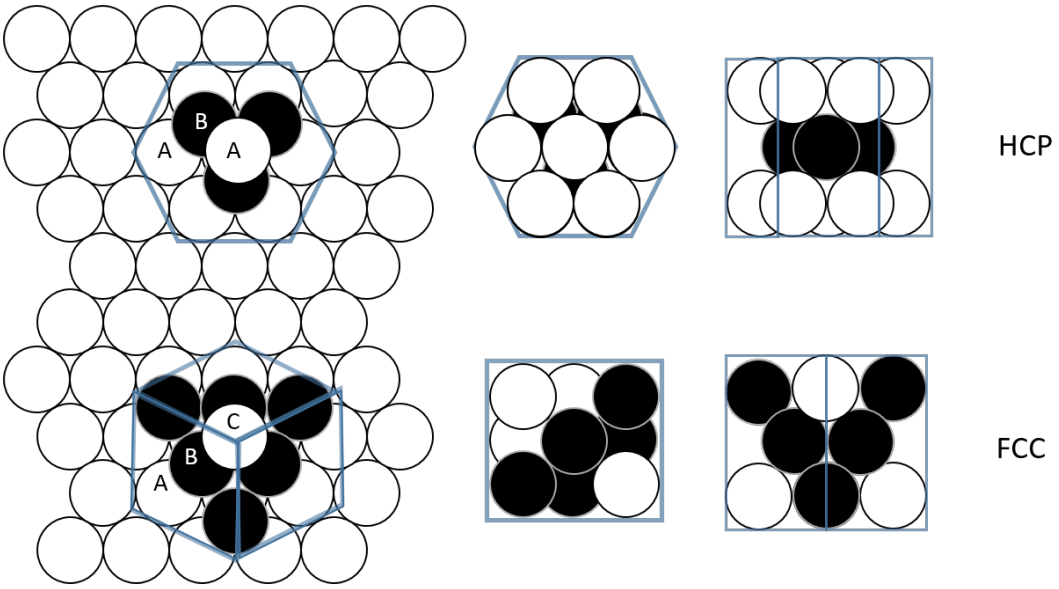

Figure 2.8: Stacking sequence ABA for hexagonal close packed (hcp) and ABCABC for face-centred cubic (fcc)

which is substantially more efficient than the simple cubic lattice. More dense again is fcc, for which $r = \frac{\sqrt{2}}{4}a$ and $N = 4$, so the packing efficiency is

$$P_{fcc} = \frac{4\pi}{3}\frac{4(2^{3/2}a^3)}{4^3a^3} = \frac{\sqrt{2}}{6}\pi \approx 0.74 \tag{2.8}$$

which is also the highest packing efficiency possible in nature. For this reason, fcc is also referred to as cubic close-packed. The other close-packed structure is hexagonal close-packed (hcp), which achieves the same packing density as fcc. This is because, like fcc, it is constructed from layers of the close-packed plane, shown in figure 2.8. It differs from fcc only in the order in which the layers are stacked upon each other. As shown in figure 2.8, there are two ways that the close-packed plane can be stacked: ABA, which is the stacking order for hcp, and ABCABC, which is the stacking order for fcc.

2.1.7 Lattice vibrations

Atoms in a solid oscillate around their equilibrium position. There are certain allowed vibrational modes which are determined by the number of atoms and the spacing between them. To explain this we can consider a one-dimensional chain of atoms in a circle (which is equivalent to a periodic boundary - this allows us to

ignore edge effects.) We can see that the allowed modes can be described by

$$Na = p\lambda \tag{2.9}$$

where N is the number of atoms, a is the spacing between atoms, λ is the wavelength and p is an integer. This can be described in terms of reciprocal space to obtain the wave number k

$$k = \frac{2\pi}{\lambda} = \frac{2\pi p}{Na} \tag{2.10}$$

The spacing between modes is therefore simply $k = \frac{2\pi p}{a}$; in other words, there are N unique modes in an interval of $\frac{2\pi}{a}$. It is conventional to place this interval at $-\pi/a < k \leq \pi/a$, the first Brillouin zone. For a regularly spaced grid, this equally holds true in two and three dimensions.

2.1.8 Defects

Defects are discontinuities in the perfect crystal structure. There are a range of different possible defects, and can generally be categorised into point defects, line defects, planar defects, and bulk defects.

Point defects refer to interstitial defects, substitutional defects, and vacancies. Interstitials are extra atoms that occupy sites that should be empty. They can either be of the same type as the rest of the crystal, or be of a different species. Substitutional defects refer to atoms of a different type that replace another atom. Vacancies refer to a missing atom from a site that should be occupied.

Line defects largely refer to dislocations, which will be further described in a dedicated section due to their importance to plastic deformation (which will also be discussed later on.) Some dislocations can move through a material through a mechanism known as glide - these are called 'glissile'. Dislocations which cannot glide are called 'sessile.' Dislocations form and move in response to shear stress in order to relieve it.

Planar defects generally refer to grain boundaries and stacking faults. Polycrystalline materials are made up of 'grains' - small crystals in different random

orientations. Where they meet, both grains' perfect structure is interrupted. A special type of grain boundary is referred to as twinning. This is where the crystals which meet at a boundary are mirror images of the other. Twinning can occur in three ways: during crystal growth, from a polymorphic phase transition during cooling, or in response to shear stress during plastic deformation.

As mentioned earlier in section 2.1, a stacking sequence refers to the order in which layers of atoms are placed atop each other. A stacking fault refers to an error or interruption in this sequence. As mentioned previously, the stacking sequence of fcc is ABCABC. A sequence of ABCABABC in fcc describes a stacking fault.

Bulk defects refer to macroscopic, three-dimensional abnormalities such as cracks and voids.

2.1.9 Stress and strain

Stress is the resistance force per unit area of a material to an applied force per unit area acting on an object. It can be described by a tensor

$$\sigma = \begin{pmatrix} \sigma_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_{zz} \end{pmatrix} \tag{2.11}$$

where the diagonal components σ represent the normal stresses (normal to the surface of the object) and the off-diagonal components τ represent shear stresses (parallel to the surface of the object.) These components are represented in figure 2.9. Note that the stresses are defined in such a way that an object subject to a compressive force experiences negative stress, while an object under tension experiences positive stress. This reflects the idea that stress is a reactive property of a material, and is therefore opposite in sign to a force enacting upon that material. For this reason, pressure is defined as the negative of the average of the normal stresses:

$$-P = \frac{\sigma_{xx} + \sigma_{yy} + \sigma_{zz}}{3} \tag{2.12}$$

so compression results in a positive pressure.

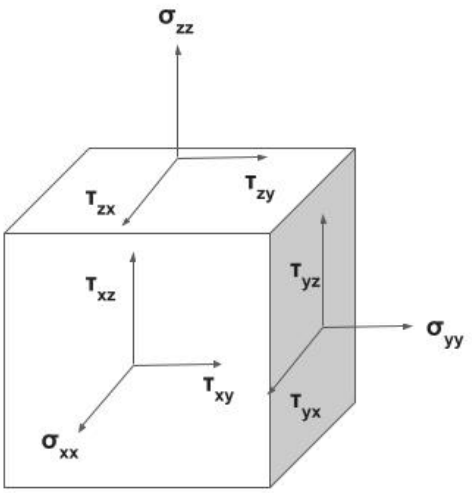

Figure 2.9: Representation of the stress tensor for an object. Normal stresses are given by σ and shear stresses are given by τ.

The stress tensor is symmetric, meaning that $\tau_{ij} = \tau_{ji}$. By rotating the coordinate axes, it is possible to express the stress tensor in such a way that the off-diagonal components are zero; this can be helpful because rather than six independent components to describe the stress, we can use only three. These three stress components are called the principal stresses and they are the eigenvalues σ of the stress tensor, such that

$$\begin{vmatrix} \sigma_{xx} - \sigma & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_{yy} - \sigma & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_{zz} - \sigma \end{vmatrix} = 0 \tag{2.13}$$

This can be expanded to find the characteristic equation:

$$\sigma^3 - I_1\sigma^2 + I_2\sigma - I_3 = 0 \tag{2.14}$$

where I_1, I_2 and I_3 are the invariants of the stress tensor. The values of these do

not change regardless of the coordinate system used. They are

$$I_1 = \sigma_{xx} + \sigma_{yy} + \sigma_{zz} \tag{2.15}$$

$$I_2 = \sigma_{xx}\sigma_{yy} + \sigma_{xx}\sigma_{zz} + \sigma_{yy}\sigma_{zz} - \tau_{xy}^2 - \tau_{yz}^2 - \tau_{xz}^2 \tag{2.16}$$

$$I_3 = \begin{vmatrix} \sigma_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_{zz} \end{vmatrix} \tag{2.17}$$

Note that the pressure, defined in 2.12, is simply $\frac{-I_1}{3}$. This shows that pressure is invariant across different coordinate axes.

Equation 2.14 has three roots, denoted σ_1, σ_2 and σ_3. These are known as the principal stresses. These are the components of the principal stress tensor:

$$\sigma_P = \begin{pmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{pmatrix} \tag{2.18}$$

The stress tensor can also be written as the sum of the hydrostatic stress (simply the negative of the pressure) and the deviatoric stress:

$$\begin{pmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{pmatrix} = \begin{pmatrix} -P & 0 & 0 \\ 0 & -P & 0 \\ 0 & 0 & -P \end{pmatrix} + \begin{pmatrix} s_{xx} & s_{xy} & s_{xz} \\ s_{yx} & s_{yy} & s_{yz} \\ s_{zx} & s_{zy} & s_{zz} \end{pmatrix} \tag{2.19}$$

where the deviatoric stresses are simply defined as the difference between the principal stress and the hydrostatic stress: $s_{ij} = \sigma_{ij} - \frac{1}{3}\delta_{ij}\sum_k \sigma_{kk}$. The deviatoric stresses are responsible for changing the shape of an object, while the hydrostatic stress is responsible for changing its volume. The deviatoric and hydrostatic stress tensors are also symmetric.

Strain is the deformation that an object experiences. Similarly to stress, it can be expressed as a tensor

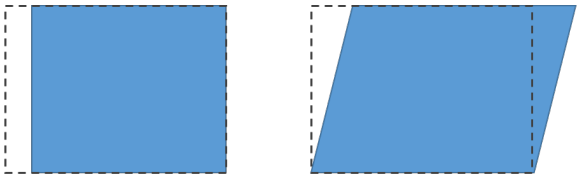

Figure 2.10: Normal deformation, which results in a change in volume, and shear deformation, which conserves volume.

$$\epsilon = \begin{pmatrix} \epsilon_{xx} & \gamma_{xy} & \gamma_{xz} \\ \gamma_{yx} & \epsilon_{yy} & \gamma_{yz} \\ \gamma_{zx} & \gamma_{zy} & \epsilon_{zz} \end{pmatrix} \tag{2.20}$$

where the diagonal components ϵ represent deformation along the normal axes and the off-diagonal components γ represent shear deformation.

As shown in figure 2.10, deformation along a normal axis results in a change in volume, while shear deformation does not. There are two common definitions of strain: the true strain and the engineering strain. The true strain is the sum of incremental changes in length as fractions of their actual, rather than original, length:

$$\epsilon = \int_{l_0}^{l} \frac{dL}{L} \tag{2.21}$$

where l is the length after deformation and l_0 is the original length. This becomes

$$\epsilon_t = \ln(\frac{l}{l_0}) = \left(\frac{l - l_0}{l_0}\right) - \left(\frac{(l - l_0)^2}{2l_0^2}\right) + \left(\frac{(l - l_0)^3}{3l_0^3}\right) + ... \tag{2.22}$$

which is the definition of true strain. The engineering strain, meanwhile, makes a linear approximation of the difference between the current length and the original length and ignores higher order terms:

$$\epsilon_e = \frac{l - l_0}{l_0} \tag{2.23}$$

The engineering strain rate is therefore an appropriate approximation only in the case of small strains.

Similarly to the stress tensor, the strain tensor is symmetric: $\epsilon_{ij} = \epsilon_{ji}$. Once again, this means it can be expressed in terms of principal strains through the same process as shown for stress. The first invariant of the strain tensor is known as the volumetric strain:

$$\frac{\Delta V}{V} = \epsilon_{xx} + \epsilon_{yy} + \epsilon_{zz} \tag{2.24}$$

which is the strain that describes volume change, but not shape change.

Furthermore, the strain tensor can be linearly separated into a hydrostatic strain tensor and a deviatoric strain tensor, defined in an analogous manner to the hydrostatic and deviatoric stress.

2.1.10 Elastic deformation

Elastic deformation is characterised by a linear relationship between stress σ exerted on a material and the strain ϵ, or deformation, that it experiences. In other words, it obeys Hooke's law. For any three-dimensional material, we can relate the stress tensor σ_{ij} to the strain tensor ϵ_{kl} by the stiffness tensor C_{ijkl}:

$$\sigma_{ij} = C_{ijkl}\epsilon_{kl} \tag{2.25}$$

This can be simplified to the one-dimensional case for a rod undergoing uniaxial compression or tension, where Hooke's law is expressed simply as

$$\sigma = E\epsilon \tag{2.26}$$

where E is a material-dependent constant referred to as Young's modulus. It can also be expressed for shear deformation:

$$\tau = \mu\gamma \tag{2.27}$$

where τ is the shear stress, γ is the shear strain and μ is a constant known as the shear modulus. If we make the assumption that the material is isotropic, we can derive another expression for three-dimensional Hooke's law. It is found experimentally that a uniaxial compressive stress will cause longitudinal compressive

strain in the same dimension of the form $\epsilon_{xx} = \frac{1}{E}\sigma_{xx}$ and corresponding tensile strains $\epsilon_{yy} = \epsilon_{zz} = \frac{\nu}{E}\sigma_{xx}$ in the transverse directions, where ν is a material dependent constant referred to as Poisson's ratio. A shear stress will only produce a shear strain in the same direction: $\gamma_{xy} = \frac{1}{\mu}\tau_{xy}$.

If we consider an isotropic cubic element under triaxial normal and shear stress we can assume that the material will behave identically in all directions. Furthermore, during elastic compression of a solid where the deformation is likely to be relatively small, we can assume that the strain contribution from each normal stress is independent. This allows us to express the normal strains as linear combinations. The normal and shear strains are therefore:

$$\epsilon_{xx} = \frac{1}{E}[\sigma_{xx} - \nu(\sigma_{yy} + \sigma_{zz}) \tag{2.28}$$

$$\epsilon_{yy} = \frac{1}{E}[\sigma_{yy} - \nu(\sigma_{xx} + \sigma_{zz}) \tag{2.29}$$

$$\epsilon_{zz} = \frac{1}{E}[\sigma_{zz} - \nu(\sigma_{xx} + \sigma_{yy}) \tag{2.30}$$

$$\gamma_{xy} = \frac{1}{\mu}\tau_{xy} \tag{2.31}$$

$$\gamma_{yz} = \frac{1}{\mu}\tau_{yz} \tag{2.32}$$

$$\gamma_{zx} = \frac{1}{\mu}\tau_{zx} \tag{2.33}$$

where the relationship between μ, E and ν is:

$$\mu = \frac{E}{2(1+\nu)} \tag{2.34}$$

We can express equations 2.28 to 2.33 more generally as:

$$\epsilon_{ij} = \frac{1+\nu}{E}\sigma_{ij} - \frac{\nu}{E}\delta_{ij}\sum_k \sigma_{kk} \tag{2.35}$$

where δ_{ij} is the Kronecker delta. This can be arranged to express stress in terms of strain. By multiplying both sides by δ_{ij} and simplifying we can find an expression for $\sum_k \sigma_{kk}$ in terms of strain:

$$\sum_k \sigma_{kk} = \frac{E\sum_k \epsilon_{kk}}{(1-2\nu)} \tag{2.36}$$

This can be substituted into equation 2.35, which can be rearranged to find σ_{ij}:

$$\sigma_{ij} = \frac{E}{(1+\nu)}\epsilon_{ij} + \frac{\nu E}{(1+\nu)(1-2\nu)}\delta_{ij}\sum_k \epsilon_{kk} \tag{2.37}$$

Equation 2.37 is the three-dimensional version of equation 2.26. By recalling that the hydrostatic stress is simply the average of the diagonal terms of the stress tensor, we can obtain the elastic relationship between hydrostatic stress and strain by multiplying equation 2.36 by $\delta_{ij}/3$:

$$\frac{\delta_{ij}\sum_k \epsilon_{kk}}{3} = (1-2\nu)\frac{\delta_{ij}\sum_k \sigma_{kk}}{3} \tag{2.38}$$

Recall from section 2.1.9 that the total stress can be written as a sum of the hydrostatic stress and the deviatoric stress. We can therefore write Hooke's law for the stress deviators s_{ij} by subtracting equation 2.38 from 2.35:

$$s_{ij} = \frac{E}{\nu+1}(\epsilon_{ij} - \frac{1}{3}\delta_{ij}\sum_k \epsilon_{kk}) \tag{2.39}$$

Elastic deformation is instantaneous and reversible. In an adiabatic system, elastic compression will result in a small temperature rise proportional to the work done in compression.

2.1.11 Yield strength

The yield strength (also called yield stress) refers to the threshold stress at which a material transitions from elastic to plastic deformation. In the following treatment, we assume that plastic deformation occurs similarly in tension and compression.

A particular criterion for yield in the isotropic case is the Von Mises criterion, which relates the yield strength Y_0 (an experimentally determined value) to the deviatoric stresses. The criterion satisfies two known requirements: (1) in uniaxial tension, one of the principal stresses must be equal to Y_0 while the others are zero, and (2) there is no yielding in hydrostatic compression where $\sigma_1 = \sigma_2 = \sigma_3$. We can consider this as defining a surface in principal stress space. Hydrostatic compression can be represented as a vector [111] - therefore the surface must separate this vector from the region where plastic yielding occurs. Equally, the surface must intersect

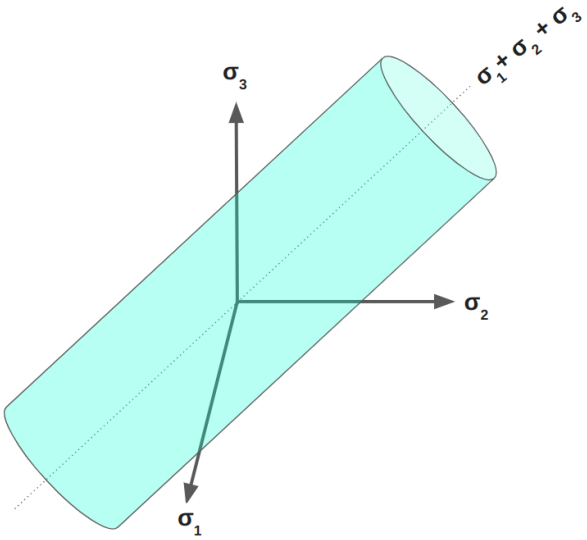

Figure 2.11: The Von Mises yield surface is a cylinder whose axis lies along the line $\sigma_1 = \sigma_2 = \sigma_3$ in principal stress space.

the points of $(Y_0, 0, 0)$, $(0, Y_0, 0)$ and $(0, 0, Y_0)$ (and the negatives of these.) This surface can be represented by the cylinder, as shown in figure 2.11 and expressed mathematically as:

$$(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2 = 2Y_0^2 \tag{2.40}$$

This is the Von Mises yield criterion in principal stress space. Recalling the definition of deviatoric stress $s_{ij} = \sigma_{ij} - \frac{1}{3}\delta_{ij}\sum_k \sigma_{kk}$, it can be noted that the second invariant of the deviatoric stress tensor is

$$J_2 = \frac{1}{6}\left[(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2\right] = \frac{1}{2}(s_1^2 + s_2^2 + s_3^3) \tag{2.41}$$

Therefore equation 2.40 can be expressed in terms of the principal deviatoric stresses

$$s_1^2 + s_2^2 + s_3^2 = \frac{2}{3}Y_0^2 \tag{2.42}$$

This also shows that the yield strength is invariant across coordinate axes.

2.1.12 Slip: a plastic deformation mechanism

Of course, rather than simply being due to a macroscopic condition, plasticity is driven by microscopic material behaviour. A major deformation mechanism seen in

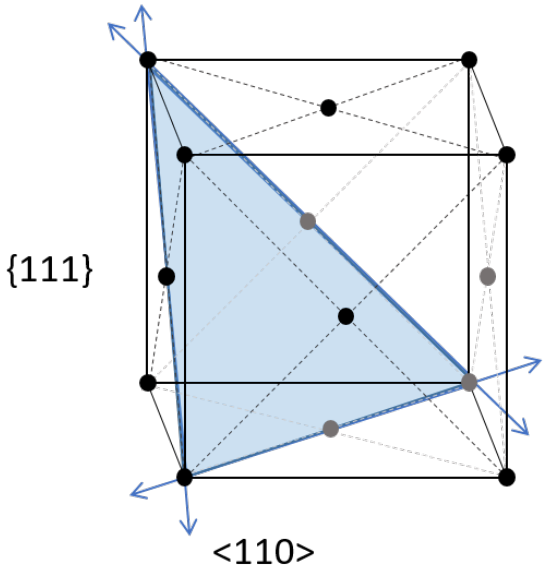

Figure 2.12: One of the $\langle 1\,1\,1 \rangle$ slip planes in fcc with its three directions shown.

crystalline solids is slip.

It occurs in response to a critical shear stress, causing a section of the material to shift. It only occurs on preferred crystallographic planes in preferred directions: the combination of a particular slip plane and slip direction is called a slip system. Different crystal structures have different slip systems, and a different number of slip systems. Usually, the slip direction is that in which the atoms are most densely packed, and the slip plane is the most densely packed plane containing that direction. This is because displaced atoms need not move as far to find a new point of equilibrium.

The most densely packed planes in FCC are the four $\{1\,1\,1\}$ planes. The most densely packed directions are $\langle 1\,1\,0 \rangle$. Each plane contains three $\langle 1\,1\,0 \rangle$ directions. This results in 12 slip systems in total. One of these planes and its three directions is shown in figure 2.12.

Hcp has only three slip systems, composed of the three closest-packed directions which lie on the basal plane. Bcc has 24 slip systems, composed of six $\{1\,1\,0\}$ planes with two $\langle 1\,1\,1 \rangle$ directions in each, and twelve $\{1\,1\,2\}$ planes with one $\langle 1\,1\,1 \rangle$ direction in each.

Although bcc crystals have the greatest number of slip systems, this does not

mean that they will more readily exhibit slip. In fact, slip is more likely to occur in close-packed crystal structures due to the simple fact that the energy required to move one atomic spacing is lower if that atomic spacing is smaller. This explains why bcc metals are typically brittle rather than ductile. However, between the two close-packed structures, slip is more likely to occur in fcc than hcp due to its greater number of slip systems. Accordingly, slip is the dominant plasticity mechanism in fcc metals.

The stress required for the onset of plastic deformation is called the yield stress (or yield strength). The yield stress is dependent on the angle that the stress direction makes with the slip plane on the primary slip system, because it is the shear stress on a slip plane that is responsible for the glide of dislocations. Since crystals have several slip systems, the primary slip plane is the first to reach what is called the critical resolved shear stress:

$$\tau = \sigma \cos\theta \cos\phi \tag{2.43}$$

where σ is the stress, ϕ is the angle between the stress direction and the normal to the slip plane, and θ is the angle between the stress direction and the slip direction. This is shown in figure 2.13.

Historically, evidence of slip was clearly visible from slip lines on a deformed material. While it was understood that this was caused by planes of atoms sliding over one another in response to shear stress, the theoretical value of necessary shear stress was far greater than the experimental value. This discrepancy was largely solved in 1934 by Orowan, Taylor and Polanyi, who separately but concurrently invented the concept of dislocations. To maintain a perfect crystal structure, slip must occur by the cooperative, simultaneous sliding of planes of atoms over one another. But if we allow interruptions to perfect periodicity, slip can occur over just a small portion of the plane at a time. The line separating the perfect-but-slipped crystal from the perfect-but-unslipped crystal is called a dislocation.

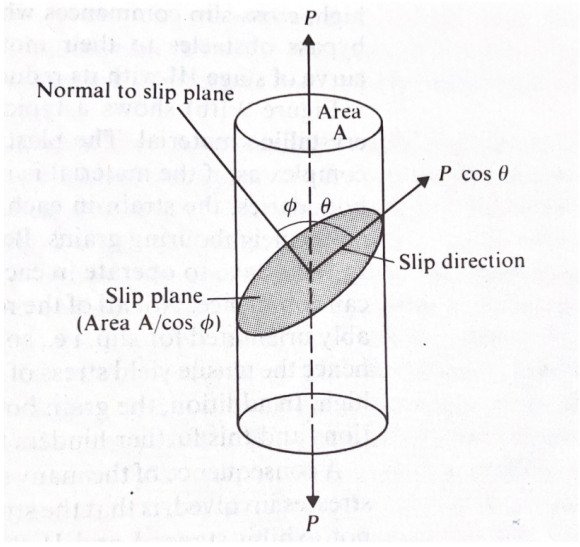

Figure 2.13: Components of shear stress on a slip plane during tensile loading. Reproduced from Anderson et al. [1] with permission from Springer Nature.

2.1.13 Dislocations

As mentioned in the earlier discussion of defects, dislocations are line defects whose creation and movement is responsible for plastic deformation. A dislocation is characterised by its Burgers vector **b**, which represents the magnitude and direction of the slip resulting from the movement of a dislocation. The direction of the Burgers vector, and whether it is perpendicular or parallel to the dislocation itself, categorises it as either an 'edge' or 'screw' dislocation, respectively. However, most dislocations are mixed, meaning they have components of both. Diagrams are given in figure 2.14.

When all atomic positions are known, a general method to obtain the magnitude and direction of a dislocation's Burgers vector is to draw a Burgers circuit. This involves drawing a closed circuit around the dislocation from atom to atom in a clockwise direction. Then, for a corresponding perfect crystal, an attempt to draw another circuit in the same direction with the same number of jumps between atoms will result in circuit that will not close. The vector needed to complete the circuit will be the Burgers vector of the dislocation. This is demonstrated in figure 2.15. The ability to draw a Burgers circuit serves as a general definition of a dislocation.

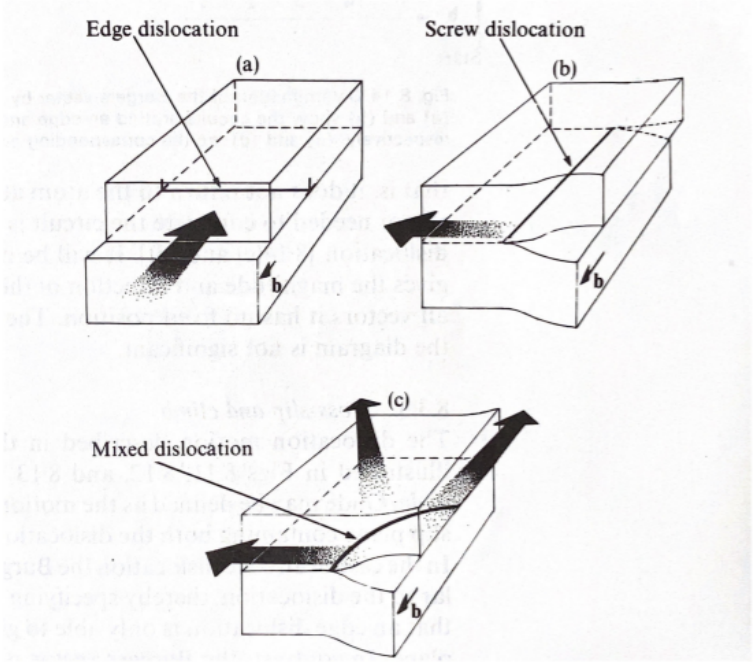

Figure 2.14: (a) An edge dislocation, (b) a screw dislocation, (c) a mixed dislocation. Reproduced from Anderson et al. [1] with permission from Springer Nature.

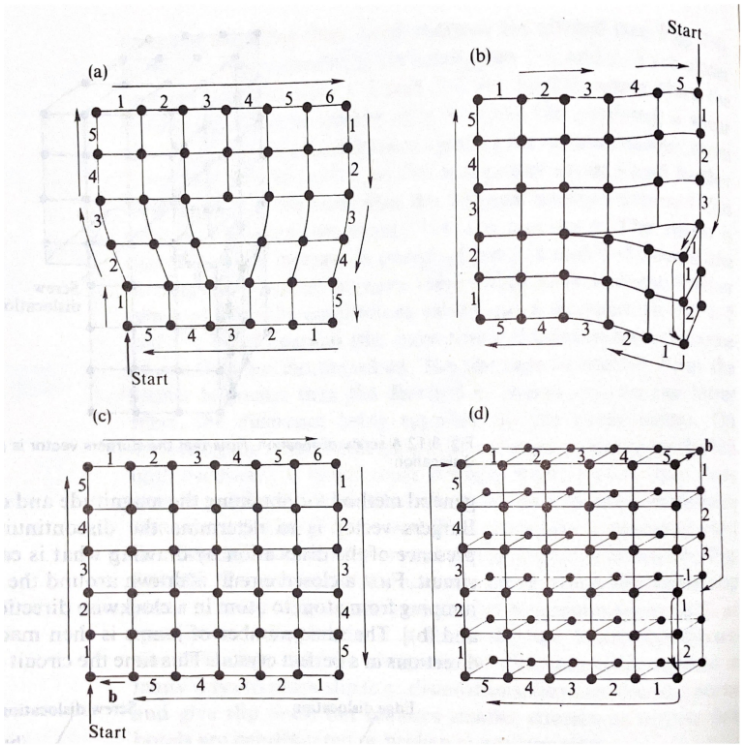

Figure 2.15: Determination of Burgers vector by drawing a Burgers circuit. (a) and (b) show circuits around an edge and screw dislocation respectively. (c) and (d) show the corresponding circuits in a perfect crystal. Reproduced from Anderson et al. [1] with permission from Springer Nature.

Different types of dislocations are characterised by their Burgers vector. A dislocation whose Burgers vector is equal to a multiple of the lattice vector is called a perfect dislocation. The glide of such a dislocation leaves behind a perfect crystal. A partial dislocation has a Burgers vector of less than one lattice vector; its glide leaves behind a stacking fault. In fcc crystals, there are two types of partial dislocations associated with single-plane stacking faults. One type is Shockley partials, which are glissile and therefore associated with slip. The other is a Frank partial, which is sessile. Some types of dislocations are formed by the intersection of other dislocations: for example, the stair-rod dislocation consists of two Shockley partials connected by a bent partial dislocation. Another example is the Hirth lock, which is formed when two perfect dislocations combine to form a stable, sessile segment. These types of dislocations serve to impede the movement of other dislocations along the slip planes. [74] [2]

Since dislocations interrupt the regularity of a lattice, and thus the minimum energy structure, dislocations represent a raising of the potential energies of the atoms who are displaced by its presence. For this reason we can say that a dislocation itself has an energy. Heterogeneous nucleation of dislocations from a pre-existing irregularity, such as a surface, grain boundary or another defect, requires less energy to be put into the system than homogeneous nucleation. This is because these irregularities, like dislocations themselves, have a higher energy than the minimum energy structure of a perfect crystal.

To understand the relationship between dislocation density, movement and plastic strain, we can derive and use the Orowan equation. We will apply the treatment by Hull and Bacon [2] which assumes movement by straight edge dislocations only, but the resulting relationship holds equally true for screw and mixed dislocations.

We first consider a volume hld as shown in figure 2.16, containing N straight edge dislocations. It is subject to a shear stress parallel to the Burgers vectors $\mathbf{b}$ of the dislocations. This causes a displacement of D of the top surface of the

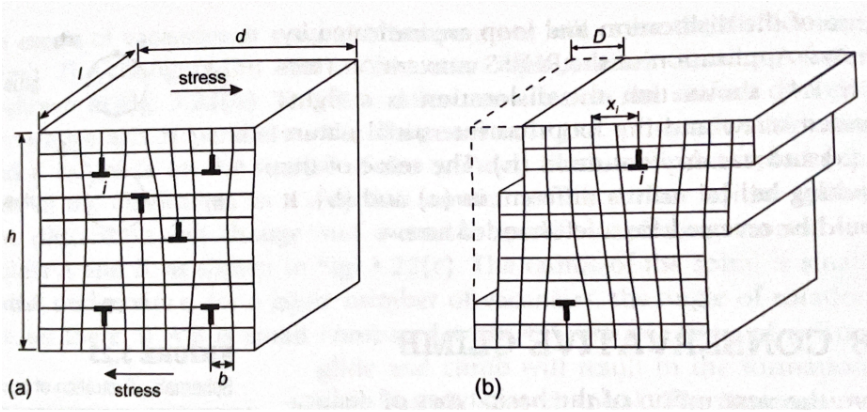

Figure 2.16: (a) Volume containing edge dislocations subject to a shear stress parallel to dislocations' Burgers vectors. (b) Slip caused by movement of dislocation i by distance x_i, resulting in a displacement of D of top surface relative to bottom surface. Reproduced from Hull and Bacon [2] with permission from Elsevier Science.

sample relative to the bottom surface. A dislocation that moves through the entire slip plane, covering a distance of d, contributes b to the overall displacement of D. If it actually moves x_i, it would contribute $\frac{bx_i}{d}$ to the total displacement. Adding contributions from all N dislocations, we can express D as

$$D = \frac{b}{d}\sum_{i=1}^{N} x_i \tag{2.44}$$

The macroscopic shear strain due to the movement of dislocations is then

$$\epsilon = \frac{D}{h} = \frac{b}{hd}\sum_{i=1}^{N} x_i \tag{2.45}$$

To simplify this, we can note that the average distance moved by a dislocation is

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{2.46}$$

and since the line density of moving straight edge dislocations is $\rho_m = Nl/lhd$ the plastic shear strain can thus be expressed:

$$\epsilon = b\rho_m\bar{x} \tag{2.47}$$

If we assume that b and ρ are constant, the plastic strain rate is

$$\dot{\epsilon} = b\rho_m\bar{v} \tag{2.48}$$

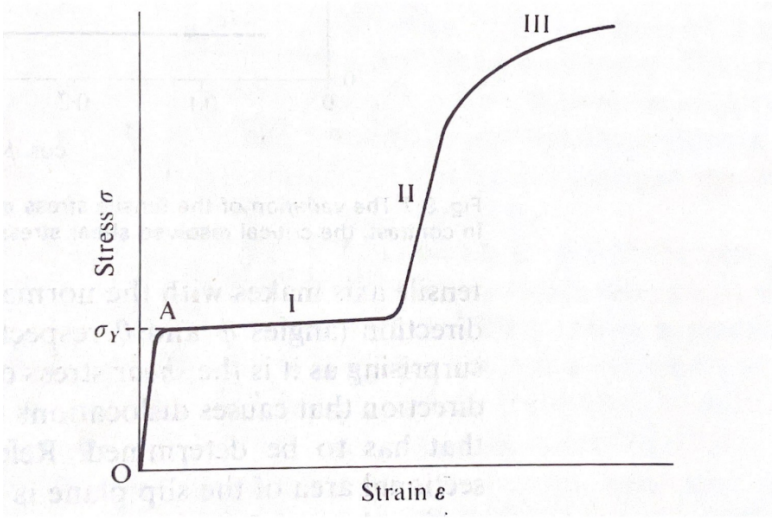

Figure 2.17: A generalised stress-strain curve for a single crystal. Different stages of plasticity are labelled. Reproduced from Anderson et al. [1] with permission from Springer Nature.

2.1.14 Plastic deformation

As discussed earlier, materials go plastic in response to shear stress over a certain threshold. The primary mechanism by which fcc materials go plastic is the growth and movement of dislocations, resulting in slip. Plastic deformation is an irreversible process, resulting in a new material microstructure. In an adiabatic system, this results in plastic heating, in addition to (and often greatly surpassing) any temperature rise due to work done in elastic compression.

There are several stages to plastic deformation: figure 2.17 shows a typical stress-strain curve for an fcc single crystal. At very low strains up to σ_Y, the stress increases linearly. After this point, the material deforms plastically. The first stage I is known as the easy glide region, which operates on the primary slip system as described in section 2.1.12. Since the dislocations are all moving on a parallel set of planes, they do not interact and move unhindered. For this reason, the material can continue to be compressed without requiring substantially more stress. In stage II, more

than one slip system begins operating. At this point, dislocations are generated and moving on intersecting slip planes, and can interfere with each other's path. Since this impedes dislocation velocity, the material experiences less plastic strain. Therefore more stress is required for further compression. This is known as work or strain-hardening. In stage III, the rate of this hardening reduces as dislocations are annihilated.

2.2 Continuum description of ramp and shock waves

When we model a system as a continuum, we assume that the pressure, temperature, and density vary continuously from point to point. The system must be divided into elements of sufficiently large volume such that they can be assigned an average quantity, but small enough that there is not significant variation within the volume. To model such a system, we can assume it is a fluid and is therefore governed by fluid mechanics. For that reason this is referred to as a hydrodynamic approach and the numerical codes used are often referred to as hydrocodes. The major texts of reference for this section are by Wilkins [66] and Meyers [75].

2.2.1 The Lagrangian frame

In continuum modelling there are two major reference frames used: Lagrangian and Eulerian. In a Lagrangian frame, we follow a particular particle or element of material as it flows. Its mass stays constant while its size and position may change. In an Eulerian frame, the cell boundaries are fixed in space and mass can flow in and out of the cell.

For an arbitrary property f, the relationship between the Lagrangian and Eulerian rates of change are

$$\frac{d}{dt}f = \frac{\partial}{\partial t}f + \boldsymbol{U}\cdot\boldsymbol{\nabla}f \tag{2.49}$$

where $\frac{d}{dt}f$ is the 'total time derivative' and is used in the Lagrangian referential, $\frac{\partial}{\partial t}f$ is the partial time derivative which is used in the Eulerian referential, and $\boldsymbol{U}\cdot\boldsymbol{\nabla}f$

is the convective rate of change. [75] The hydrocode work presented in this thesis uses the Lagrangian referential.

2.2.2 Fundamental equations of fluid mechanics

There are four fundamental equations of fluid mechanics. The first three express the conservation of momentum, mass, and energy. The fourth links these three together and is referred to as the equation of state.

The conservation of momentum is also referred to as the equation of motion, as it is simply Newton's second law applied to a Lagrangian fluid element:

$$\rho \frac{\mathrm{d}\boldsymbol{U}}{\mathrm{d}t} = -\nabla P \tag{2.50}$$

where ρ is density, t is time, $\boldsymbol{U}$ is the velocity vector, and P is the pressure.

The equation for conservation of mass is also referred to as the continuity equation. It states that in a continuous system, the rate of change of density is equal to the rate at which mass flows in to or out of that system:

$$\frac{1}{\rho}\frac{\mathrm{d}\rho}{\mathrm{d}t} + \nabla \cdot \boldsymbol{U} = 0 \tag{2.51}$$

For simplicity, we assume that processes are adiabatic and reversible. To express conservation of energy we therefore state that the rate of change of internal energy for a fluid element, if neglecting heat conduction and viscosity, is equal to the work done on that element by the pressure of its neighbouring elements:

$$\frac{\mathrm{d}E}{\mathrm{d}t} + P\frac{\mathrm{d}V}{\mathrm{d}t} = 0 \tag{2.52}$$

where E is the internal energy per unit mass of a fluid element, and V is its specific volume (equal to $1/\rho$.)

At this point, there are more unknowns than equations. Therefore to close the system, we must introduce the equation of state. This takes the general form

$$P = P(\rho, E) \tag{2.53}$$

The equation of state can take a multitude of different forms; it can be purely analytical, or may be purely interpolated from experimental data. Equations of state will be further discussed in section 2.2.5.

2.2.3 Sound (and ramp) waves

We will now consider what happens when a small perturbation is propagated through a fluid, for example by a piston. This will be an adiabatic, isentropic process. We will follow Wilkins' treatment. [66]

In one dimension, the equation of motion 2.50 becomes

$$\frac{\partial U}{\partial t} + U\frac{\partial U}{\partial X} + \frac{1}{\rho}\frac{\partial P}{\partial X} = 0 \tag{2.54}$$

and the conservation of mass equation becomes

$$\frac{\partial \rho}{\partial t} + U\frac{\partial \rho}{\partial X} = 0 \tag{2.55}$$

Since we are assuming the perturbation is isentropic, we can write the equation of state as

$$P = P(\rho) \tag{2.56}$$

We can consider the wave front as dividing the region into two states. On one side, we describe the density, pressure, and velocity as ρ_0, P_0, and U_0; on the other, we can write them as ρ_1, P_1 and U_1. Furthermore, we will define difference parameters L, M and N such that:

$$L = U_1 - U_0$$

$$M = \rho_1 - \rho_0$$

$$N = P_1 - P0$$

We denote U_S as the speed of the wave. The condition that a difference parameter Z is zero on the wave front is described by:

$$\frac{\partial Z}{\partial t} + U_S\frac{\partial Z}{\partial X} = 0 \tag{2.57}$$

By writing the earlier one-dimensional conservation equations for both regions and subtracting them, we arrive at:

$$\frac{\partial L}{\partial t} + U\frac{\partial L}{\partial X} + \frac{1}{\rho}\frac{\partial N}{\partial X} = 0 \tag{2.58}$$

$$\frac{\partial M}{\partial t} + U\frac{\partial M}{\partial X} + \rho\frac{\partial L}{\partial X} = 0 \tag{2.59}$$

We can find an equivalent expression for the equation of state at the wave front:

$$\frac{\partial N}{\partial t} + U\frac{\partial N}{\partial X} = \frac{\partial P}{\partial \rho}\left(\frac{\partial M}{\partial t} + U\frac{\partial M}{\partial X}\right) \tag{2.60}$$

By applying equation 2.57 to the above three expressions, we arrive at this system of equations:

$$\frac{1}{\rho}\frac{\partial N}{\partial X} + (U - U_S)\frac{\partial L}{\partial X} = 0 \tag{2.61}$$

$$\rho\frac{\partial L}{\partial X} + (U - U_S)\frac{\partial M}{\partial X} = 0 \tag{2.62}$$

$$(U - U_S)\frac{\partial N}{\partial X} - (U - U_S)\frac{\partial M}{\partial X}\frac{dP}{d\rho} = 0 \tag{2.63}$$

For a travelling perturbation, this system is only satisfied if

$$U_S = U \pm \sqrt{\frac{dP}{d\rho}} \tag{2.64}$$

where the term $\sqrt{\frac{dP}{d\rho}}$ is the speed of sound. If a piston continuously accelerates, this produces a 'ramp' wave, which, in the ideal scenario with no friction or heat flow, isentropically compresses the fluid. We can note that

$$\frac{dP}{d\rho} = \frac{K}{\rho_0} \tag{2.65}$$

where K is the bulk modulus, a measure of a substance's resistance to compression, and ρ_0 is the initial density. Therefore we can write the sound speed as

$$C = \sqrt{\frac{K}{\rho_0}} \tag{2.66}$$

Since materials become less compressible as they are compressed, the fate of a ramp wave can be inferred from equation 2.66: the sound speed is greater in more-

compressed regions, therefore the high amplitude section of the wave will travel faster than the low amplitude section. This means that the ramp wave will progressively steepen until it forms a discontinuity in P, U, ρ, and T. This discontinuity is called a shock wave, which will be discussed next.

2.2.4 Shock waves

A shock wave is a sharp discontinuity in pressure, density, temperature and velocity travelling through a medium. In our previously defined fluid, a shock could be formed from a sudden increase in piston velocity, or from a steepened ramp wave.

This discontinuity means that a new set of analogous fluid conservation equations, which connect the shocked and unshocked regions, are needed. These are referred to as the Rankine-Hugoniot relations.

They can be derived from conservation equations when considering a simple system of a piston being pushed at velocity U_p into a volume of stationary compressible material with an uncompressed density of ρ_0, pressure P_0 and internal energy E_0. [75] The compressed region of material just ahead of the piston, at pressure P and density ρ will get larger over time, meaning that the disturbance - the shock - will move faster than the piston and the compressed region. The plane separating the moving compressed material from the stationary uncompressed material is the shock front.

The Rankine-Hugoniot relationships, which describe the propagation of a pressure discontinuity through a material, can be derived from conservation of momentum, mass and internal energy in the compressed region. [75] From mass conservation,

$$U_s \rho_0 = (U_s - U_p)\rho \tag{2.67}$$

where U_s is the velocity of the shock front, ρ_0 is the density of the initial uncompressed material, ρ is the density of the compressed material between the shock front and the piston, and U_p is the particle velocity. From conservation of

momentum,

$$\rho U_s U_p = P - P_0 \tag{2.68}$$

where P_0 is the pressure in the initial uncompressed region and P is the pressure in the compressed region. From conservation of energy,

$$E_1 - E_0 = \frac{1}{2}(P + P_0)(V_0 - V) \tag{2.69}$$

Similarly to the original conservation equations, these must be closed by an equation of state $P(\rho, E)$.

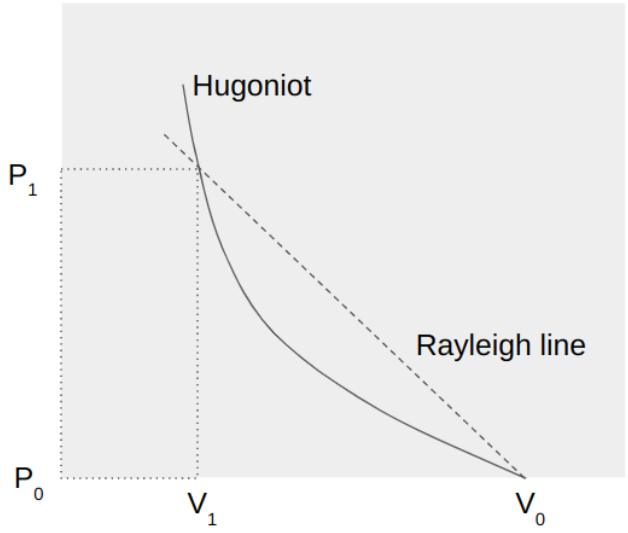

Figure 2.18: The Hugoniot curve and the Rayleigh line for a given equation of state.

Equation 2.69 establishes the relationship between pressure P and volume V. This is known as the Rankine-Hugoniot relation and is often simply referred to as the 'Hugoniot'. [75] For a given equation of state $P(\rho, E)$, the 'Hugoniot curve' can be seen in figure 2.18 and represents the locus of possible shocked states in a material. The straight line connecting (P_0, V_0) and (P_1, P_2) is known as the Rayleigh line. Using the relationships above, its slope can be found to be

$$\frac{P_1 - P_0}{V_1 - V_0} = -(\rho_0 U_S)^2 \tag{2.70}$$

which relates the initial and final states to the shock speed.

2.2.5 Equations of state

To quote Wilkins: "The greatest uncertainty in applying hydrodynamic theory to physical situations lies in the equation of state of the materials." [66] As discussed in earlier sections, these equations of state are necessary to understand a material's response to compression, which in turn informs the nature of further compression.

The most well-known equation of state is the ideal gas law:

$$PV = NkT \tag{2.71}$$

where N is the number of molecules, k is the Boltzmann constant, and T is the temperature. This closes the fundamental fluid equations because N/V is directly proportional to ρ and, since there are no interatomic forces and thus no potential energy, T is directly proportional to internal energy.

The Mie-Gruneisen equation of state is commonly used for describing metals at high pressures:

$$P = -\frac{dE_L}{dV} + \frac{\gamma}{V}(E - E_L) \tag{2.72}$$

where the first term is the pressure due to lattice potential energy E_L at absolute zero, and the second term is the thermal pressure due to the lattice vibrational energy $E - E_L$. γ is known as the Gruneisen parameter which has several definitions, but in this context the following definition is useful:

$$\gamma = -\left(\frac{\partial ln(T)}{\partial ln(V)}\right)_S \tag{2.73}$$

which describes how the temperature T evolves in response to isentropic changes in volume V. γ is often determined experimentally or via *ab initio* calculations.

It is important to note that there are a great range of different techniques for determining the EOS of a material. They are often a combination of analytical, experimental and computational methods. The SESAME look-up tables, for example, employ a great number of different models to represent different regions of phase space, and which are supported by a substantial experimental database. [76]

2.3 Numerical techniques in continuum modelling

We will now discuss the computational aspect of the continuum modelling of materials subject to dynamic compression. While this is a broad field, we will focus on the background of a particular scheme developed by Wilkins [66] for simulating the shock compression of an elastic-plastic material due to its role in the work of this thesis.

2.3.1 Finite difference schemes

The goal of finite difference schemes in the continuum modelling of materials is to use finite difference approximations to numerically evaluate the fundamental equations of fluid mechanics at discrete points of space and time. A finite difference approximation is a method to numerically solve differential equations. There are three major approximations that can be made: forward difference, backward difference, and central difference.

All of these are based on Taylor series expansions of a function $f(x)$ about x:

$$f(x + \Delta x) = f(x) + \Delta x f'(x) + \frac{1}{2}\Delta x^2 f''(x) + ... \tag{2.74}$$

which can be rearranged for its derivative:

$$f'(x) = \frac{f(x + \Delta x) - f(x)}{\Delta x} - \frac{1}{2}\Delta x f''(x) + ... \tag{2.75}$$

The forward difference approximation is then simply the first term on the right hand side. The later terms are known as the truncation error. Since the leading order term of the truncation error grows linearly with Δx, it is known as a first-order approximation. The backward difference approximation is simply the mirror image of the forward difference approximation, and can be found by writing the Taylor expansion of $f(x - \Delta x)$:

$$f(x - \Delta x) = f(x) - \Delta x f'(x) + \frac{\Delta x^2}{2} f''(x) + ... \tag{2.76}$$

and again rearranging for $f'(x)$

$$f'(x) = \frac{f(x) - f(x - \Delta x)}{\Delta x} + \frac{1}{2}\Delta x f''(x) + ... \tag{2.77}$$

where the backward difference approximation is again the first term on the right hand side and the later terms sum to the truncation error. Both the forward and backward differences are first-order approximations. We can obtain a more accurate approximation by subtracting 2.76 from 2.74, and again rearranging for $f'(x)$:

$$f'(x) = \frac{f(z + \Delta x/2) - f(x - \Delta/2)}{\Delta x} - \frac{1}{24}\Delta x^2 f'''(x) + ... \tag{2.78}$$

where the first term on the right hand side is the central difference approximation. Here, the greatest term of the truncation error is an order higher in Δx, making it a second order approximation.

2.3.2 Time integration

We mentioned previously that to numerically approximate time-dependent differential equations, both time and space must be discretised. There are two general approaches to time integration, known as explicit and implicit methods.

In explicit methods, also known as 'time-marching' methods, the state of the system at the later time step is computed from known quantities at the current time:

$$y_{n+1} = y_n + \Delta t f(y_n) \tag{2.79}$$

In implicit methods, an equation relating the current state of the system to the later state is solved:

$$y_{n+1} = y_n + \Delta t f(y_{n+1}) \tag{2.80}$$

A numerical solution must therefore be found for each time step. For this reason implicit methods can be computationally demanding. As well as being computationally cheaper, explicit methods are widely considered to be easier to implement. The continuum modelling in this thesis uses explicit time integration. However, explicit time integration is less numerically stable than implicit methods,

and must obey the Courant condition.

2.3.3 Courant condition

During explicit time integration of differential equations, the Courant condition requires that the time step Δt is sufficiently small that a perturbation travelling at velocity a cannot complete a distance larger than the grid spacing. In one space dimension x it can be expressed by:

$$\frac{a\Delta t}{\Delta x} < C \tag{2.81}$$

C is known as the "Courant number" and, in keeping with the stated requirement, must not exceed 1. This means that information can only propagate to immediate neighbouring cells in a single time step - it cannot skip over a cell. To do so would result in an unstable solution.

2.3.4 Staggered grids

When using central differences to propagate a wave in space and time, it is common to use staggered grids. This allows higher accuracy in computing difference equations without increasing the number of grid points. To demonstrate this, we can consider the following one-dimensional equation of motion:

$$\frac{\partial u}{\partial t} = \frac{1}{\rho}\frac{\partial \sigma}{\partial x} \tag{2.82}$$

Firstly, let us consider discretising this equation using a regularly spaced grid where velocity u and stress σ are defined at the same points in space and time, shown in figure 2.19a. To define this relation at position j at time step n, equation 2.82 would become

$$\frac{u_j^{n+1} - u_j^{n-1}}{2\Delta t} = \frac{1}{\rho}\frac{\sigma_{j+1}^n - \sigma_{j-1}^n}{2\Delta x} \tag{2.83}$$

where Δt is the time step size and Δx is the spatial mesh size. However, we can also use a staggered grid, shown in figure 2.19b, which results in the following discretised

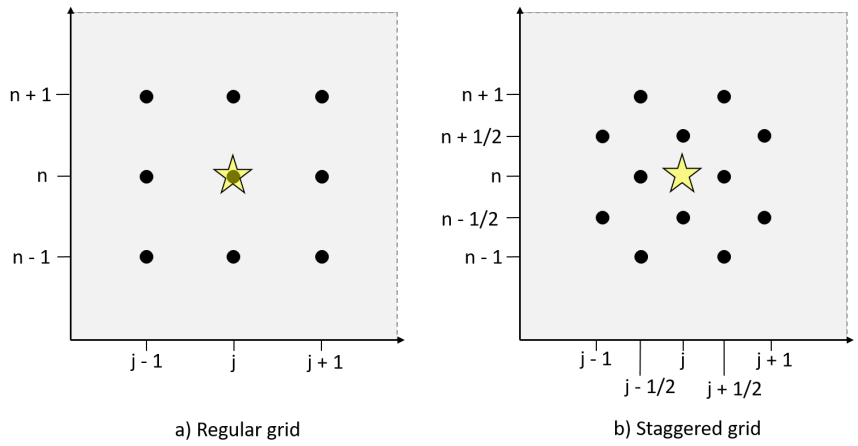

Figure 2.19: a) A representation of a regular grid in a finite difference scheme. To sample at the position j and at time n, the values at the points $(j-1, n)$, $(j+1, n)$, $(j, n-1)$, and $(j, n+1)$ must be used. b) A representation of a staggered grid. To sample at (j, n), the values at $(j-1/2, n)$, $(j+1/2, n)$, $(j, n-1/2)$ and $(j, n+1/2)$ can be used.

equation:

$$\frac{u_j^{n+\frac{1}{2}} - u_j^{n-\frac{1}{2}}}{\Delta t} = \frac{1}{\rho}\frac{\sigma^n_{j+\frac{1}{2}} - \sigma^n_{j-\frac{1}{2}}}{\Delta x} \tag{2.84}$$

Note that when using a regular grid, we are making a linear approximation over double the time step size and the spatial mesh size. By using a staggered mesh, we approximate over half those values. Therefore the staggered grid achieves double the accuracy without unnecessarily doubling the grid points in each dimension.

Consequently, when discussing spatial points, we use the terms nodes and zones. Zones refer to the space in between consecutive grid lines and nodes refers to the points at which they intersect (or their 'faces'.) If the points j, $j+1$ and $j-1$ are nodes, then $j+\frac{1}{2}$ and $j-\frac{1}{2}$ are zones. We will use this terminology later.

2.3.5 Stability analysis of finite difference equations

The Courant condition is necessary but not sufficient to ensure stability if we are using finite difference equations. All numerical approximations performed by a computer will have some degree of 'round-off' error ϵ because they have finite precision:

$$\epsilon = N - D \tag{2.85}$$

where N is the finite-precision numerical solution from a real computer and D is the exact solution of the finite difference equation. The question of whether this error grows over time determines the stability of the approximation. For a solution to be stable, it must not increase between time t_n and $t_n - \Delta t$:

$$|\epsilon^{t_n}/\epsilon^{t_n-\Delta t}| \leq 1 \tag{2.86}$$

To determine under what conditions a finite difference equation is stable we can use von Neumann stability analysis. We will follow the outline of this analysis given by Cassel [77]. Von Neumann analysis rests on the representation of the error as a Fourier series in x:

$$\epsilon(x,t) = \sum_m a_m(t) e^{ik_m x} \tag{2.87}$$

where each mode m has a particular amplitude $a(t)$, which is a function of time. Since we have written the error as a linear combination of modes, we need only examine the stability of one mode to determine the stability of the summation. To do this, we introduce the amplification factor G:

$$G_m(x,n) = \frac{a_m(t_n)}{a_m(t_n - \Delta t)} \tag{2.88}$$

which simply indicates the degree to which a particular mode has grown or shrunk over one time step. If $|G| > 1$, the amplitude of a mode is growing. If $|G| \leq 1$, the amplitude has either stayed the same or is decreasing. The amplification factor after n time steps is simply the n^{th} power of G_m:

$$(G_m)^n = \frac{a_m(t_n)}{a_m(t_n - \Delta t)} \cdot \frac{a_m(t_n - \Delta t)}{a_m(t_n - 2\Delta t} \cdot \ldots \cdot \frac{a_m(\Delta t)}{a_m(0)} \tag{2.89}$$

which simplifies to

$$(G_m)^n = \frac{a_m(t_n)}{a_m(0)} \tag{2.90}$$

The amplitude of the m^{th} mode at time t_n is therefore simply $a_m(t_n) = (G_m)^n a_m(0)$. Therefore the error at time t_n can be written as

$$\epsilon(x,t_n) = \sum_m (G_m)^n a_m(0) e^{ik_m x} \tag{2.91}$$

We can see from equation 2.85 that the numerical solution N is a linear summation of ϵ and D. Since N and D by definition must satisfy a particular finite difference equation, ϵ must equally satisfy that equation. By substituting the error of one mode $(G_m)^n a_m(0)e^{ik_m x}$ into our desired finite difference equation we can solve for the amplification factor G_m. By recalling that $|G_m| \leq 1$ for stability, we can determine the stability criteria for a particular finite difference equation.

2.3.6 Artificial viscosity

Artificial viscosity was first introduced as a tool in 1950 by Von Neumann and Richtmyer [78]. It emerged from the difficulty in simulating shocks using finite difference programs. This is because shocks are presented as discontinuities i.e. not differentiable. At its core, artificial viscosity is a mathematical technique to artificially smooth out these discontinuities, giving them a finite derivative and preventing the growth of a pathological instability in a hydrocode. This works by adding a term q to the pressure. Von Neumann and Richtmyer emphasise that q may take any convenient form, provided that (1) it serves the function of removing discontinuities from solutions of the fundamental fluid equations, (2) it results in a shock thickness on the order of the mesh size, (3) its effect is negligible outside the shock and (4) it does not affect the Hugoniot relations when the shock thickness is of negligible size compared to other aspects of the flow. Wilkins uses this following form for q in one dimension:

$$q = C_0^2 \rho (\Delta x)^2 \left(\frac{\partial \dot{x}}{\partial x}\right)^2 + C_L \rho \Delta x \sqrt{\frac{P}{\rho}} \left|\frac{\partial \dot{x}}{\partial x}\right| \tag{2.92}$$

where C_0 and C_L are dimensionless viscosity coefficients that can be adjusted by the user, though they are typically near unity. It can be observed from equation 2.92 that a greater spatial derivative of velocity results in a larger value of q. Correspondingly, a constant velocity results in a q of zero.

2.3.7 Continuum modelling of materials: a one-dimensional finite difference program for elastic-plastic flow

To simulate the compression of solid materials using continuum modelling, analytic elastic-plastic models can be used in tandem with the fluid equations. An example of this is the following one-dimensional finite difference program by Wilkins which simulates elastic-plastic flow using the Von Mises yield criterion, discussed in section 2.1.11. This scheme uses a staggered grid, as discussed in section 2.3.4. Spatial points are either 'zones', located between the grid points at $j + \frac{1}{2}$ and $j - \frac{1}{2}$, or 'nodes', located at the faces of the grid points at j, $j + 1$ and $j - 1$. We begin by initialising the mass in each zone:

$$m_{j+\frac{1}{2}} = \frac{\rho_0}{V_0}\left[r^0_{j+1} - r^0_j\right] \tag{2.93}$$

where ρ_0 is the initial density (simply a line density for one dimension) and V_0 is the initial relative volume (which is dimensionless.) Since it is a Lagrangian grid, the mass of each zone stays constant throughout the simulation. At this point the main loop begins. We first calculate the velocities of the nodes using the equation of motion:

$$U^{n+\frac{1}{2}}_j = U^{n-\frac{1}{2}}_j + \frac{\Delta t^n}{\phi^n_j}\left[\Sigma^n_{j+\frac{1}{2}} - \Sigma^n_{j-\frac{1}{2}}\right] \tag{2.94}$$

where $\Sigma^n_{j+\frac{1}{2}}$ is the total stress at the zones, computed from the pressure P, deviatoric stress s and artificial viscosity q:

$$\Sigma^n_{j+\frac{1}{2}} = \left[-(P^n + q^{n-\frac{1}{2}}) + s^n_1\right]_{j+\frac{1}{2}} \tag{2.95}$$

and ϕ^n_j is the nodal mass, which can be thought of as the areal density of a point (it is therefore distinct from the zonal mass, which is the total mass of a zone):

$$\phi^n_j = \frac{1}{2}\left[\rho_{j+\frac{1}{2}}\left(\frac{r^n_{j+1} - r^n_j}{V^n_{j+\frac{1}{2}}}\right) + \rho_{0j+\frac{1}{2}}\left(\frac{r^n_j - r^n_{j-1}}{V^n_{j-\frac{1}{2}}}\right)\right] \tag{2.96}$$

We then use the conservation of mass equation to calculate the new relative volumes V of the zones using the new positions r of the nodes:

$$V_{j+\frac{1}{2}}^{n+1} = \frac{\rho_0}{m_{j+\frac{1}{2}}} \left[r_{j+1} - r_j\right]^{n+1} \tag{2.97}$$

where

$$r_j^{n+1} = r_j^n + U_j^{n+\frac{1}{2}} \Delta t^{n+\frac{1}{2}} \tag{2.98}$$

The true strain rate for each zone is then computed:

$$\dot{\epsilon}_{j+\frac{1}{2}}^{n+\frac{1}{2}} = \frac{U_{j+1}^{n+\frac{1}{2}} - U_j^{n+\frac{1}{2}}}{r_{j+1}^{n+\frac{1}{2}} - r_j^{n+\frac{1}{2}}} \tag{2.99}$$

At this point the stress deviators of the zones are computed, whose new values depends on a given shear modulus:

$$(s_1)_{j+\frac{1}{2}}^{n+1} = (s_1)_{j+\frac{1}{2}}^{n} + 2\mu \left[\dot{\epsilon}_{j+\frac{1}{2}}^{n+\frac{1}{2}} \Delta t^{n+\frac{1}{2}} - \frac{1}{3}\left(\frac{V^{n+1} - V^n}{V^{n+\frac{1}{2}}}\right)_{j+\frac{1}{2}}\right] \tag{2.100}$$

$$(s_2)_{j+\frac{1}{2}}^{n+1} = (s_2)_{j+\frac{1}{2}}^{n} + 2\mu \left[-\frac{1}{3}\left(\frac{V^{n+1} - V^n}{V^{n+\frac{1}{2}}}\right)_{j+\frac{1}{2}}\right] \tag{2.101}$$

$$(s_3)_{j+\frac{1}{2}}^{n+1} = -\left[(s_1)_{j+\frac{1}{2}}^{n+1} + (s_2)_{j+\frac{1}{2}}^{n+1}\right] \tag{2.102}$$

The above computation for the stress deviators is all based on Hooke's law, which of course applies only to elasticity, not plasticity. To incorporate plasticity, we employ the Von Mises yield condition, which was discussed back in section 2.1.11:

$$(s_1^2 + s_2^2 + s_3^3)^{n+1} - 2/3(Y^0)^2 = K^{n+1} \tag{2.103}$$

where Y^0 is the user-specified yield strength. The value of K^{n+1} is checked to determine if the material is within the elastic limit; if it isn't, the stress deviators are modified to obey the criterion. Therefore if $K^{n+1} \leq 0$ then no changes are made. If $K^{n+1} > 0$ then the stress deviators must be multiplied by $\sqrt{2/3}Y^0/\sqrt{(s_1^2 + s_2^2 + s_3^2)}$.

If $U_{j+1}^{n+\frac{1}{2}} < U_j^{n+\frac{1}{2}}$ and $(V_{j+\frac{1}{2}}^{n+1} - V_{j+\frac{1}{2}}^{n}) < 0$ then the q value for artificial viscosity is computed:

$$q_{j+\frac{1}{2}}^{n+\frac{1}{2}} = C_0^2 \rho_{j+\frac{1}{2}}^{n+\frac{1}{2}} (U_{j+1}^{n+\frac{1}{2}} - U_j^{n+\frac{1}{2}})^2 + C_L \sqrt{P/\rho}_{j+\frac{1}{2}}^{n+\frac{1}{2}} |U_{j+1}^{n+\frac{1}{2}} - U_j^{n+\frac{1}{2}}| \tag{2.104}$$

where P is the local pressure, and C_0 and C_L are dimensionless viscosity coefficients. Wilkins sets these parameters as $C_0 = 2$ and $C_L = 1$.

The change in internal energy is then computed:

$$\Delta E = -(P + q)\Delta V + \Delta Z \tag{2.105}$$

where ΔZ is the change in distortion energy:

$$\Delta Z_{j+\frac{1}{2}}^{n+\frac{1}{2}} = V_{j+\frac{1}{2}}^{n+\frac{1}{2}} [s_1 \dot{\epsilon}_1]_{j+\frac{1}{2}}^{n+\frac{1}{2}} \Delta t^{n+\frac{1}{2}} \tag{2.106}$$

and where

$$s_1^{n+\frac{1}{2}} = \frac{1}{2}(s_1^{n+1} - s_1^n) \tag{2.107}$$

The total internal energy E is

$$E_{j+\frac{1}{2}}^{n+1} = \left(\frac{E^n - (\frac{1}{2}[A(\eta^{n+1}) + P^n] + \bar{q})[V^{n+1} - V^n] + \Delta Z}{1 + \frac{1}{2}[B(\eta^{n+1})][V^{n+1} - V^n]} \right)_{j+\frac{1}{2}} \tag{2.108}$$

where $\eta = \frac{1}{V} = \frac{\rho}{\rho_0}$ and $\bar{q} = \frac{1}{2}(q^{n+\frac{1}{2}} + q^{n-\frac{1}{2}})$. This is then used to calculate the pressure using an equation of state:

$$P_{j+\frac{1}{2}}^{n+1} = A(\eta_{j+\frac{1}{2}}^{n+1}) + B(\eta_{j+\frac{1}{2}}^{n+1}) E_{j+\frac{1}{2}}^{n+1} \tag{2.109}$$

In Wilkins' formulation, the time step size is variable. Its new value is computed in the final step of the main loop. It is computed according to the Courant condition and from further stability conditions derived from the application of von Neumann analysis, which have been combined by Wilkins.

$$\Delta t^{n+\frac{3}{2}} = \frac{2}{3} \left. \frac{\Delta r^{n+1}}{\sqrt{\sqrt{\frac{P}{\rho}}^2 + b^2}} \right|_{\text{minimum over j}} \tag{2.110}$$

where

$$\Delta r^{n+1} = r_{j+1}^{n+1} - r_j^{n+1} \tag{2.111}$$

and where, if $\dot{V}/V < 0$,

$$b = 8(C_0^2 + C_L)\Delta r^{n+1} \left(\frac{\dot{V}}{V} \right)^{n+\frac{1}{2}} \tag{2.112}$$

and otherwise:

$$b = 0 \tag{2.113}$$

And if $\Delta t^{n+\frac{3}{2}} > (1.1)\Delta t^{n+\frac{1}{2}}$, then use

$$\Delta t^{n+\frac{3}{2}} = (1.1)\Delta t^{n+\frac{1}{2}} \tag{2.114}$$

Finally, the new time step is computed:

$$\Delta t^{n+1} = \frac{1}{2}(\Delta t^{n+\frac{3}{2}} + \Delta t^{n+\frac{1}{2}}) \tag{2.115}$$

2.3.8 Boundary conditions

Since computing the derivative requires points on either side of j, the conditions on the regional boundaries must be defined separately. In Wilkins' formulation it is accomplished in the following way. 'Inner' refers to low j values, while 'outside' refers to the maximum j values. These can also be thought of as the left hand and right hand boundaries. On an outside regional boundary J

$$\phi_J^n = \frac{1}{2}\rho_{0J-\frac{1}{2}} \left(\frac{r_J^n - r_{J-1}^n}{V_{J-\frac{1}{2}}^n} \right) \tag{2.116}$$

and on an inside regional boundary J

$$\phi_J^n = \frac{1}{2}\rho_{0J+\frac{1}{2}} \left(\frac{r_{J+1}^n - r_J^n}{V_{J+\frac{1}{2}}^n} \right) \tag{2.117}$$

2.4 Molecular dynamics

2.4.1 General description

Molecular dynamics is an atomistic method of materials modelling. It is a deterministic technique that numerically integrates classical equations of motion for a group of atoms or molecules to calculate their trajectory in time. Beginning

with a group of atoms at time $t = 0$ whose positions and velocities are specified, the forces on each atom are computed according to a list of interatomic potentials. By choosing a small time step Δt and integrating forward in time, the positions and velocities of the atoms are updated. This is repeated for a specified number of time steps. The time step Δt must be short enough to capture the important dynamics of the system, but not so short as to make the simulation long and inefficient.

Numerical integration can be done with various finite difference methods for ordinary differential equations. The MD code used in this thesis, LAMMPS (which stands for Large Scale Atomic/Molecular Massively Parallel Simulator, distributed by Sandia National Laboratories [63]) uses the velocity Verlet integrator [79] where the position $\mathbf{r}$ and the velocity $\mathbf{v}$ of each atom are updated like so:

$$\mathbf{r}(t + \Delta t) = \mathbf{r}(t) + \Delta t \mathbf{v}(t) + \frac{1}{2}\Delta t^2 \mathbf{a}(t) \tag{2.118}$$

$$\mathbf{v}(t + \Delta t) = \mathbf{v}(t) + \frac{1}{2}\Delta t[\mathbf{a}(t) + \mathbf{a}(t + \Delta t)] \tag{2.119}$$

where the acceleration $\mathbf{a}$ is derived from an interatomic potential function of $\mathbf{r}$:

$$\mathbf{a} = \frac{\mathbf{F}}{m} = -\frac{1}{m}\nabla E(\mathbf{r}) \tag{2.120}$$

Different forms of the potential function $E(\mathbf{r})$ will be described in section 2.4.2.

2.4.2 Interatomic potentials

Interatomic potentials describe the forces between particles. They can range from simple pair potentials to more sophisticated many-body potentials. The Lennard-Jones potential, shown in figure 2.20, is an example of a simple pair potential:

$$E(r) = 4\epsilon \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6} \right] \tag{2.121}$$

where r is the distance, ϵ is the potential well depth and σ is the distance at which the potential between two interacting particles is zero.

Pair-potentials make the assumption that the bonds between atoms can be treated independently. However, this is not generally true for metals. [80] To account

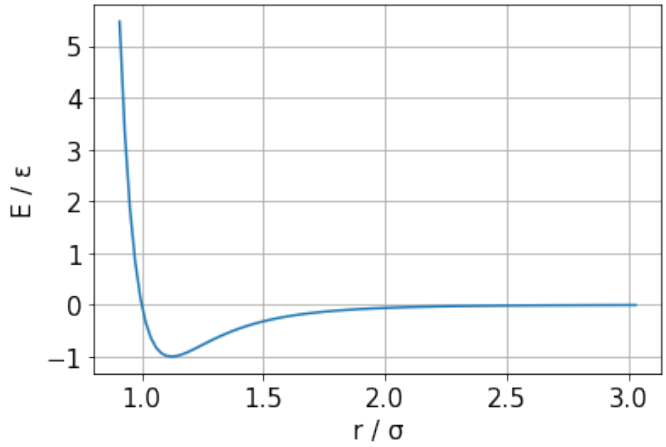

Figure 2.20: The Lennard-Jones pair potential. The potential is a function of the distance r between two particles.

for the many-atom interactions that are present in metals, it is more appropriate to use a many-body potential such as the Embedded Atom Model (EAM). [81] In this model, the total energy E_i of an atom i is given by

$$E_i = F_\alpha \left(\sum_{j \neq i} \rho_\beta(r_{ij}) \right) + \frac{1}{2} \sum_{j \neq i} \phi_{\alpha\beta}(r_{ij}) \tag{2.122}$$

where F is a function of the electron density ρ, ϕ is a short-range pair potential, and α and β refer to the element types of atoms i and j (in the case of a metal alloy.) [82] EAM is referred to as a many-body potential, because it sums over all neighbours j of atom i within a particular cut-off distance. When used in MD, EAM potentials are typically referenced from a tabular format which contain F, ρ, ϕ, and the cut-off distance. They can be developed from theory (predominantly density functional theory or the tight binding model) or fitted to experimental data, or a combination. The accuracy of an MD simulation largely depends on the accuracy of this interatomic potential.

2.4.3 Boundary conditions

LAMMPS supports three main types of simulation boundaries: periodic, non-periodic and fixed, and non-periodic and 'shrink-wrapped'. A periodic boundary means that a particle can exit the boundary and re-enter at the opposite side. Particles will interact with particles 'across' the boundary if they lie within the

cut-off distance; these are referred to as 'ghost' atoms. The purpose of periodic boundary conditions is to approximate an infinite system on finite computational resources. They should only be used in dimensions where there are no directional processes, for example wave propagation.

In these other cases, boundaries can either be fixed in space, which means that atoms that cross the boundaries are simply removed from the simulation, or 'shrink-wrapped', where the boundary position is set by the farthest-flung atom to ensure all atoms are included in the simulation boundaries.

Different boundary conditions can be used in different dimensions. For example, if a wave is propagating in the $\hat{z}$ direction, periodic boundaries can still be used in directions $\hat{x}$ and $\hat{y}$ while a shrink-wrapped boundary may be used in $\hat{z}$.

2.4.4 System measurements

The total energy of an MD simulation is the sum of the potential and kinetic energies.

$$E_T = U + K \tag{2.123}$$

The potential energy is evaluated using a interatomic potential function E_P described in section 2.4.2. The total potential energy is simply the sum over all atoms:

$$U = \sum_{i<j} E_P(r_{ij}) \tag{2.124}$$

where i and j are different atoms and r_{ij} is the distance between them. The total kinetic energy is simply the sum of the kinetic energies of each atom, evaluated using their masses and velocities:

$$K = \frac{1}{2} \sum_{i}^{N} m_i v_i^2 \tag{2.125}$$

The temperature for a three-dimensional simulation is calculated using the kinetic energy:

$$T = \frac{2}{3} \frac{K}{k_B N} \tag{2.126}$$

where k_B is Boltzmann's constant and N is the number of atoms.

The pressure is computed using

$$P = \frac{Nk_BT}{V} + \frac{\sum_i^{N'} r_i \cdot f_i}{dV} \tag{2.127}$$

where the first term is the kinetic energy contribution and the second term is the virial contribution from interactions with other atoms. V is the volume, r_i and f_i are the position and force vectors of atom i. N' is the total number of atoms plus the 'ghost' atoms from using periodic boundary conditions. The stress tensor for the system is computed using

$$-S_{AB} = \frac{\sum_i^{N} m_i v_{iA} v_{iB}}{V} + \frac{\sum_i^{N'} r_{iA} f_{iB}}{V} \tag{2.128}$$

where A and B are directions x, y, z.

The volume of an atom is estimated using Voronoi tessellation. This defines a geometric cell around the atom whose faces are positioned at the midpoint between the atom's central position and the central position of a neighbouring atom.

2.4.5 Thermodynamic ensembles

In order to derive thermodynamic properties from a system, it is necessary to make use of statistical ensembles. In molecular dynamics, we chiefly make use of three ensembles: the microcanonical (NVE) ensemble, the isothermal-isobaric (NPT) ensemble, and the canonical (NVT) ensemble.

The work of this thesis predominantly uses the microcanonical ensemble. This applies unmodified integration of Newton's equations of motion. If the simulation is subject to no other manipulations or forces, there will be no change in number of particles, volume, or total energy, hence why it is also known as the "constant NVE" ensemble. The ensemble is still appropriate to use when the system is subject to a driving force or manipulation, which may change the volume or energy.

The isothermal-isobaric ensemble (NPT) may be used to set a system to a particular pressure and temperature, combining a barostat and a thermostat. In the work of this thesis, this ensemble is typically used for the purpose of equilibration at the beginning of a simulation. The canonical ensemble (NVT) only thermostats.

Both of these ensembles apply time integration on the Nose-Hoover style non-Hamiltonian equations of motion. [83] [84] The canonical ensemble is not used in the work of this thesis.

2.4.6 Existing methods of simulating shock and ramp compression

Shock waves travelling through a solid 3-D material have been modelled by non-equilibrium MD for 40 years. [85] However, it was not until the turn of the century, with the advent of massively parallel MD simulations and more realistic semi-empirical interatomic potentials, that its value in modelling the shock compression of condensed matter was fully recognised. [86] [87]

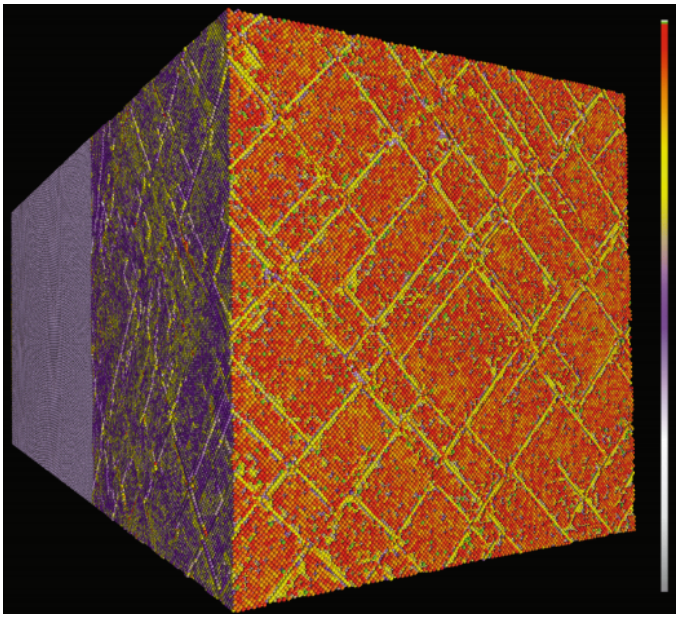

Figure 2.21: Multi-million atom NEMD simulation of shock compression induced by a momentum-mirror. Reproduced from Holian and Lomdahl [3] with permission from The American Association for the Advancement of Science.

Modelling a shock can be done mechanically, such as by using a frozen layer of atoms as a piston or by using a momentum mirror technique on a moving wall. This latter technique flips an atom's velocity relative to the moving wall when it comes in contact. This latter technique is shown in figure 2.21.) [3] These are non-equilibrium molecular dynamics (NEMD) techniques, which describe a system that is subject to a driving outside force. These NEMD techniques capture all

the dynamics that take place at the shock front as well as the evolution of the shocked material and so they are favoured for understanding how a system evolves under shock or ramp compression. However, they can be a more computationally expensive option because sufficient material must be simulated through which the shock can propagate. If only the shock front is of interest, the moving window technique may be used [88] in which a shock wave is propagated through the system (by using, for example, a piston) and then the material behind the shock front is deleted while material ahead of the shock front is created.

If only the final state of shock compression is of interest, a computationally cheaper equilibrium alternative that can be used on smaller system sizes is the Hugoniostat method developed by Ravelo *et al.* [89]. This reproduces the final state of shock-compressed material by uniaxially and homogenously compressing the system to conditions on the Hugoniot locus according to a target stress value. Similar equilibrium techniques include the constant-volume Hugoniostat method [90] and the multi-scale shock technique [91] which both restrain the systems to the Hugoniot and Rayleigh lines.

An alternative technique to simulate both the propagation of the shock front and the evolution of the shocked area is the absorbing boundary condition. [92] In this method, a piston is used to drive a shock into the material. Once the shock reaches the back of the sample, the back boundary starts to move with the same velocity as the first piston. This prevents the reflection of the wave and subsequent interference if the shock were to hit the back of the sample, while still allowing the material to evolve.

2.4.7 Structural analysis methods

This section will review two common methods for structural analysis which are relevant to this thesis. One of the most commonly used methods to determine local crystal structure in an atomistic simulation is common neighbour analysis (CNA) [93] which assigns a crystal structure depending on how well the topology of bonds matches a reference signature of a particular crystal structure. Two atoms

are classified as near-neighbours if they are within a specified cut-off distance. Then each near-neighbour bond of a central atom is assigned three characteristic numbers: the number of near-neighbours the atoms have in common; the total number of bonds between these common neighbours; and the number of bonds in the longest chain connecting these common neighbours. This set of numbers is then compared to a reference signature in order to categorise the central atom as belonging to a particular structure.

Adaptive CNA can be used for multi-phase systems, because rather than using a single-specified cut-off distance (determined by one particular reference structure) a cut-off radius is picked individually for each atom by comparing its generated reference signature with different candidate structures. [94]

The dislocation extraction algorithm (DXA) [64] is built upon CNA to identify dislocations and measure their count and length. It first uses CNA to distinguish crystalline atoms from disordered atoms. It then constructs a closed, two-dimensional interface mesh which separates the crystalline atoms from the disordered atoms. It then generates an initial Burgers circuit on this mesh, which is used to capture the shape of the dislocation, and produces a one-dimensional line to represent the dislocation.

Adaptive CNA and DXA are both implemented and accessible in OVITO, an open-source software for visualisation and analysis of atomistic simulations. [65]

2.5 Neural networks

Artificial neural networks are a machine learning technique - an algorithm to 'learn' the relationship between an input and an output. They can be used for regression problems whose objective is to produce a numerical quantity in response to an input value, or for classification problems whose objective is to categorise a particular input. In this thesis, the application is regression, and the type of neural network used is the most basic (though highly effective): a feedforward network. Other network types, such as convolutional neural networks, will not be discussed.

In the following sections we describe the basic elements of a neural network and the necessary considerations when designing them.

2.5.1 Architecture design

The basic architecture of a feed-forward neural network is composed of an input layer, an output layer, and a group of 'hidden' layers. We will describe them by following the notation of Bishop [95]. Layers themselves are composed of nodes, or 'neurons', which are not connected to one another but which are typically connected to all nodes in the preceding and following layers.

We will first consider how a simple linear function can be expressed as a single-layer network. We first consider the function

$$y(\bar{x}) = \bar{w}^T \bar{x} + \omega_0 \tag{2.129}$$

where $\bar{w}$ is a 'weight' vector, $\bar{x}$ is the input vector, and ω_0 is a 'bias' scalar. We can write this more simply by defining new vectors $\tilde{w} = (\omega_0, \bar{w})$ and $\tilde{x} = (1, \bar{x})$ so that equation 2.129 becomes

$$y(\bar{x}) = \tilde{w}^T \tilde{x} \tag{2.130}$$

This equation is represented in figure 2.22 as a single-layer network. We can represent a multi-variate nonlinear function by adding further layers, called 'hidden' layers, in the following way.

A network with a single hidden layer is shown in figure 2.23. In this network there are d input nodes, M is the number of nodes in the hidden layer, and c is the number of output nodes. Every connection, represented in the figure as straight lines, has an associated weight. The outputs of a layer are obtained via a nonlinear transformation of a weighted sum of the previous layer's output. For a given node j in the hidden layer, the weighted sum is first obtained:

$$a_j = \sum_{i=0}^{d} w_{ji}^{(1)} x_i \tag{2.131}$$

where w_{ij} is the weight associated with the connection between node i in the input

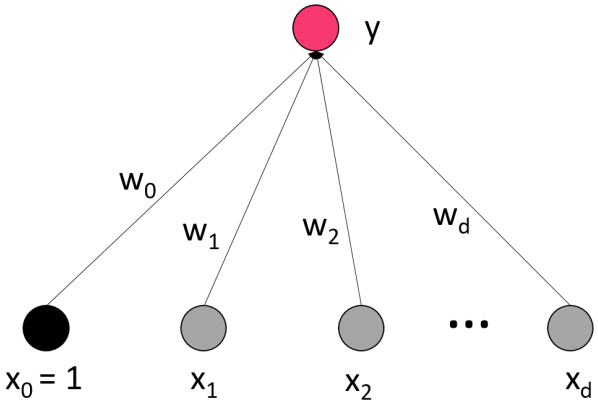

Figure 2.22: A single layer network with d inputs. x is an input vector, to which weights w are applied to result in the output y. x_0 is known as the 'bias' and is set to 1.

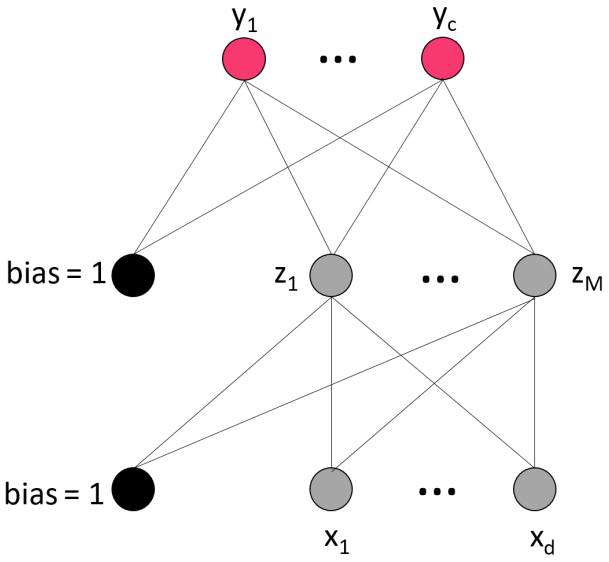

Figure 2.23: A feedforward network with a single hidden layer. x is the input vector, z is the vector calculated using an activation function $g(\cdot)$ and y is the output vector.

layer and node j in the hidden layer. We then transform this weighted sum a_j using a nonlinear 'activation' function $g(\cdot)$ to obtain the output z_j:

$$z_j = g(a_j) \tag{2.132}$$

In the final layer, the output nodes behave similarly. The only difference is that their input is the output of the hidden layer, and if desired they use a different activation function:

$$y_k = \tilde{g}(\sum_{j=0}^{M} w_{kj}^{(2)} z_j) \tag{2.133}$$

Common nonlinear activation functions - and the requirement that they be differentiable in the context of training - will be discussed later. In regression problems (as opposed to classification problems, which will not feature in this thesis) a linear activation function is typically used for the nodes of the output layer.

It is hopefully obvious how this can be extended to a multi-layer network with an arbitrary number of hidden layers. The output z_j of each node in a hidden layer can be written as

$$z_j = g(\sum_{i} w_{ij} z_i) \tag{2.134}$$

where g is the activation function of node j, z_i is the output of node i which sends a connection to node j, and w_{ij} is the weight associated with the connection.

The number of layers, and the number of nodes in each layer, is what is meant by the 'architecture' of a feed-forward neural network. The specific architecture most suited for a particular problem depends on the problem at hand. While theoretically a single hidden layer is sufficient to model any arbitrary function [95] in practice it may require an infeasibly high number of nodes. [96] Deep networks - which refer to networks with more than two hidden layers - perform more efficiently than shallow networks, requiring a smaller number of training parameters. [97]

Unfortunately, the specific combination of layers and neurons for solving a problem cannot be rigorously derived. For a robust optimisation of the architecture, either manual systematic experimentation or an automated search are necessary, both of which can be time consuming and correspondingly necessary to limit in

scope, especially for large networks or large data sets. Ultimately it is often necessary to rely on a combination of intuition and trial and error. As Goodfellow et al. note in *Deep Learning* [96] on the topic of architecture design:

> Any time we choose a specific machine learning algorithm, we are implicitly stating some set of prior beliefs we have about what kind of function the algorithm should learn. Choosing a deep model encodes a very general belief that the function we want to learn should involve composition of several simpler functions. This can be interpreted from a representation learning point of view as saying that we believe the learning problem consists of discovering a set of underlying factors of variation that can in turn be described in terms of other, simpler underlying factors of variation.

2.5.2 Gradient descent

As noted by Goodfellow et al. in *Deep Learning*, all machine learning models have the same basic components: a model, a cost (or error) function, and an optimisation procedure. The goal is to optimise the parameters of the model so that the cost function, which measures the difference between the model's output and the training data, is minimised. The training data contains the relationship we wish to reproduce; it is composed of examples. The size of each example must equal the number of inputs to the model.

Two common cost functions used in regression problems are the mean squared error,

$$E = \frac{1}{M} \sum_{m=1}^{M} (y_m - t_m)^2 \tag{2.135}$$

and the mean absolute error,

$$E = \frac{1}{M} \sum_{m=1}^{M} |y_m - t_m| \tag{2.136}$$

where M is the total number of examples in a training set, m is each particular example, y is the output of the model, and t is the true value of the example.

The choice of which cost function to use depends on the training data and a prior motivation of whether larger deviations should be more heavily penalised. However, like other aspects of neural network design, for optimisation it is best to test multiple cost functions.

In a neural network, the weights and biases are the parameters to be optimised. Neural networks usually use a gradient-based optimisation procedure because, due to their nonlinearity, a linear equation solver cannot be used. For this reason, it is not guaranteed that a neural network's cost function will reach a global minimum; it may arrive at a local minimum. However, provided this local minimum itself is at a very low value, this is often sufficient to get reasonably accurate predictions.

Gradient descent works by computing the directional derivative of the cost function, $\nabla E(x)$ (where x is the parameter we are optimising) and adjusting x accordingly in the next iteration $n + 1$:

$$x^{n+1} = xn - \alpha \nabla E(x^n) \tag{2.137}$$

where α is the 'learning rate.' The learning rate dictates how large a change can be made to parameter x in a single iteration. The learning rate is yet another choice that a model developer must make when designing and training a neural network. If the learning rate is too small, it may converge very slowly. If it is too large, it may not converge at all (by jumping past the minimum and arriving at a higher, rather than lower, error.)

It is also important to note here that in the case of multiple inputs, gradient descent works most effectively if they are on a similar scale. This is to ensure that the learning step size is appropriate for all the weights. For this reason, the input data is usually scaled to between -1 and 1 or 0 and 1.

The optimisation of a neural network's weights and biases to minimize the error function can be very challenging, for reasons well-explained by Goodfellow et al. To get around some of these difficulties, many different gradient-based optimisation procedures have been developed. For example, gradient descent with

momentum uses an exponentially decaying moving average of the derivatives to assign 'momentum' to a particular 'direction' where the derivative is greatest. This allows faster convergence, but also requires the introduction of a momentum parameter α in addition to the learning rate ϵ. Another method is root-mean-square propagation (RMSProp) which essentially aims to achieve the same goal as gradient descent with momentum - to take steps in the most 'efficient' direction - but does so in a slightly different way. RMSProp belongs to a category of algorithms which assign separate learning rates for each parameter and automatically updates these throughout the training. The update is performed by taking the partial derivative of the loss with respect to each parameter. The learning rates are then individually scaled by an exponentially weighted moving average of past squared values of the gradient. The movement in each dimension is then governed by its own individual learning rate.

The Adam optimizer is similar to RMSProp in that each parameter has its own adaptive learning rate. However, momentum is also incorporated into the update of each learning rate. Adam is therefore considered to benefit from the speed of momentum and the adaptiveness of RMSProp, and is a popular 'default' optimizer for contemporary applications. [98]

2.5.3 Activation functions

While activation functions are used in both the hidden layers and the output layer, we will first discuss the activation functions of the hidden layers of a network. If only linear activation functions were used in a network, it would only be possible to make linear transformations of the input. To reproduce nonlinear functions, it is therefore necessary that the activation functions themselves be nonlinear. For training purposes, which will be discussed soon, it is also necessary that they be differentiable (or, at least, mostly differentiable - more on that later.)

Beyond this, hypothetically any function could be used. However in practice there are some commonly used activation functions. The sigmoid function is a well-known activation function due to its historical use and its common use in teaching

material:

$$g(a) = \frac{1}{1 + e^{-a}} \tag{2.138}$$

Closely related to the sigmoid function, and which was empirically often found to converge faster, is the tanh function:

$$g(a) = \frac{e^{a} - e^{-a}}{e^{a} + e^{-a}} \tag{2.139}$$

The major disadvantage of the sigmoid and tanh functions is that they easily saturate. In other words, large values of a will immediately be transformed to the upper boundary of 1 and small values will immediately be transformed to the lower boundary of zero (for the sigmoid function) or -1 (for the tanh function.) Therefore there is not great sensitivity between two different large values or two different small values. The point of greatest sensitivity is when a is close to zero. This can make training a network using sigmoidal activation functions difficult and inefficient.

The default activation function used in contemporary applications is the Rectified Linear Unit (ReLU):

$$g(a) = \max(0, a) \tag{2.140}$$

This is very similar to a regular linear function, which facilitates training. However, it is nonlinear: for values of $a \leq 0$ it returns 0, so it can still be used to approximate nonlinear functions. Note that it is not differentiable at $a = 0$. However, a value of exactly $a = 0$ is likely to be relatively rare, and so software implementations will typically simply set its derivative to zero.

While it is typical that all hidden nodes in a network share the same activation function, the nodes of the output layer may use a different activation function. For output units, the choice of activation function depends on whether the output of the model is a quantity (for a regression problem) or a class (for a classification problem.) For regression problems, linear activations are used. For classification problems, typically a type of sigmoidal activation is used. Classification will not be further discussed in this thesis.

2.5.4 Training: the back-propagation algorithm

Training a neural network can be considered to be composed of a forward phase and a backward phase. Here we will describe both by following the treatment of Bishop. [95] In the forward phase the inputs are 'fed-forward' from the input nodes through the hidden layers to the output nodes, being transformed at every node, which computes a weighted sum a_j of the inputs

$$a_j = \sum_i w_{ij} z_i \tag{2.141}$$

which is then transformed by an activation function g to produce the node's output z_j:

$$z_j = g(a_j) \tag{2.142}$$

From the output nodes, a final error E is calculated as a function of the difference between the output value and the true value. The error function can be written as a sum of errors of each example in the batch:

$$E = \sum_m E^m \tag{2.143}$$

where m refers to a particular example, and E^m is a differentiable function of the network's outputs: $E^m = E^m(y_1, ..., y_k)$ For the purposes of describing backpropagation, we will consider one example m at a time. As explained earlier, gradient descent requires the computation of the derivative of the error function with respect to our parameters. For a neural network, we wish to evaluate the derivative of E^m with respect to a weight w_{ij} - the weight of the connection feeding into node j from node i in the preceding layer. From equation 2.141 we can express this as

$$\frac{\partial E^m}{\partial w_{ij}} = \frac{\partial E^m}{\partial a_j} \frac{\partial a_j}{\partial w_{ij}} \tag{2.144}$$

At this point we can introduce the notation

$$\delta_j = \frac{\partial E^m}{\partial a_j} \tag{2.145}$$

where δ_j is conventionally referred to as the 'error' of node j. From equation 2.141 we can also write

$$\frac{\partial a_j}{\partial w_{ij}} = z_i \tag{2.146}$$

We can now write equation 2.144 as

$$\frac{\partial E^m}{\partial w_{ij}} = \delta_j z_i \tag{2.147}$$

In words, to obtain the derivative with respect to a specific weight w_{ij}, we only require z_i, obtained from the feedforward phase, and δ_j. We therefore need to compute δ_j for all the nodes in the network. For the output nodes:

$$\delta_k = \frac{\partial E^m}{\partial a_k} = \frac{\partial E^m}{\partial y^k}\frac{dg(a_k)}{da_k} \tag{2.148}$$

For the hidden nodes:

$$\delta_j = \frac{\partial E^m}{\partial a_j} = \sum_k \frac{\partial E^m}{\partial a_k}\frac{\partial a_k}{\partial a_j} \tag{2.149}$$

where the sum runs over all nodes k to which j sends connections. This can be rewritten using 2.141 and 2.142:

$$\delta_j = g'(a_j)\sum_k \delta_k w_{jk} \tag{2.150}$$

Equation 2.150 is the back-propagation formula. From this we can see that the error δ for any particular node is obtained by using the error values of nodes further on in the network. Equation 2.150 can therefore be used recursively backwards through the network.

For a particular training example, the procedure then is as follows:

Algorithm 1 Backpropagation procedure

1. Apply forward propagation to obtain the activations of all the hidden and output nodes.
2. Evaluate δ_k for the output nodes using 2.148.
3. Use δ_k to compute δ_j for each hidden node.
4. Obtain the error derivatives with respect to weight using 2.147.

These error derivatives can then be used in a gradient descent-style minimization algorithm to update the values of the weights. A complete forward and backward

pass of a batch of data, with a corresponding update of the weights, is referred to as one iteration. If the entire training set is used in one iteration, the neural network is said to have trained for one epoch.

2.5.5 Diagnosis and solutions to common training issues

Training and validation sets

Two types of error are generally used to evaluate the effectiveness of a model during development. The first is the training error, which indicates how well the model reproduces the relationship between the training inputs and the expected values. The second is the validation error, which measures how well the model performs on examples not included in its training set. This second type is important because it indicates how well the model can generalise to unseen data.

Underfitting

Underfitting occurs when model is not effectively reproducing the relationship between the training inputs and the expected values. In other words, it has high training error. The validation error will most likely also be high. There are several reasons that training error may be high. The first may be that it simply hasn't converged on a minimum value, in which case longer training or a larger learning rate could reduce it. This would be appropriate if it is clear that the loss is smoothly decreasing with each training iteration.

However, if the training loss behaves discontinuously or begins to increase, it can indicate that the learning rate is too large. This means that the steps taken are too large to arrive at the minimum - they 'step over' it, resulting in larger training error. In this case the learning rate should be reduced.

If the model converges in error but still cannot effectively reproduce the relationship, it may have reached a local minimum. It may be necessary to experiment with different optimizers to avoid this. Also, using a different random initialization of weights can serve to avoid a particular local minimum.

If the training error persists then the model may simply have an insufficiently sophisticated architecture. Similarly to how greater orders of polynomials are

necessary to fit more complex linear relationships, a neural network will require more neurons and/or layers to fit a more complicated relationship. Other hyperparameters such as the activation functions or the error function can also impact how effectively the model can fit to the training data.

Overfitting

Overfitting occurs when a model can exactly or almost exactly reproduce the relationship between inputs and outputs for the training data, but performs poorly on unseen data. This corresponds to a low training error and high validation error, or observing that the validation error increases while the training error continues to decrease.

The first option to avoid overfitting is simply to stop training if the validation error is detected to begin increasing. This is known as 'early stopping.' Similarly to how underfitting can be symptom of an overly simple model, overfitting can be due to an unnecessarily complex model. Overfitting can also be due to a sparsity of training data leading to the model not learning an accurate relationship, and adding more relevant data can be a solution.

Note that the given problems and solutions are only a narrow slice of a vast and continually evolving field. However, what has been described so far should be sufficient in aiding the design and use of a neural network for a basic application.

Chapter 3

Molecular dynamics simulations of local ramp compression

This chapter presents the first section of work completed by the author. We aim to answer this question: can the local physics of ramp compression be adequately approximated by uniform uniaxial compression? By the 'local physics' of ramp compression, we refer to the experience of a single "Lagrangian" cell (a piece of material composed of a group of atoms whose identities do not change) under compression; we ask whether a small molecular dynamics simulation of material under compression may act as an appropriate model for this Lagrangian cell. This makes the following assumption: that the spatial stress gradient during ramp compression is sufficiently low that stress can be approximated as uniform across the cell. This would mean that a molecular dynamics simulation whose atomic coordinates were scaled such that the material experienced an identical strain at every point, and thus an identical stress, could reproduce the the physics of a Lagrangian cell compressed by a ramp wave. This assumption would be most appropriate for ramp waves with long rise times, whose spatial gradients are sufficiently shallow that for a single Lagrangian cell they could be approximated as flat. For steep gradients, this assumption would be less applicable.

The motivation for asking this question lies in the reality of finite computational resources. The NEMD piston simulations presented in this thesis simulate 1.2 microns of copper for 230 picoseconds. This requires approximately 1500 CPU hours, requiring the use of a high-performing computer cluster or supercomputer.

Some of the largest MD ramp simulations have simulated 4 microns of copper for 600 picoseconds. [99] This is on the lower end of realistic laser ramp compression experiments, whose sample thicknesses are typically at least several microns and whose pulse rise times are at least several nanoseconds. [54] [55] At the National Ignition Facility, sample thicknesses can reach the order of 100 microns, with pulse rise times in the tens of nanoseconds. [56] In pulsed-power ramp compression experiments, samples can be measured in millimetres and the pulse length in microseconds.[57] [32] It is therefore highly challenging to reach experimentally realistic temporal and spatial scales in NEMD piston simulations of ramp compression. This motivates the question of whether the relevant physics could be accessed on vastly reduced length scales.

This has been developed for shocks: the Hugoniostat method and the multi-scale shock technique both seek to reduce the problem of scale, with more methods described in section 2.4.6. Similar approaches do not exist for ramp compression. This is partially because ramps are not steady - their constant evolution means there is not a clear set of constraints to follow, unlike the Rankine-Hugoniot equations for shocks. Furthermore, one could argue that an analogous technique is unnecessary for ramp compression: in ideal isentropic compression - of an ideal gas - any rise in temperature should be purely due to the work done in compression.

However, our focus is not on the compression of ideal gases, but solid materials. For compression beyond the elastic limit, any further deformation is irreversible and results in plastic heating. For this reason, ramp compression has been referred to as 'quasi-isentropic' to reflect that it results in a state close to the isentrope when compared to a shock.

It is in the context of plastic deformation that our uniform stress approximation becomes an interesting, or non-trivial, subject of study. We ask this question: could a small uniformly compressed piece of material undergo plastic deformation in a manner similar to a Lagrangian element of a full ramp simulation? Would it evolve comparably in its thermodynamics and microstructure?

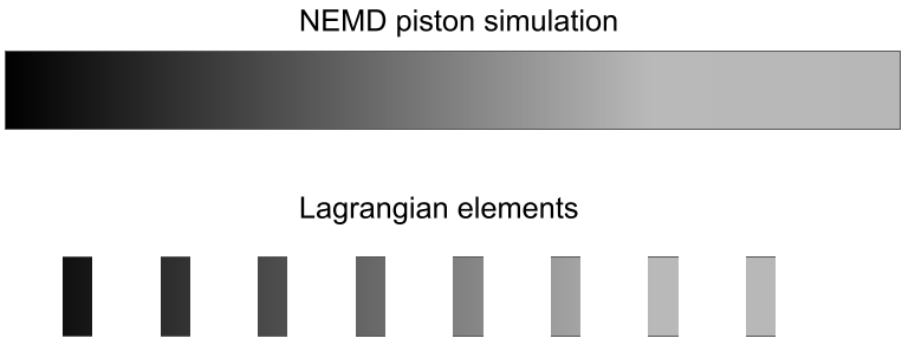

Figure 3.1: Representation of a piston ramp compression simulation as a series of Lagrangian elements.

Essentially we are asking if the fundamental differences between these two cases are significant enough to render this approximation inappropriate. These differences include the isolated nature of a small, uniformly compressed piece of material. For example, while we would expect the adiabatic assumption to hold regarding heat flow, it isn't clear whether the flow of defects into or out of the sample would affect its plastic deformation. Furthermore, it is not certain if the uniform nature of the compression itself would cause the plastic deformation to evolve similarly, with the same level of plastic heating or with the same distribution of dislocation types.

The following piece of work sets out to explore such questions, and to answer them with respect to the specific case the elastic-plastic transition in copper. A simple test was constructed to assess the validity of the uniform stress approximation. To give a quick overview before delving into the details: a large ramp simulation using the piston technique was performed on the target material. The volume, stresses, and temperature of a small section of material - about 10 Å in length, and which we will refer to as our 'Lagrangian element' - was recorded through time. An MD simulation of a small cube of material was uniformly uniaxially compressed at a strain rate matching that of the small section from the full ramp simulation. The evolution of stresses, temperature and microstructure were then compared.

Copper was chosen because it is a well-characterised fcc material with no

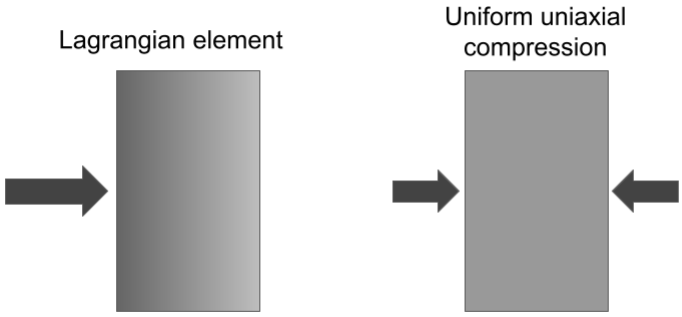

Figure 3.2: The approximation of a Lagrangian element as a uniform, uniaxial ramp compression simulation.

experimentally documented phase changes in the pressure regime we explored, and under shock and ramp compression exhibits a classic elastic-plastic two-wave structure. For these reasons it is also an oft-chosen material for experiments into the fundamental physics of materials under rapid compression. [44] [100] The $\langle 111 \rangle$ orientation was chosen as the direction of compression, avoiding the Bain distortion mentioned in section 2.1.5. The interatomic potential used by Mishin et al. is a commonly used potential for shock and high strain rate studies due to its agreement with shock experiments. [101] The code LAMMPS was used for all MD simulations.

We will first discuss the relevant time scales involved and their impact on shock versus isentropic heating, dynamic compression experiments, and in molecular dynamics simulations. We will then present results for size convergence for the onset of plasticity, a necessary preliminary step. After this convergence testing, we establish baseline behaviour by performing NEMD piston ramp simulations. Finally, we will present the method and results of the uniform stress approximation in MD simulations.

3.1 A note on time scales

The commonly understood difference between a shock and ramp is whether a change is applied suddenly, causing a significant discontinuity in variables, or gradually with no discontinuity. While the continuum explanation of this difference has been

outlined in section 2.2, it is worthwhile to consider the difference atomistically in the context of solids. This has been discussed by Higginbotham et al. [99] in exploring the time scales necessary for quasi-isentropic compression of copper. A shock's resulting heat is caused by a brief 'over-compression' of neighbouring atoms, which can only be achieved if the atoms are compressed on a time scale comparable or smaller than the fundamental material response. In the elastic case, it is therefore possible to quasi-isentropically compress a material sample if the compression is long compared to the vibrational period. In the plastic case, the rate of compression must be low compared to the rate at which dislocations and defects relieve shear stress, and it was found that this could be achieved on picosecond time scales for copper.

However, as noted in section 2.2.3, ramps steepen as they propagate. Longer rise times are necessary to delay shock formation and ensure the greatest possible quantity of material is isentropically compressed; consequently, ramp compression experiments often compress on a nanosecond time scale. The compressions investigated here are at a tenth of this, on the order of tens of picoseconds. This is both to reduce computational cost and to test the limits of the uniform stress approximation. This latter point assumes that as the rate of change of stress decreases, the assumption of spatially uniform stress becomes more applicable; therefore the limits of the assumption are determined by shorter ramp rise times.

3.2 Convergence testing for onset of plasticity

In MD simulations with periodic boundary conditions, the onset of plasticity of a perfect fcc single crystal during uniaxial compression can be affected by the size of the simulation cell. This is due to the fundamental mechanism of how stacking faults are formed during the uniaxial compression of single crystals. Kimminau et al. [4] demonstrated that defects are generated when phonon modes intersect with regions in the reciprocal lattice associated with an unstable dispersion relation. Figure 3.3, produced by Kimminau et al., demonstrates this process: a 20x20x20 cell of

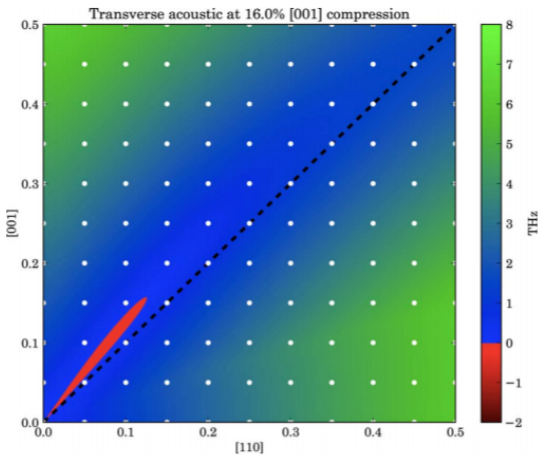

Figure 3.3: Dispersion relation in the $\langle 1\,\bar{1}\,0 \rangle$ plane of reciprocal space for 16% uniaxial compression of Cu along [0 0 1]. The white dots show the allowable phonon modes for a 20x20x20 simulation box. Negative values in the frequency indicate the unstable region. Reproduced from Kimminau et al. [4] with permission from the American Physical Society.

Cu is uniaxially compressed by 16%, which results in the growth of an unstable region in the dispersion relation. The white dots represent the allowable phonon modes for this size of simulation cell, none of which intersect with the unstable region. The unstable region must grow for this intersection to occur, requiring further compression.

Conversely, a larger simulation cell would have more phonon modes within the same region of reciprocal space. There would therefore be greater likelihood of of an unstable phonon mode being able to exist. This then would lead to defect generation, and the material would go plastic at a lower percentage of compression. Of course, at a certain point the density of phonon modes in reciprocal space ceases to result in a significant difference in the point at which the material goes plastic.

It was therefore necessary to conduct a convergence test on simulation box size. These sizes were defined in terms of the unit cell size. The unit cell's geometry was chosen to orient $[0\,1\,\bar{1}]$ along the x-axis, $[\bar{2}\,1\,1]$ along the y-axis, and $[1\,1\,1]$ along the z-axis. A scaling factor of $a = 3.6149$ Å was chosen, corresponding to the lattice constant of copper. The spacing of the lattice points was then calculated according to $a\sqrt{h^2 + k^2 + l^2}$, which equalled 5.11224 Å in x, 8.85466 Å in y (comprising two conventional unit cells in that direction) and 6.26119 Å in z. System sizes in the

rest of this work will be written as XxYxZ, where X, Y and Z refer to the number of lattice spacings in that direction.

Six simulations of copper with dimensions of 5x5x5, 10x10x10, 20x20x20, 30x30x30, 40x40x40 and 50x50x50 were initialised with periodic boundary conditions. They were equilibrated to 300 K over 50 picoseconds using a Nose-Hoover thermostat. The thermostat was then removed and replaced with the microcanonical ensemble. Using the LAMMPS command fix deform, the simulation box's length in [1 1 1] was controlled to follow a realistic compression pathway. Each time the length changed (which occurred every time step) the positions of all atoms were remapped into the new simulation box size. Each simulation was repeated five times with different seed values for the random number generator used to generate the ensemble of velocities.

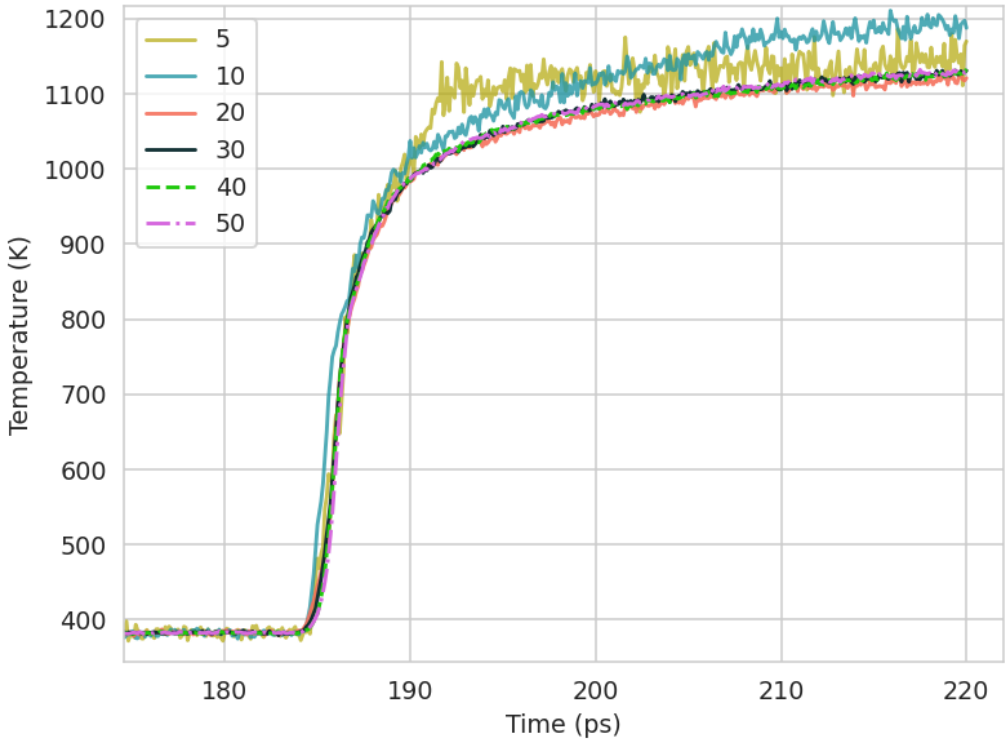

Figure 3.4: Temperature against time, averaged across different velocity seeds for each system size.

Temperature was used to assess the onset of plasticity, as the plastic response is

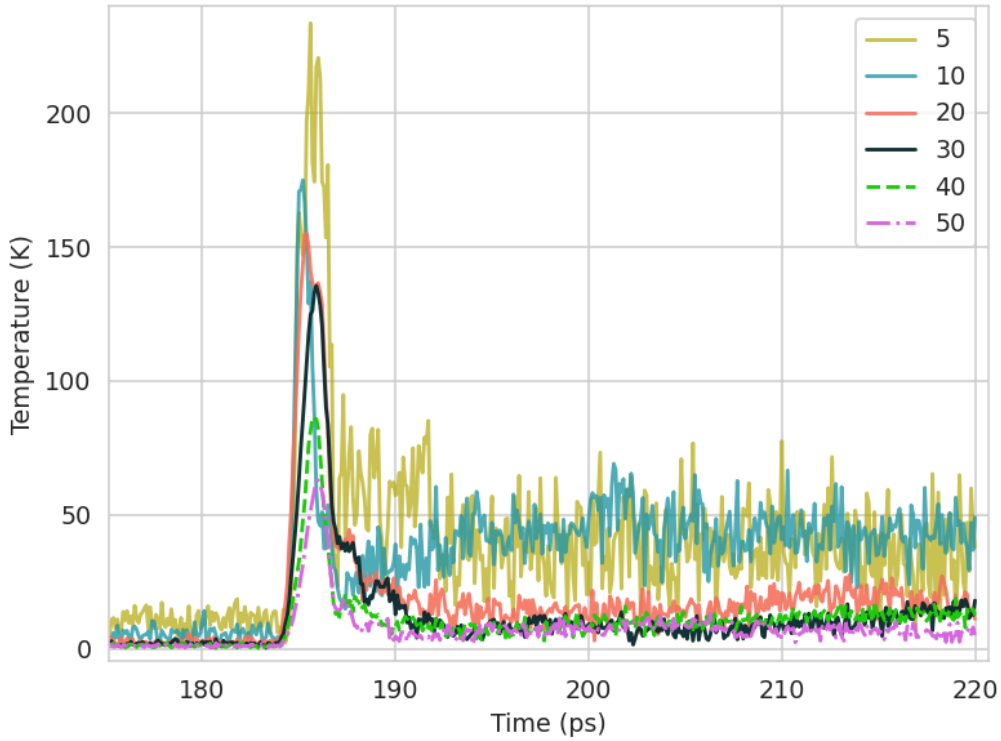

Figure 3.5: Standard deviation of temperature across different velocity seeds against time, for each system size.

accompanied by a sharp rise in temperature due to plastic heating. For each size, the average temperature across velocity seeds is shown in figure 3.4. The standard deviation for each size across velocity seeds is shown in figure 3.5.

From figure 3.4, an appreciably later onset of plasticity cannot be observed from the temperature spikes of the smaller system sizes than for larger systems. However, the smaller systems of 5x5x5 and 10x10x10 do reach a higher peak temperature. This indicates that they experienced a higher shear stress before going plastic, requiring more plastic work to relieve this stress. Perhaps this later onset would be more obvious if not for the greater noise in systems of smaller size, whose average velocity value is more sensitive to outliers due to having fewer atoms overall. The final temperature appears to converge at a size of 20x20x20. However, if we look at the standard deviations of the temperature across simulations, we can see that larger systems are less affected by using different seed values for the velocity ensemble. We

do not reach convergence of the standard deviation with the sizes sampled during the sharp onset of plasticity. However, observe that the standard deviation of the post-spike temperature does appear to converge at 30x30x30; this indicates that although there was greater spread in the timing of the onsets at 30x30x30, this spread was not significant enough to produce a larger difference in the quantity of plastic work than 40x40x40 or 50x50x50. 30x30x30 was therefore determined to be the most appropriate size.

3.3 Establishing baseline behaviour: the piston method

Our baseline method, against which we will evaluate the uniform stress approximation, is an NEMD ramped piston simulation. In this section we outline the method for constructing and running this simulation, as well as explaining a choice we made regarding initial defects in recognition of the effect this has in either suppressing or promoting plasticity.

A simulation of copper whose z-axis was in the $\langle 111 \rangle$ orientation with dimensions 30x30x2000 lattice spacings (corresponding to approximately 153x266x12522 Å) was initialised. This corresponded to 43.2 million atoms and required 1500 CPU hours to run for 230 ps, run in parallel on the Viking computer cluster at the University of York. A cross-section of 30x30 was chosen as the above convergence testing indicates this will not prohibit the generation of dislocations. A region between $z = 0$ and $z = 25$ Å was specified to be the piston, containing 86400 atoms. This was found to be sufficiently thick that no atoms in the rest of the sample escaped. The simulation was equilibrated with periodic boundary conditions over 5000 steps (corresponding to 50 picoseconds) to 300 K using a Nose-Hoover thermostat.

After equilibration, the thermostat was removed and replaced with the microcanonical ensemble. The boundary on the z-axis became shrink-wrapped (non-periodic). The piston's velocity in z was linearly increased from rest to 1100m/s over 100 ps. This final speed was chosen so that it would push the material comfortably

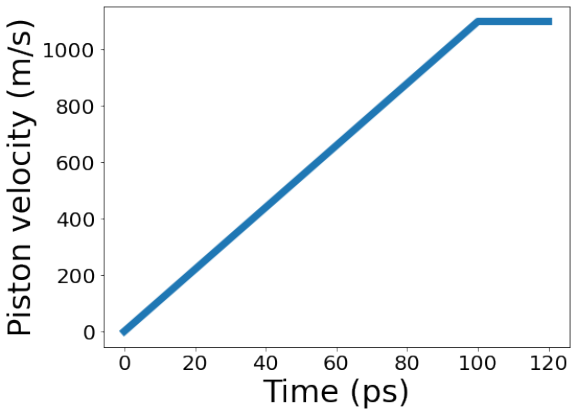

Figure 3.6: Temporal profile of piston velocity for linear ramp.

beyond its elastic limit, the range of which was known through previous use of this interatomic potential. The simulation ran for a further 80 picoseconds with the piston moving at this peak velocity. The spatial profiles for stress, temperature and density are shown in figure 3.7. It can be observed that there is extreme elastic overcompression, followed by a pullback, before plastic deformation. This overcompression can be seen in all three stress components, with the stress in the compression direction ($\hat{z}$) reaching $\sim$ 60 GPa before falling back down to $\sim$ 20 GPa. Equally, the density reaches a peak elastic value of $\sim$ 10.4 g/cm^3 before 'bouncing' back to under $\sim$ 9.5 g/cm^3. With no initial defects present, the plastic response can rely only on the homogeneous nucleation of dislocations, which results in a much later onset of plasticity. While high elastic precursors up to $\sim$73 GPa have been seen in the compression of relatively defect-free single crystal copper [44] there has not yet been observation of a pullback. In other words, while the high stresses associated with elastic overcompression has been observed, the 'bouncing back' to a reduced stress and lower density before the material goes plastic has not been seen. It is also important to emphasise that this overcompression is only seen in materials that are extremely defect-free. The more common scenario is that a crystal will have some pre-existing defects which act to promote plasticity. This results in a lower peak elastic stress. [100]

For these reasons defects were added to prevent the elastic overcompression and pullback. For simplicity, regularly spaced spherical voids were chosen as appropriate

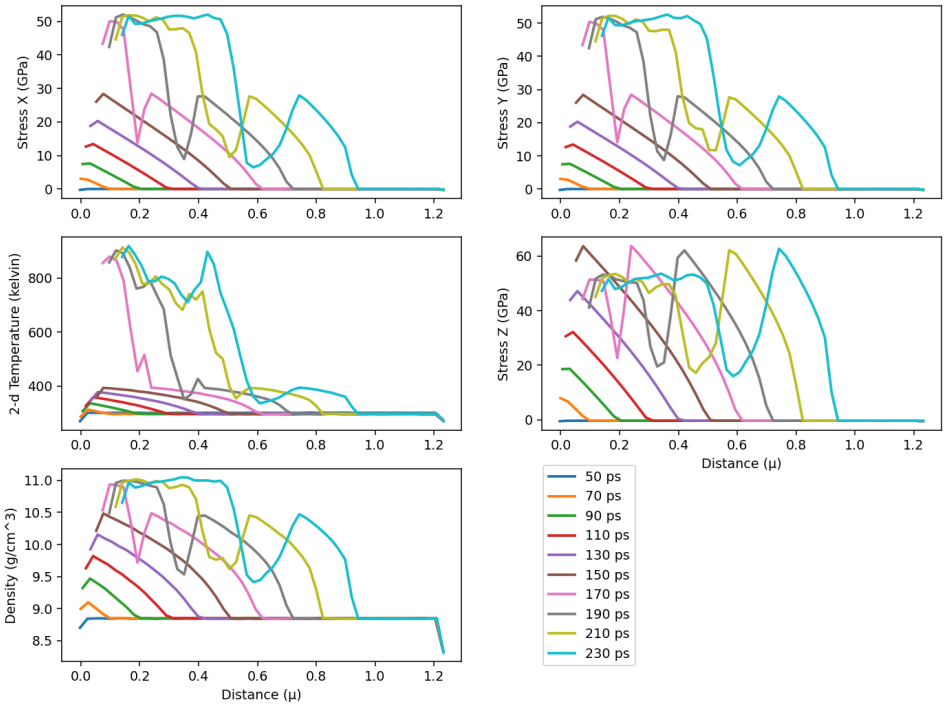

Figure 3.7: Spatial profiles of stress (x, y, and z components), 2-dimensional temperature (x and y components), and density of the compression of Cu using a linearly ramped piston with no initial voids.

initial defects: voids provide effective nucleation surfaces for dislocations [102] [103] [104] and are simple to add and alter. Accordingly, after equilibration and before compression, spherical voids of radius ∼12.5 Å were seeded at regularly spaced intervals of ∼313 Å throughout the main sample. In total, this resulted in the deletion of 0.034% of the sample's atoms, and was therefore a very minor reduction in the bulk material density. Other void sizes with different spacing could certainly be used, but the computational expense of these simulations prohibited a systematic exploration - the priority was to find a size and regularity that encouraged defect growth with minimal impact on the total density. The result of adding these voids on the stresses, density and temperature are shown in figure 3.8.

To record local data from this second simulation using voids, two sections - each 10 Å thick - were chosen at initial distances of z = 2500 and z = 4000. While distances into the sample were used to initialise these sections, they were defined by

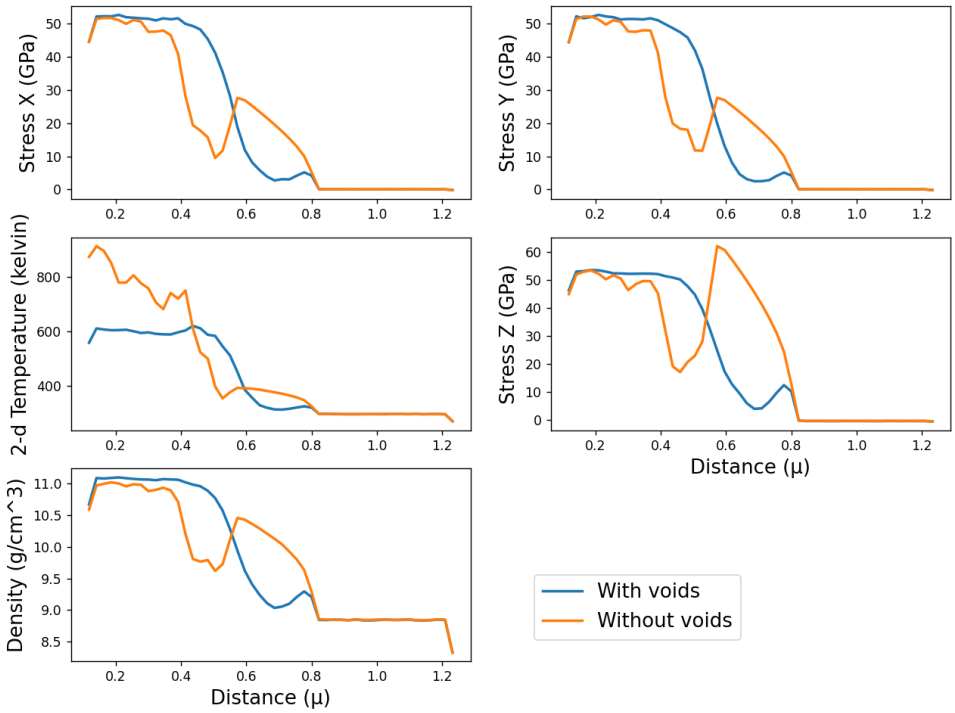

Figure 3.8: A comparison of spatial profiles at 210 ps resulting from the compression of Cu using a linearly ramped piston for simulation with no initial voids and one with initial voids of radius ∼12.5 Å spaced every ∼ 313 Å through the sample.

their composition: of a particular group of atoms whose identities did not change throughout the simulation. The boundaries and the centre of mass of these sections were therefore mobile, and for this reason we name these sections our 'Lagrangian' elements, since they can be regarded to be analogous to a Lagrangian cell as defined in a continuum simulation. Their volume, stresses, and temperature were recorded every picosecond so that their evolution could serve as the baseline against which the uniform stress approximation could be evaluated.

3.4 Method to test the uniform stress approximation

We will now describe the second simulation type. As discussed in the beginning of this chapter, this is the uniform, uniaxial compression (UUC) of small simulation cell. We wish to evaluate whether this simulation can approximate the

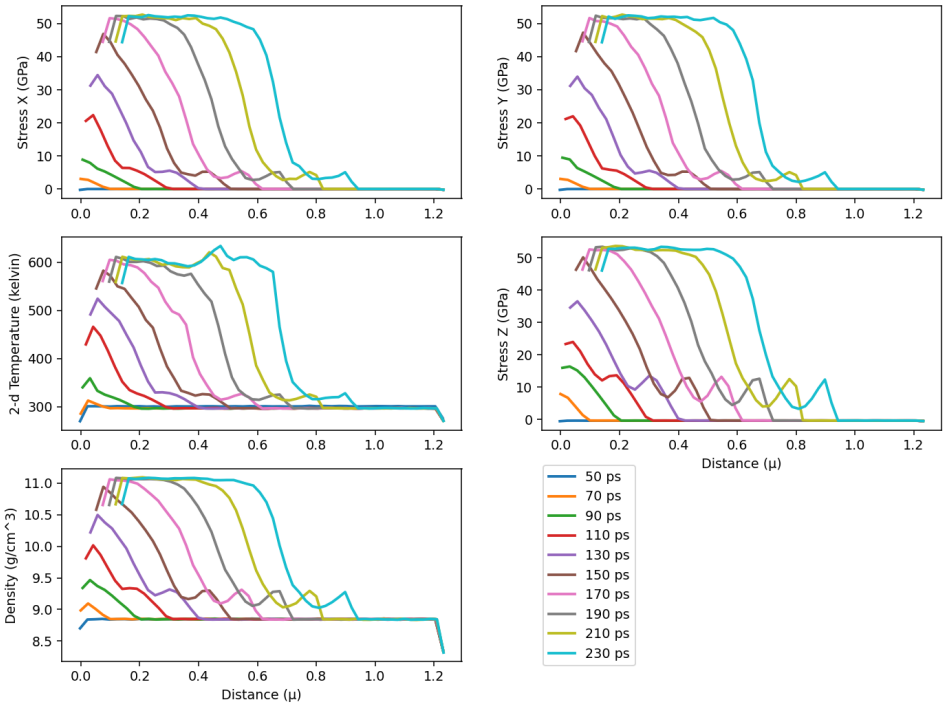

Figure 3.9: Spatial profiles of stress (x, y, and z components), 2-dimensional temperature (x and y components), and density of the compression of Cu using a linearly ramped piston with initial voids of radius ∼12.5 Å spaced every ∼313 Å through the sample.

thermodynamic and microstructural evolution of a Lagrangian element in the NEMD piston ramp simulation if compressed at the same rate.

A cuboid of copper whose z-axis was in the ⟨111⟩ orientation was initialised. It had dimensions of 30x30x30 in lattice spacings, equivalent to approximately 153x266x188 Å and containing 648000 atoms, with periodic boundaries. This size is 1.5% of the baseline piston simulation and resulted in an almost linear reduction in total simulation time, from 1500 to 26.7 CPU hours. It was equilibrated to 300 K over 5000 time steps of 0.01 picoseconds using the Nose-Hoover thermostat. After equilibration, the thermostat was removed and replaced with the microcanonical ensemble.

As mentioned in the previous section, we had previously recorded data from a single Lagrangian element in the baseline piston simulation, which included its

relative volume over time. In the UUC simulation, we controlled the length in z such that it experienced the same rate of change in its relative volume over time. Each time the length changed, the positions of all atoms were remapped into the new simulation box size. This ensured that every point in the material experienced equal strain. The stresses, 2-dimensional temperature and Voronoi volume were recorded.

It was judged that this simulation of a perfect, defect-free single crystal would likely be vulnerable to elastic overcompression, as seen in the previous section. And so, similarly to the NEMD piston simulation, spherical voids were chosen to act as dislocation nucleation sites. Consequently, four additional otherwise-identical simulations were performed where a void of varying sizes was added at an arbitrary location after equilibration and before compression. The void radii tested were 3Å, 7Å, 8.5Å, and 10Å.

3.5 Results and discussion of the uniform stress approximation

Figure 3.10 shows the x, y, and z stress components, the two-dimensional temperature, and the Voronoi volume of both the uniform uniaxial compression simulation and the piston-ramped Lagrangian element. The average Voronoi volume of the atoms matches completely - this is expected as the volume is the controlled parameter. First let us compare the voidless 30x30x30 simulation with the ramped group, and the clear differences in the stresses and the temperature. Most notably, the z-stress of the 30x30x30 box greatly overshoots to 70 GPa before experiencing a dramatic pullback. This corresponds in time to the sharp rise in temperature, which previously had lagged below that of the ramped element but now overshoots by more than 400 kelvin. The pullback and the temperature spike signify the onset of plastic deformation, which clearly occurs approximately 25 ps after that of the ramped element. This behaviour can be explained by the lack of initial defects in the simulation cell, which suppressed the onset of plasticity as dislocations could

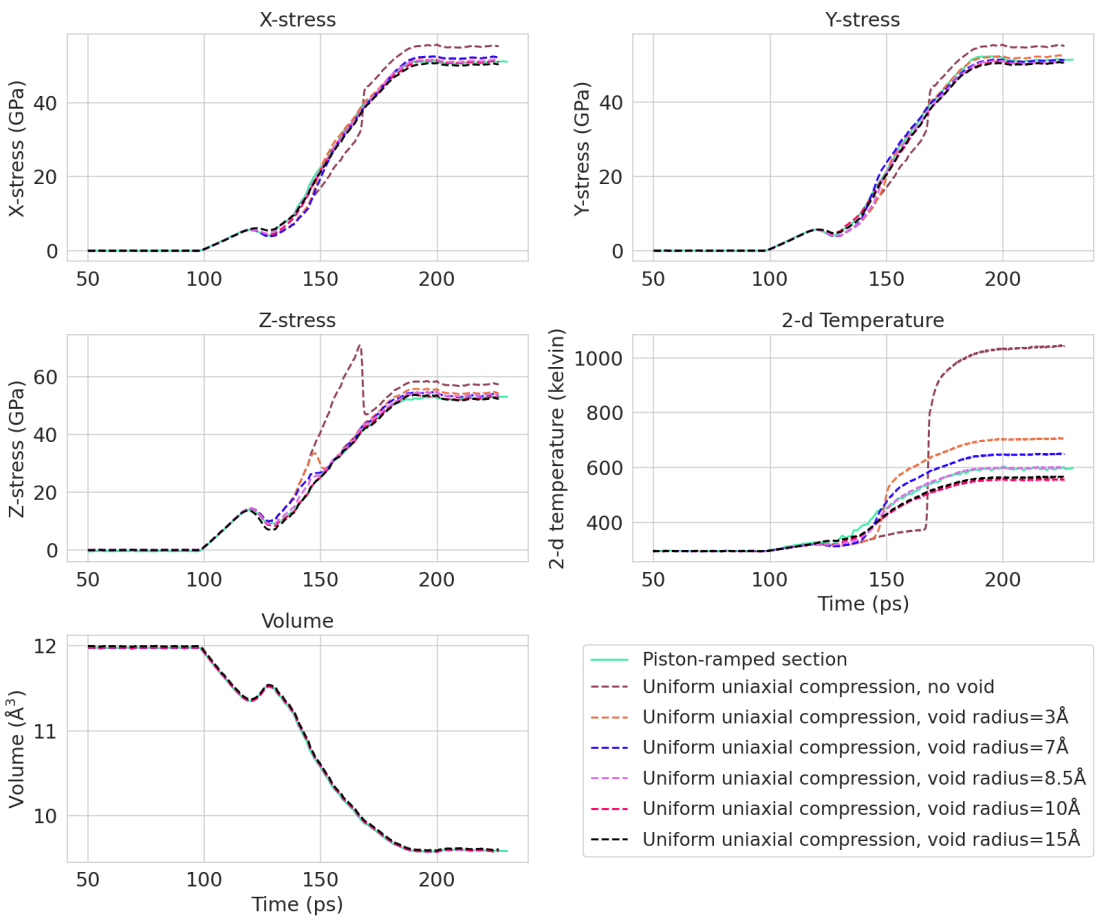

Figure 3.10: Comparison plots of stress components, temperature and volume against time for the uniform uniaxial compression simulation and the piston-ramped Lagrangian element.

only be generated through homogeneous nucleation.

It therefore occurred at a later time and with greater vigour: since dislocations could not be generated sufficiently quickly to promote plastic relaxation, the system reached a higher peak shear stress. More plastic work was therefore required to relieve it when plastic relaxation finally occurred, resulting in greater heating. As mentioned in section 2.1.13 describing Orowan's equation, the average Burgers vector is likely to remain constant for the same material compressed along the same axis, and so the product of average dislocation velocity and dislocation density must be greater. The piston simulation, in contrast, was not only seeded with voids which acted as nucleation sites for dislocations, but benefited from the flow of dislocations from plastic areas. It was therefore not overdriven into a state of such high shear

stress. Its elastic overcompression, while still evident by the small pullback in the z-stress at 125 ps, is less dramatic; it reaches a peak elastic stress of ∼13 GPa rather than ∼70 GPa.

To mimic this swifter onset of plastic deformation, voids of varying sizes were seeded in the uniform uniaxial compression simulation, with the purpose of acting as a nucleation site for dislocations. These had a negligible impact on the density of the bulk material; the largest void size of 15 Å only reduced the total number of atoms by 0.183%. Figure 3.10 demonstrates that this takes effect as intended: with larger void sizes the temperature rise associated with plastic deformation is lesser and happens sooner. The UUC simulation with a void radius of 8.5 Å appears to most closely match both the stress and temperature evolution of the ramped element - though note that the temperature of the latter is greater at the onset, after which the discrepancy reduces. This may be due to the flow of dislocations into the element from upstream, a process that would be absent from the UUC simulation.

A reader may question why the void radius used in the NEMD piston simulation is 12.5 Å, while the void radius that matches most effectively in the UUC simulations is 8.5 Å. This question assumes that there is reason for us to expect a one-to-one correspondence, but this assumption is tenuous: there is likely a more complex relationship between the two parameters, the exhaustive investigation of which was deemed beyond the scope of this thesis. As shown in the convergence testing for box size above, the mechanism of dislocation nucleation is highly sensitive to initial conditions. We can therefore expect not only the void size to have an impact, but also the void shape, placement, and if there are any other voids or defects in the vicinity; the precise mechanisms of dislocation nucleation and multiplication from different defect types and configurations is a rich area of study. [102] [105] [103] [104] [106] [107] [108] [109] However, a comprehensive understanding of the differences in nucleation and multiplication mechanisms from both simulations should certainly be sought in future work. In this context, we use void size as a tuneable plasticity factor for increasing or decreasing the available surface area for heterogeneous nucleation

in a general way, accepting that it is a simplified approach.

To assess that the plasticity mechanisms were similar in both methods, the microstructure was evaluated for both the UUC simulation and the NEMD ramp simulation at the onset of plasticity. This was accomplished using the Dislocation Extraction Algorithm (or DXA) developed by Stukowski et al. (described in section 2.4.7). Our element of 10 Å thickness could not be used in the dislocation analysis because the length of dislocations typically surpassed the length of the element itself. Therefore a wider section which encompassed the original element was used, from $z = 2400$ to $z = 2600$ Å. While this greater size makes direct comparison slightly more challenging as it renders the uniform stress approximation less applicable - and furthermore whose average Voronoi volume does not exactly match that of the element - it is still an effective method for broadly comparing dominant plasticity mechanisms.

Consequently, DXA was performed on both the 30x30x30 uniformly uniaxially compressed material (roughly 170 Å in length) and this larger section of piston-ramped material (200 Å in length) every five picoseconds. All the previous void sizes (3Å, 7Å, 8.5Å, and 10Å) were analysed for the 30x30x30 simulation. The total dislocation length through time for each dislocation type, divided by the number of atoms in each sample, is shown in figure 3.11. A discussion of the different dislocation types presented here is given in section 2.1.13.

The first observation to note from figure 3.11 is that the ramped section and uniformly uniaxially compressed sample, regardless of void size, have very similar distributions of dislocation types. The dominant dislocations are Shockley partials, whose total length is an order of magnitude greater than the second greatest, stair-rods. Since Shockley partials are the dominant dislocation mechanism for fcc metals, this is not an unexpected result, but the similarity of the onset and magnitude of Shockley partials in both samples, particularly as the void size is increased for the 30x30x30 simulation, demonstrates the similarity in the dominant plasticity mechanisms for both.

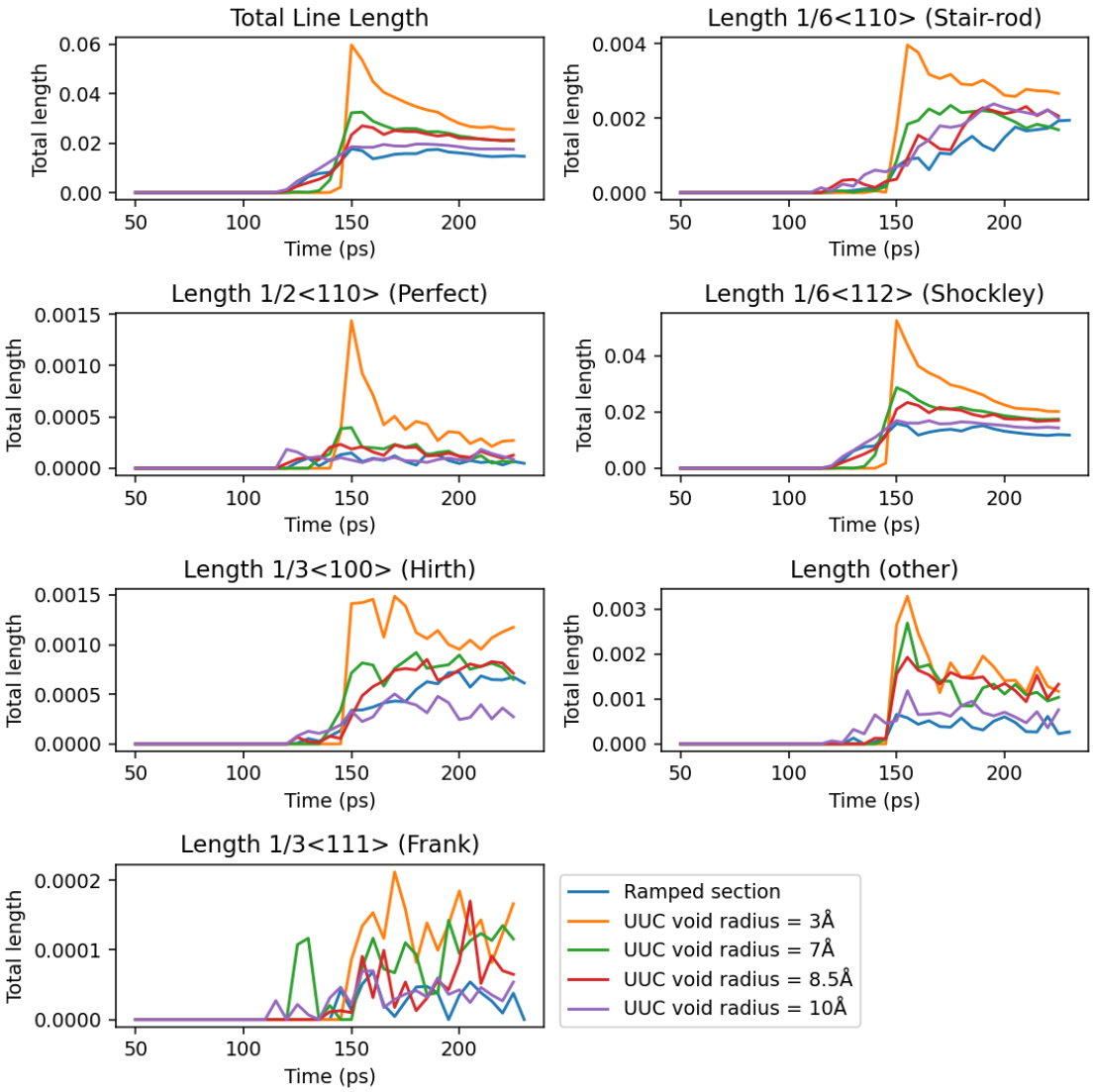

Figure 3.11: Total dislocation lengths through time, categorised by dislocation type and divided by the number of atoms in each sample.

We can also observe the impact of adding a void to the uniformly uniaxially compressed sample. The growth of all dislocation types is delayed with a smaller void size, with a delay of approximately 20 picoseconds between void radii of 10Å and 3Å. This is in accordance with the delayed rise in temperature in the thermodynamic results of figure 3.10. Furthermore, when the delayed plastic response does eventually occur, it does so in a shorter period of time and reaches a greater peak dislocation length. This length then decreases as the sample is held at a constant volume and approaches a hydrostatic state.

It is interesting to interpret this behaviour through the lens of Orowan's equation

(given in equation 2.48 in section 2.1.13) with the stipulation that in this complex and dynamic environment the dislocation densities and velocities will not be constant and thus beyond Orowan equation's purview. The initial plastic strain rate is higher for simulations with small or no voids. The assumption of the average Burgers vector remaining constant across scenarios is also demonstrated to be reasonable here due to the similar distribution of dislocation types. Since the plastic strain rate is the product of the Burgers vector, the average dislocation velocity and the dislocation density, the increase in plastic strain rate must therefore be due to a greater product of these latter two values. We can observe that the ramped section has lower total dislocation length density than the simulation with void size 8.5, despite experiencing a similar quantity of plastic heating (and so presumably a similar plastic strain rate.) If we assume that the overall dislocation length is proportionate to the dislocation number, we can infer that the uniform uniaxial compression simulation generates more dislocations with lower average velocity than the ramped section, which correspondingly has fewer dislocations that travel faster. However, this is likely an oversimplification because the densities and velocities are dynamic, and the portion of plastic work which is converted to heat is itself condition and history dependent. [110]

The same test was conducted for the second element (at an initial distance of $z = 4000$) the results of which are shown in figure 3.12. As this element was further from the piston, we can expect that the ramp wave will have steepened by some degree by the time it was reached, rendering the uniform stress approximation less accurate. And indeed, while there is still broadly good agreement, there is slightly greater discrepancy in the temperature during the elastic and early-plastic compression stages. This shows that as a ramp steepens into a shock, the uniform stress approximation rightly becomes less and less applicable.

To evaluate the technique for a different compression pathway, a quadratic ramp was also tested. For the ramped piston simulation, a piston's velocity was quadratically increased with time from rest to a final value of 1100 m/s. Voids were

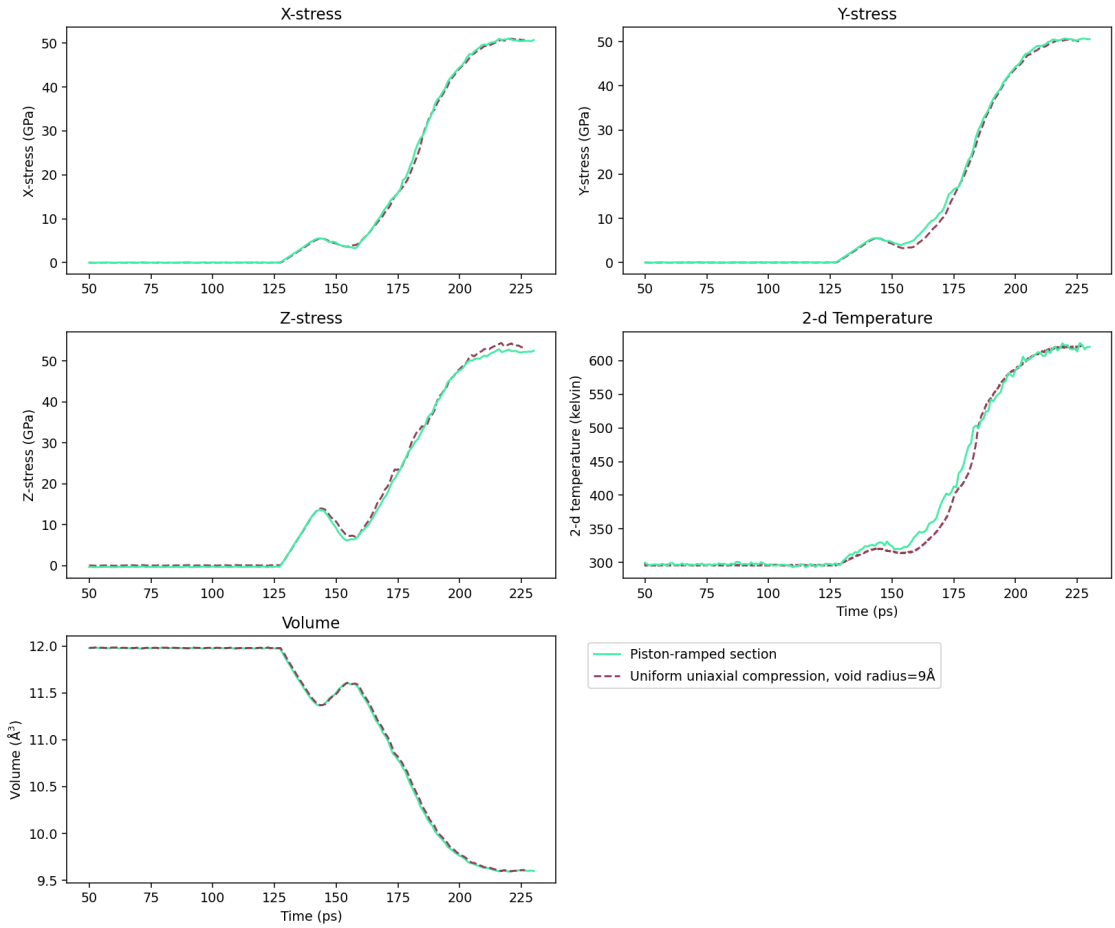

Figure 3.12: The stress, temperature and volume data from section 2 of the linearly piston-ramped material, compared to a 30x30x30 uniformly uniaxially compressed sample whose volume was reduced at the same rate.

again added with the same size and regularity. The results are shown in figure 3.14. It can be seen that the uniaxial uniform compression effectively reproduces the stress and temperature. However, there are two aspects to note. Firstly, we can once again see the greater temperature in the ramped element at the onset of plasticity which levels out to the same final temperature. Again, we believe this is due to the flow of dislocations into the element from the adjacent compressed region. Secondly, it can be observed that otherwise identical simulations with different initial velocity seeds can result in significantly different final temperatures. This demonstrates the stochastic nature of plasticity, and further shows the utility of performing these smaller simulations; it is more practical to sample different possible plastic responses with 30x30x30 simulation boxes than in large piston simulations.

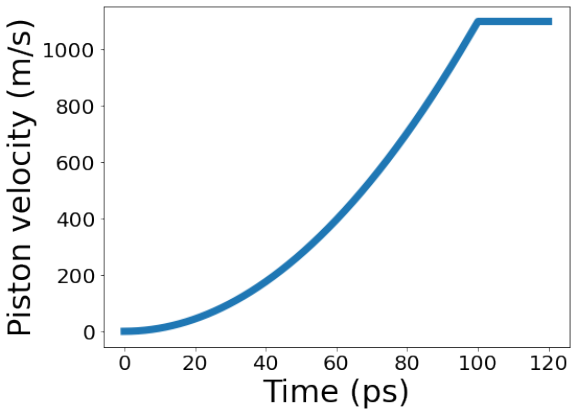

Figure 3.13: Temporal profile of piston velocity for quadratic ramp.

3.6 Conclusion of chapter 3

It has been demonstrated that small box approximations of uniform uniaxial compression can effectively recreate the thermodynamic response to elastic-plastic deformation of a ramp wave for a single Lagrangian cell of the target material. It should be noted that while this behaviour may be representative of other elastic-plastic fcc metals, all testing was performed only on copper in the $\langle 111 \rangle$ orientation using the interatomic potential by Mishin et al. It should also be noted that since a material's plastic response is highly dependent on its initial microstructure, a tunable parameter is needed to influence the onset of plastic deformation. The size of an initial void in the simulation was found to be a highly effective method of control over the onset and nature of the elastic-plastic transition.

This method allows the study of the effect of different ramp compression pathways on materials, without the need to perform a full piston-ramp simulation. It benefits from only needing to increase the simulation time without a corresponding increase in simulation space in order to observe the effect of longer ramps. While the cost of simulating a ramp through 1.2 microns of material for 230 ps required approximately 1500 CPU hours, simulating 1.5% of that amount required only 26.7 CPU hours.

However, this method does require that the specific compression pathway of the material is already known. As has already been demonstrated, a linearly ramped

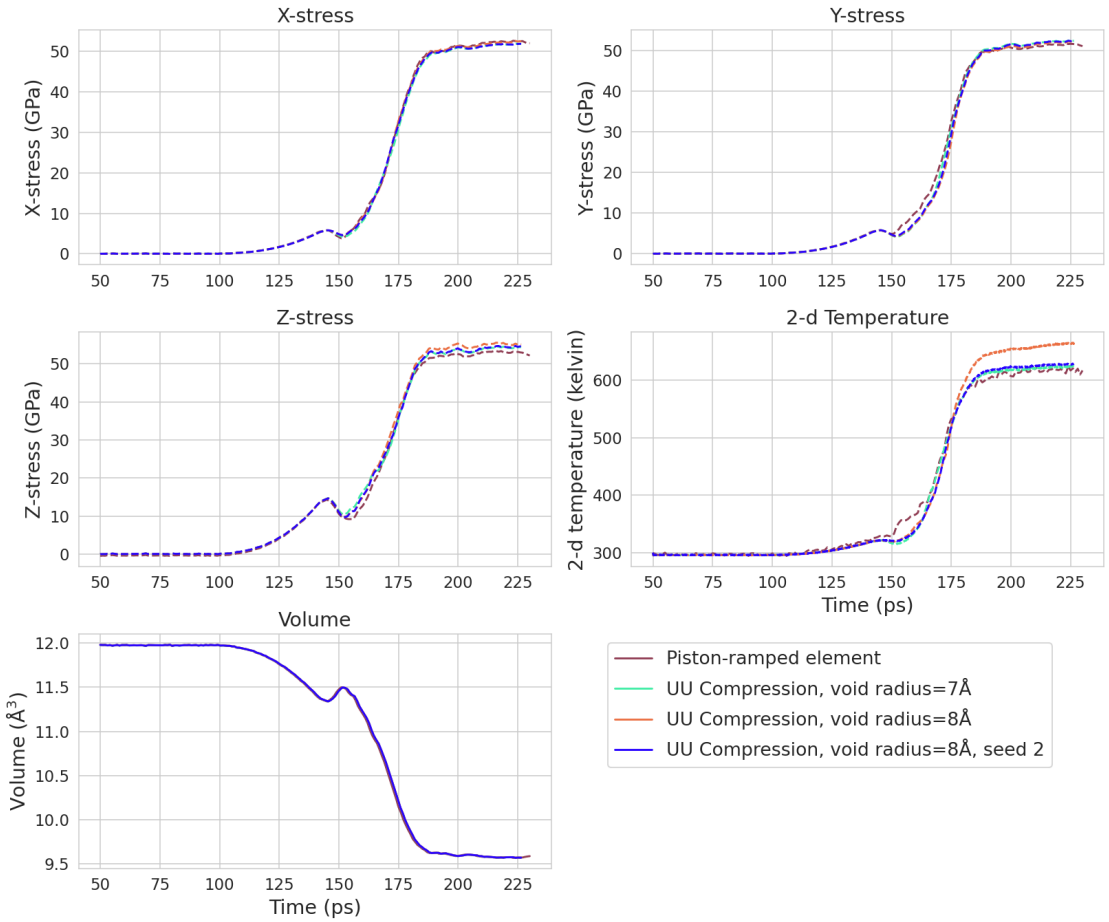

Figure 3.14: The stress, temperature and volume data from section 1 of the quadratically piston-ramped material, compared to a 30x30x30 uniformly uniaxially compressed sample whose volume was reduced at the same rate.

piston does not result in a section of material whose volume changes linearly beyond the elastic limit. The next chapter details an extension of this work which considers wave propagation.

Chapter 4

Development of multi-scale model

Chapter 3 detailed the MD investigation into whether small, uniformly, uniaxially compressed (UUC) samples can adequately reproduce the thermodynamic and microstructural results of ramp compression for an elastic-plastic fcc metal. It was found that by seeding an initial void and controlling its size, the stress and temperature response very closely matched that of a piston ramp simulation.

In this chapter, we seek to go beyond this finding. If the relevant thermodynamic and material behaviour of a small section of an NEMD ramp compression simulation can be reproduced by small simulations, it raises the question of whether an entire NEMD profile could be reproduced without prior knowledge of how each cell is compressed. This would require a continuum code to link these otherwise independent Lagrangian cells together and handle the growth and propagation of the ramp wave. It would also require a surrogate model or emulator in place of the small MD simulations, as running these simulations on-demand with a continuum code would be highly impractical.

Linking a continuum model to an atomistic model via a surrogate model could be considered a classic multi-scale approach, as discussed in the introduction: a hydrocode whose equation of state/constitutive model is handled by an MD-emulator (in this case a feedforward neural network) of material behaviour. Figure 4.1 depicts its basic functionality and structure. Unlike related techniques [111] the main distinction of this method is the independence from an intermediary analytic plasticity model and the ability to handle arbitrary ramp compression pathways.

The goal of this chapter is to present a proof of concept of this approach. We describe the design methodology and development of such a tool, and evaluate its performance in reproducing an NEMD ramped piston profile. Our aim is not to develop a fully optimised tool or procedure, but we make clear the benefits of this approach: due to the vastly reduced computational expense, as discussed in chapter 1 and as will be further discussed later in this chapter, ramp compression can be simulated on far greater length and time scales than in MD alone. This means that a large number of different ramp compression pathways can be tested without running a set of expensive NEMD simulations.

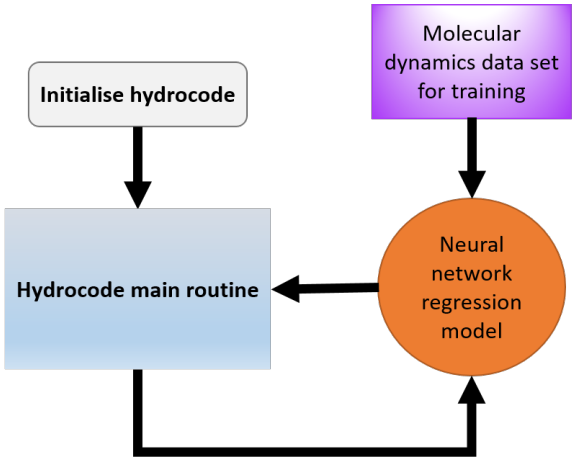

Figure 4.1: Diagram demonstrating the basic functionality and structure of the developed multi-scale modelling tool.

The contents of this chapter will address the development of each of these three major components. The performance of the final multi-scale simulation tool will then be assessed in its ability to reproduce the spatial profiles of an MD piston simulation. The specific case under study will be the ramp compression of copper.

4.1 Multi-scale model formulation for one-dimensional hydrocode

We will first present the hydrocode, as it provides the constraints for the neural network and the MD data set. Wilkin's one-dimensional formulation for elastic-plastic flow [66] was chosen as an appropriate base for the continuum model, as the neural network regression model could easily replace its equation of state and

plasticity model while keeping most of the wave propagation mechanics intact. His original finite difference program is presented in background section 2.3.7.

The amendments made to Wilkins' model were largely subtractive, as our interests lay in replacing the hydrocode fundamentals of an equation of state and plasticity model with an MD-informed emulator which could handle both the thermodynamic and material properties. The benefit of one tool for both lies in the reality that they are not truly separate; they both emerge from first principles. The key question then was how to compute the new stresses using the regression model. The variable nature of plasticity, which could occur at higher stresses if the compression was rapid or if the material was a perfect or nearly-perfect crystal, required that some knowledge of history be built-in. In other words, the regression model needed to know whether the cell had already gone plastic. We were not interested in using a separate plasticity model or mechanism as that seemed to subvert the original utility of using MD data. In an MD simulation, as in reality, plasticity arises from the movement and slippage of atoms rather than an on-off switch when a certain macroscopic criterion has been reached, and the onset and development of plasticity can be intuitively identified from the stresses. In particular, they will cease to increase linearly with strain and will lose strength, approaching a more hydrostatic state.

For this reason, the new stresses were computed by using the stresses from the previous time step, along with the current strain rate and the void radius. The strain rate is necessary to inform the regression model that compression is occurring, as well as building in some rate dependence. The void radius, which is a tuneable parameter set by the user and stays constant throughout the simulation, affects the onset of plasticity and can be considered to be a measurement of the initial crystal imperfection of the material. The void radius parameter is likely the greatest obfuscation in the formulation, as it essentially collapses a great diversity of possible defect types, sizes and configurations which could affect the onset and development of plasticity into a single parameter. This is largely for convenience rather than

necessity; a model could certainly be built from a data set which contains a greater diversity of defects, and for which different defect types and characteristics could be inputs to the regression model. But for the purposes of this work, a single parameter was found to be both simple and effective.

In order to be independent of time step size, the outputs of the regression model were required to be the rates of change of the stresses. The hydrocode then makes a simple linear approximation, calculating the new stress as this rate of change multiplied by the time step. If the time step is too large compared to the time interval of the MD data, this could lead to inaccurate predictions by the regression model. While this has not yet been an issue, it is certainly something to consider and investigate if a simulation is displaying unexpected behaviour.

Computation of the internal energy became redundant as it was no longer used to compute the pressure, and conservation of energy is implicit in the MD data set. Temperature can be considered a more useful variable to access since it can be more directly compared to experiment, so its computation via the regression model effectively replaced the internal energy. It was added as an input to the regression model, and its forecasted rate of change was an output, very similarly to the stresses. It was judged to be reasonable to add the temperature to the same feedforward network as the stresses, as the temperature itself can both signpost plasticity and is sensitive to the rate of the plasticity.

The most minor change in the formulation involved re-defining stain rate as engineering rate rather than true strain rate. This was done purely for consistency with the MD data set, which happened to use the engineering strain rate (though this isn't required by any means) and has no physical consequence. It would perhaps be more appropriate to think of it simply as 'deformation rate', though we have chosen to keep the notation and terms of the Wilkins' model to make the significant changes clearer.

The fundamental equations then are the equation of motion:

$$\frac{\rho \dot{U}}{V} = \frac{\partial \Sigma}{\partial r} \tag{4.1}$$

where ρ is the density, U is the velocity, V is the relative volume, and Σ is the total compressive stress. The conservation of mass:

$$\frac{dM}{dt} = 0 \tag{4.2}$$

The strain rate:

$$\dot{\epsilon} = \frac{dV}{dt} \tag{4.3}$$

The longitudinal and transverse stresses:

$$\sigma_L(t+dt) = \sigma_L(t) + \int_t^{t+dt} \frac{d\sigma_L}{dt} dt \tag{4.4}$$

$$\sigma_T(t+dt) = \sigma_T(t) + \int_t^{t+dt} \frac{d\sigma_T}{dt} dt \tag{4.5}$$

and the temperature

$$T(t+dt) = T(t) + \int_t^{t+dt} \frac{dT}{dt} dt \tag{4.6}$$

where

$$\frac{d}{dt}\begin{pmatrix} \sigma_L \\ \sigma_T \\ T \end{pmatrix} = \boldsymbol{f}(\sigma_L(t), \sigma_T(t), \dot{\epsilon}, T(t), \nu) \tag{4.7}$$

and $\boldsymbol{f}$ is computed by the neural network regression model. ν is the adjustable void size parameter. The total compressive stress Σ_L is then moderated by the artificial viscosity q

$$\Sigma_L = \sigma_L - q \tag{4.8}$$

where q is

$$q = 4\frac{\rho_0}{V}\left(\frac{\partial U}{\partial r}\right)^2 (\Delta r)^2 + \frac{\rho_0}{V}\sqrt{\frac{P}{\rho}}\left(\frac{\partial U}{\partial r}\right)\Delta r \tag{4.9}$$

and where $P = (\sigma_L + 2\sigma_T)/3$ is the pressure and Δr is the grid spacing. These fundamental equations were rewritten as finite difference equations, using the same scheme as Wilkins in section 2.3.7, and coded in C++ in collaboration with Dr. Andrew Higginbotham.

4.2 MD data set of localised ramp compression

4.2.1 Data set design methodology

Let us first review the design methodology and approach that was used to develop our data set. The data set construction is a fundamental parameter in the performance of the neural network, particularly in the case where it is acting as an emulator for a larger multi-scale model. It is well-known that a neural network, or any machine learning algorithm, should not be used to make predictions outside the range of its training. In a standalone neural network model, it is only necessary that the user of the model work within these constraints. A neural network that acts as a component of a larger numerical model, however, may be subject to requests by the larger model that takes it to the edge or outside of its training regime. If the neural network makes a poor prediction in response, this can prompt the numerical model to move even farther from the training regime, resulting in a feedback loop of poor predictions and ultimately poor and non-physical performance of the numerical model. There are two approaches to prevent this situation. One is to enforce constraints on the numerical model such that it will not make 'unreasonable' requests on the neural network. But this should only be a last resort in limited situations. After all, if the numerical model itself is working correctly and is an appropriate tool for the simulation scenario, why should its requests be unreasonable? The initial focus should therefore be on the second approach: developing a training data set that anticipates all likely scenarios and combinations of features, such that it is unlikely to be put in a situation for which it has not trained. Furthermore, in addition to expanding the boundaries of the training region, the region itself must be sufficiently densely sampled. Some areas may require higher density than others due to greater sensitivity to multiple variables at that point.

For this reason, the training data set itself could be considered the most challenging parameter to optimise. Unlike the neural network architecture, whose calibre can be quantified by error convergence, the development of the data set relies on the developer's own sense and foresight on what is important for the model

to know, and know well. Its calibre can only be evaluated by its performance in combination with the numerical model.

It is therefore important to note that the following data set is not unique; other data sets with varying proportions of training scenarios, or even a sparser training set overall, could be equally or more effective. On a related note, some characteristics of the data set are artefacts of development; for example, a higher density of sampling around a particular scenario chosen for testing. While in excess this could in principle contribute to the neural network over-prioritising this scenario at the expense of others, this has not been found to occur on a noticeable scale for the given data set.

In order to judge with which situations the neural network ought to be familiar, it is important to recollect the information it receives, which is the longitudinal and transverse stress, strain rate, temperature and void size. The goal of covering a wide range of scenarios is thus reduced down to the problem of trying to sample as many different combinations of these variables as possible, with the requirement that they must be from a UUC simulation.

4.2.2 Data set construction

A data set was constructed from 466 simulations of uniform uniaxial compression, which together provided 96885 data points (measurements were recorded every picosecond of each simulation's duration.) The set-up of all simulations was identical to the initialisation and equilibration stages from the UUC simulations of chapter 3.

A void was seeded at an arbitrary location in the cell, with a radius between a specified range. As mentioned in chapter 3, the size of this void acts as a strength parameter, playing an important role in determining at what point the material goes plastic. Using the LAMMPS command fix deform, the simulation box's length in z was controlled so that it updated every time step, which was 0.01 picoseconds. The simulation was first held at constant volume for 15 picoseconds. Its length was changed to follow a compression pathway - either linear, quadratic, cubic, radical,

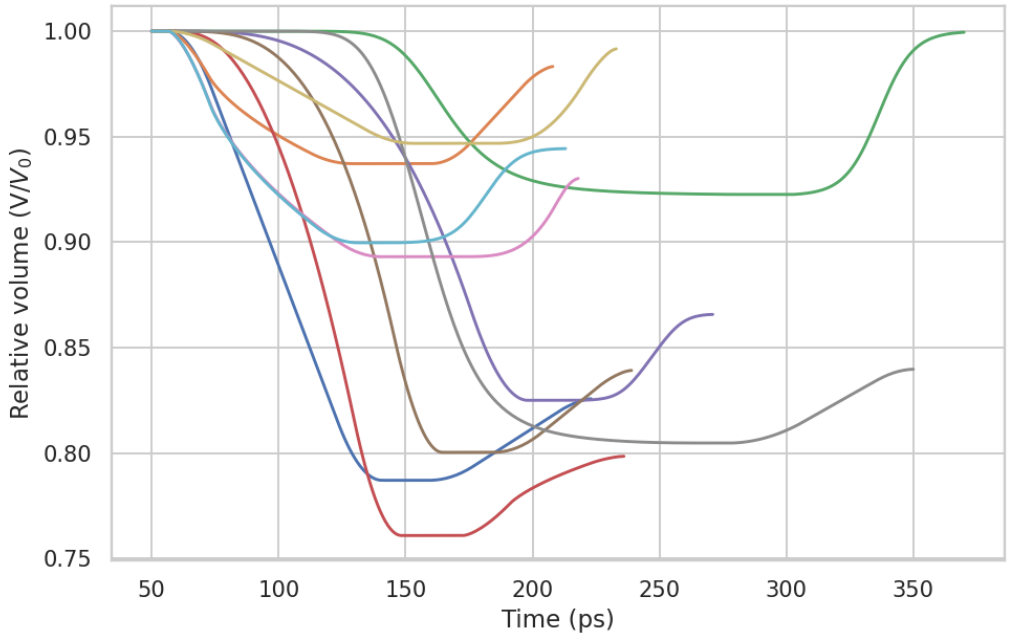

Figure 4.2: A sample set of the 466 compression pathways which constituted the MD training data set. The compressions and expansions follow combinations of linear, quadratic, cubic, radical, or sigmoidal pathways.

or sigmoid - to a particular volume within a particular time period. This set of pathways was not chosen because they represent realistic pathways to compression; they don't. They were chosen to span a parameter space, ensuring that the training data included volumes that corresponded to different strain rates. This served to teach the model that there were multiple pathways to arrive at the same volume, and so neither feature became redundant. As an example, figure 4.2 shows the volume through time of a small set of these compression pathways. In this plot, eight different strain rates correspond to the relative volume of 0.9, including a negative strain rate during expansion, and a strain rate of zero where the sample was held at constant volume. The pathways were also smoothed to remove pathological discontinuities in the strain rate.

These volumes and time periods were selected from specified ranges, which contained the pressure regime and time scales of interest. It was held at a constant volume for a length of time within a specified range to provide the neural network with training data corresponding to zero strain rate. The majority of the simulations

Void radius (Å)	0.6 - 19.9
Compressed relative volume ($\frac{V}{V_0}$)	0.7403 - 0.9499
Compression time (ps)	30 - 294
Hold time (ps)	5 - 55
Expansion time (ps)	35 - 89

Table 4.1: Ranges of parameters which constituted MD data set.

were then expanded by 5% of their compressed volumes over a time period within a specified range, while a minority (42 of the 466 simulations) expanded back to the original volume. This expansion was necessary to contribute examples of negative strain rates to the training data. They were then held at this final volume for 5 picoseconds. The time, volume, temperature in the x-y directions and the simulation's stresses in the x, y and z directions were recorded every picosecond. On average, each simulation required approximately 27 CPU hours. In total, running all the simulations required approximately 12582 CPU hours. However, as previously noted, this data set's density as not been formally optimised; it may be possible to achieve the same performance from a sparser data set and therefore use fewer CPU hours.

The ranges for the void radius, compressed volumes, compression, hold, and expansion times are given in table 4.1 and the distribution of these values is shown in figure 4.3. The parameter space sampled was chosen with an awareness of regime of interest, how the simulation tool would work, and the features that were physically important for the neural network regression model to learn. Note from figure 4.3 that most of these distributions are not uniform. For example, 3 Å was the most common void radius, square root was the most common compression pathway, and the compression and expansion times were mostly on the shorter end of the scale. This was done to ensure sufficient density of data in the regime of interest, which in this case was ramp compression on a 100 ps time scale to several tens of GPa. The higher density of data around a particular but arbitrarily chosen void size of 3 Å was produced to provide confidence that a lack of smoothness for in the void radius variable was not contributing to any nonphysical effects in the simulation tool

performance, and testing during development was mainly done using this void size. Figure 4.3 provides insight on how to design a set of simulations that will result in a data set to feed to a neural network; it is not the training data. The data that used to train the neural network model is the output data of these simulations, and its distribution is shown in figure 4.4. The top two rows - comprising void radius, relative volume, strain rate, the longitudinal and transverse stresses, and the 2d temperature - represent the six 'features', while the last row - comprising the rates of change of longitudinal stress, transverse stress, and temperature - represents the three 'labels'. The post-processing of the data averaged the stress in the x and y direction into a single transverse stress value. The strain rate was computed from the volume and the time. Each row of data was matched with the stress values from the previous time step. For the first row, where there was no previous time step, it was matched to the current stress value. The time derivative of the compressive and transverse stresses were computed using a picosecond time step. The data from all the simulations was then concatenated into a single array. Each column's values was then divided by the maximum value in that column. For the input columns (the stresses, temperature, strain rate and void size) the scaling of all features between 0 and 1 allowed the minimization algorithm to work more efficiently, as mentioned in background section 2.5.2. This was necessary because these inputs were on vastly different scales: for example, the stresses reached values which were roughly 10 orders of magnitude greater than the void sizes. The scaling of the output columns (the rates of change of stresses and of temperature) was intended to prevent the error function weighting each output's error unequally.

4.3 Neural network architecture and training

The neural network received as input the strain rate, the void size, and the previous time step's compressive stress, transverse stress, and temperature (all scaled by the largest value of that input in the training set.) Its output was the time derivative of the compressive and transverse stress, and the time derivative of the temperature.

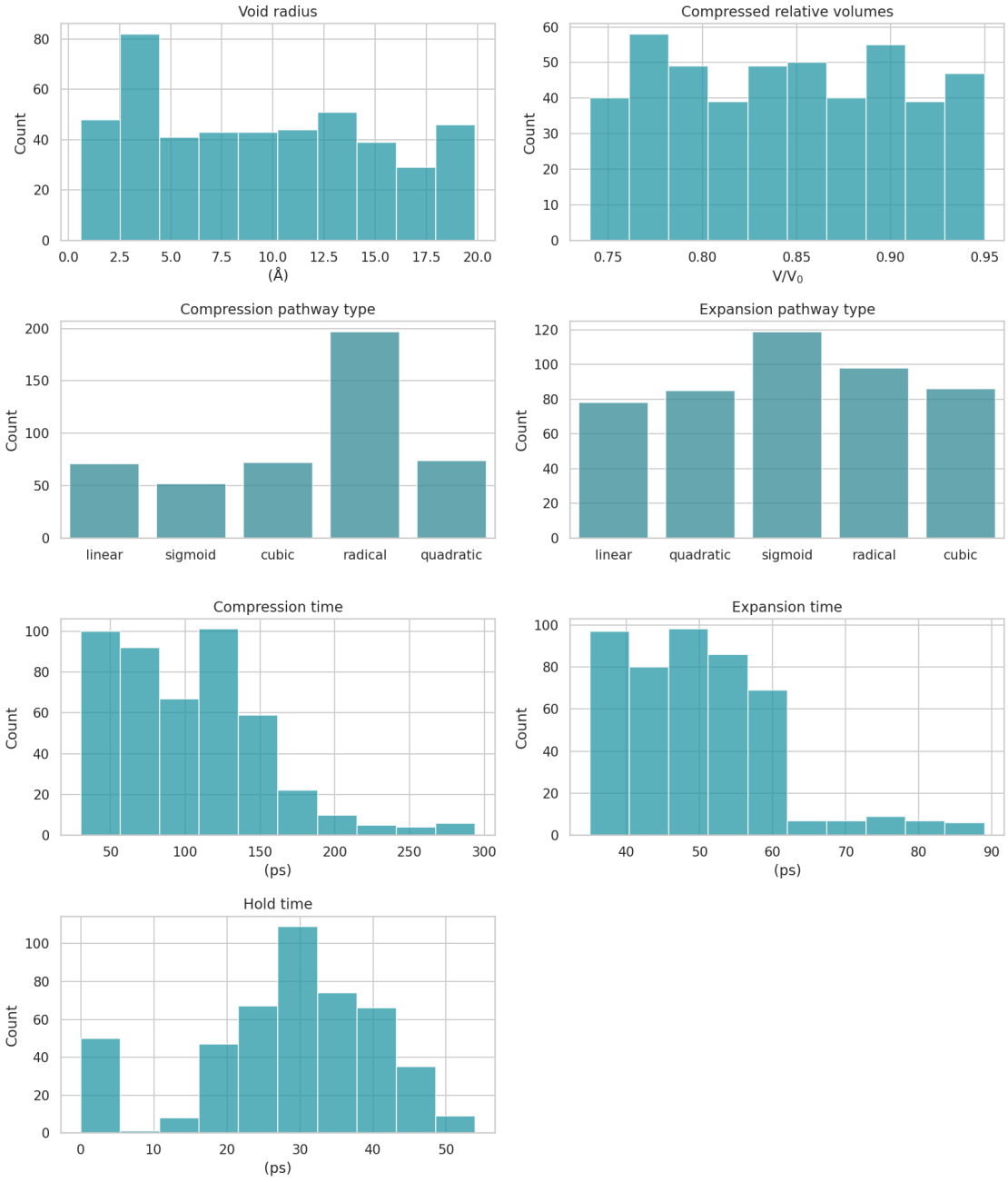

Figure 4.3: Distribution of simulations for all variable parameters.

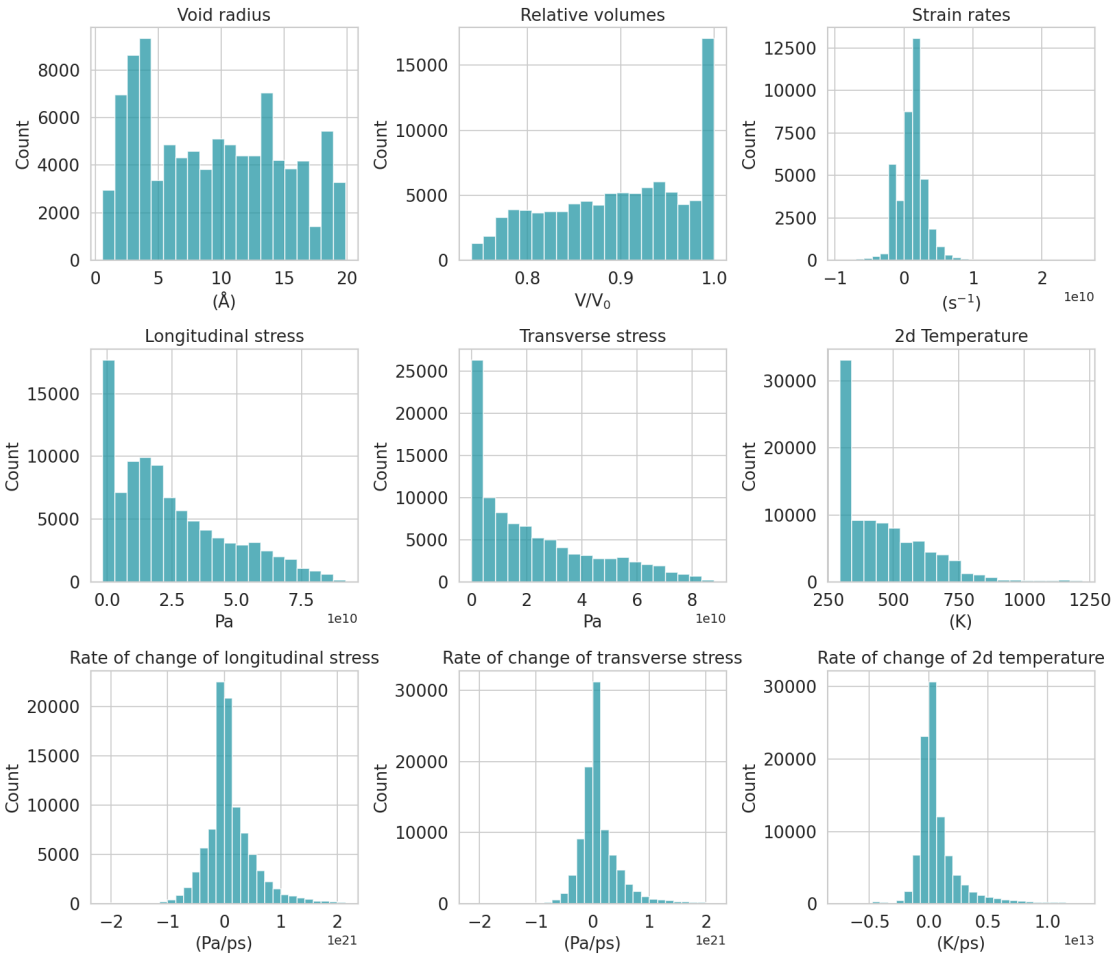

Figure 4.4: Distribution of the output data from the simulations used to train the neural network.

The model was developed using Keras. [67]

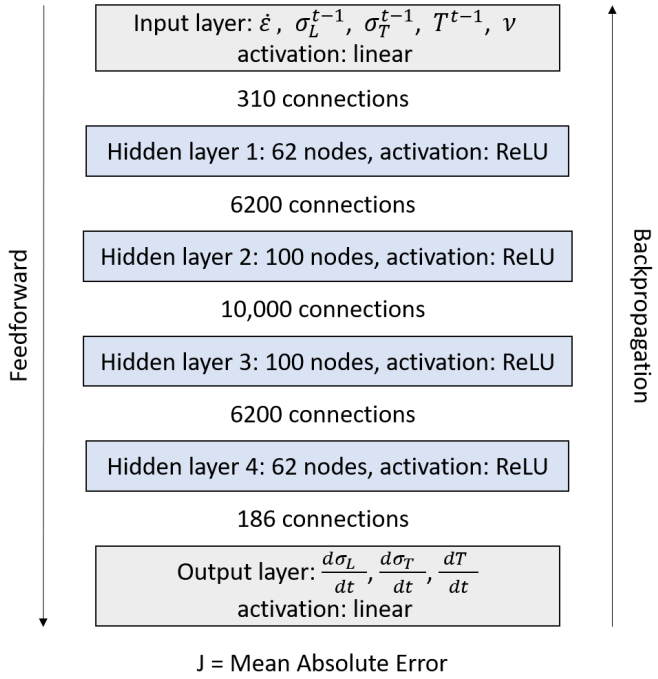

Figure 4.5: Diagram of the neural network model's architecture.

The final network design was a standard feedforward network with 4 hidden layers. The first layer had 62 nodes, the second and third layers had 100 nodes each, and the fourth layer had 62 nodes. This combination of layers and number of nodes in each layer was found to sufficiently balance the needs of a low training runtime, with a CPU time of 33 minutes to train on this work's data set for 500 epochs, and low final error. A diagram of the architecture is shown in figure 4.5. The activation function chosen for all hidden layers was ReLU [112] which as described in section 2.5.3 is commonly used in nonlinear regression problems and was found to work well for our application. The error function was the mean absolute error. The Adam optimizer, a commonly used minimisation algorithm for many purposes, was used with a learning rate of 0.0001. [113] This rate allowed a reasonably fast convergence. It should be emphasised that while these hyperparameters - the architecture, activation function, error function, learning rate, etc. - appear to perform successfully, they are unlikely to be optimised. Hyperparameter optimization is a non-trivial task in neural network design; while

there are several automated procedures to accomplish this optimisation [114] [115] [116] it was judged to be beyond the scope of this thesis. We took a heuristic approach in network design and training, following the principles described in section 2.5.

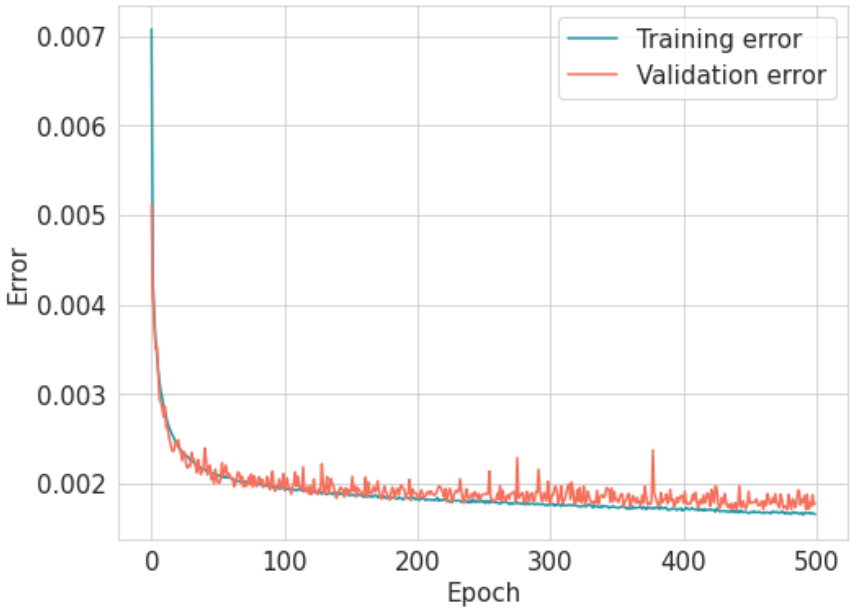

Figure 4.6: The training and validation error of the neural network as a function of epoch.

It was identified during the development process that the neural network, when initialised on different random weights, converged at different minima. While it usually converged to similar values, there were some outliers. For this reason, 50 separate models were built with different randomly initialised weights. For each model, 90% of the original MD data set was selected at random for training, meaning each model was trained on a slightly different set of values. This, in addition to the random initialisation, encouraged the models to converge at slightly different weight values. Of that 90% of the data, 10% was used for validation. The models were then all trained for 500 epochs, which in total took 27.5 CPU hours. The training and validation error for one model is shown as an example in figure 4.6. It can be seen from this figure that both the training and validation error have almost reached convergence. The training error continues to decrease slightly while the validation error ceases to match it; at this point the training ends to prevent overfitting.

This example is representative of the models: the average final training error of all the models was 0.00168 with a standard deviation of 0.000038. The average final validation error of all the models was 0.00175 with a standard deviation of 0.000065. To put these error values in context, recall that each output was scaled by its largest value in that model's training set. Therefore the average final validation error was 0.175 % of those maximum values. In the entire data set, the maximum value for the rate of change of longitudinal stress is 11.17 GPa per picosecond; a maximum error of 0.1815% (the mean plus the standard deviation of the validation error) of this value equals 0.02 GPa per picosecond. For the transverse stress, the equivalent 'maximum average' error is 0.018 GPa per picosecond. For the temperature, it is 0.44 Kelvin per picosecond.

4.4 Performance in simulating elastic-plastic compression of copper

To compare to an NEMD simulation of an identical compression scenario, 50 hydrocode simulations were performed. Each of these used a different neural network regression model. The user input to these simulations included a grid spacing of 2 nm, an initial density of 8.855 gcm^{-3}, an initial sound speed estimate of 5000 ms^{-1} (used only to compute the initial time step size; afterwards, it is computed according to the Courant condition) and a void radius parameter of 6 angstroms.

The mesh was composed of 500 cells, resulting in a total simulation length of 1000 nm. The first test case was a simulation of copper subject to a piston linearly ramped from rest to 1100 m/s in 100 ps. Each hydrocode simulation required approximately 20 minutes to complete using a single CPU; running 50 therefore required 16.7 CPU hours.

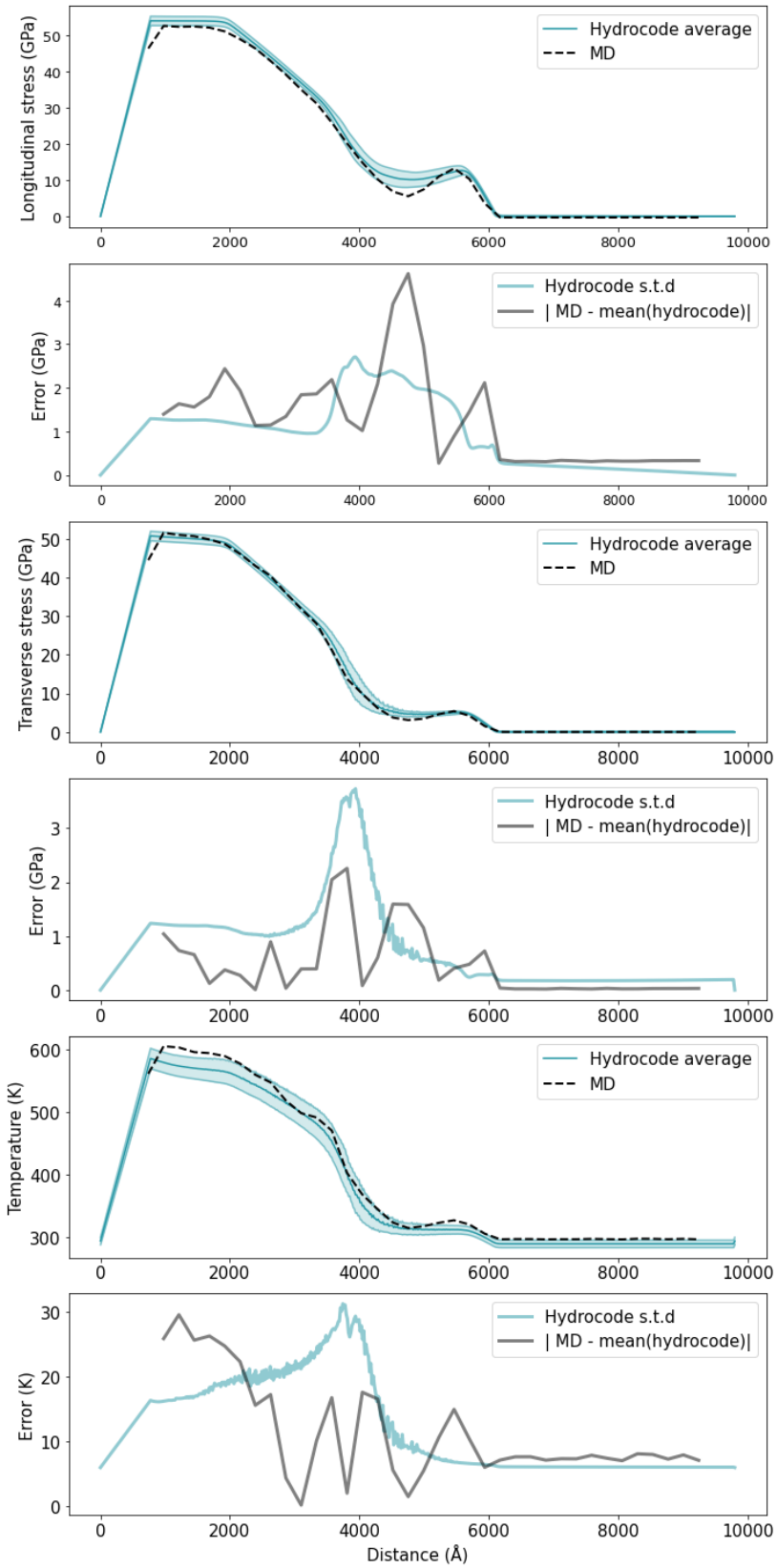

Figure 4.7: Hydrocode results at 120 ps compared to MD for linear ramp compression, showing longitudinal stress, transverse stress, and temperature. Beneath each spatial profile is the corresponding standard deviation of the hydrocode simulations and the absolute difference between the hydrocode average and the MD simulation.

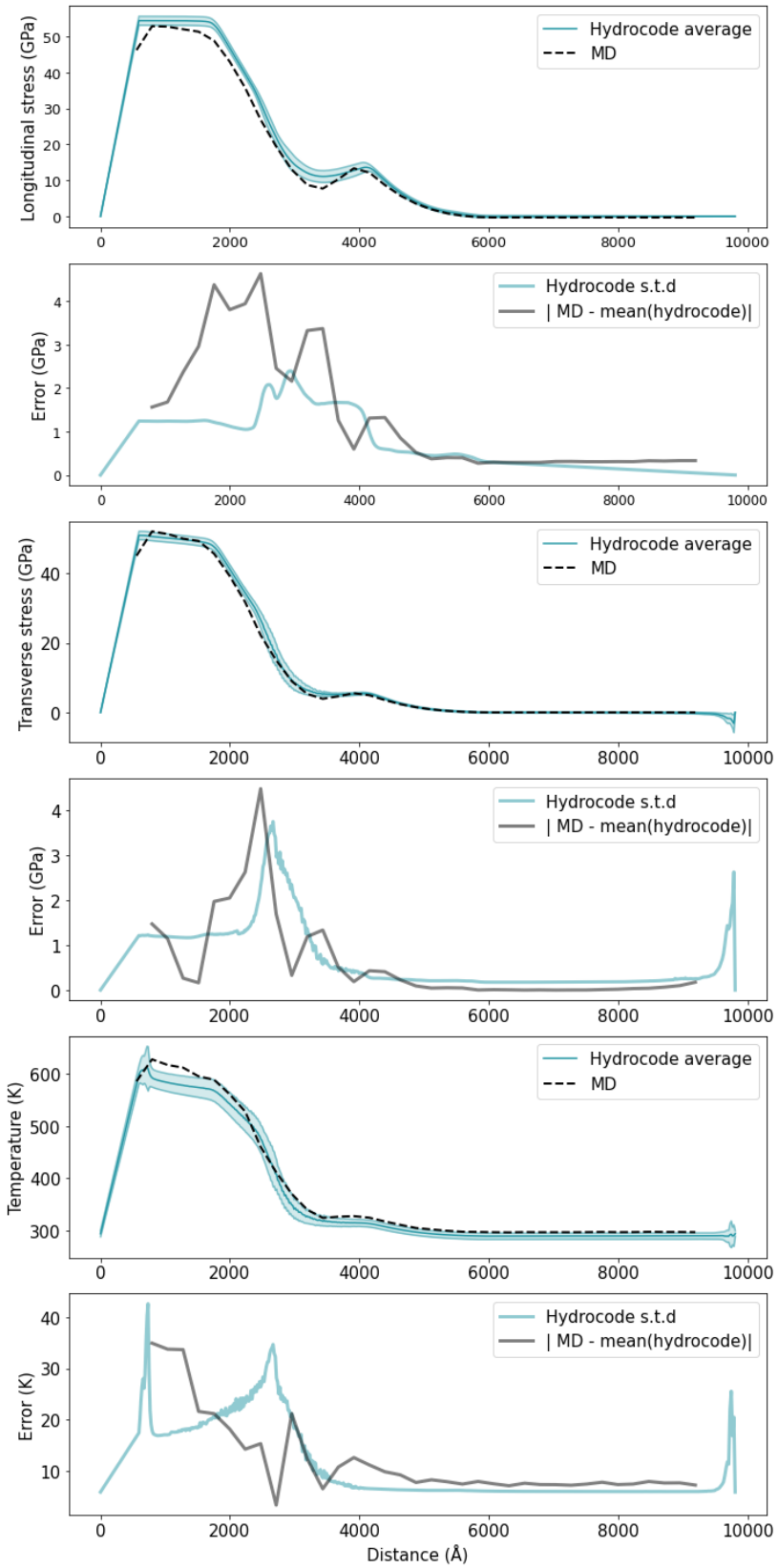

Figure 4.8: Hydrocode results at 120 ps compared to MD for quadratic ramp compression, showing longitudinal stress, transverse stress, and temperature. Beneath each spatial profile is the corresponding standard deviation of the hydrocode simulations and the absolute difference between the hydrocode average and the MD simulation.

The hydrocode results of the compressive stress, transverse stress, and temperature of all 50 hydrocode simulations are shown in figure 4.7, where they are compared to an NEMD simulation of the same scenario. For the details on initialising and running this NEMD simulation, refer back to section 3.3 in chapter 3; it is the same simulation. The hydrocode average reproduces the longitudinal and transverse stress of the MD simulation with relatively high fidelity, with a peak difference of less than 4.5 GPa for the longitudinal stress and less than 2.5 GPa for the transverse stress. The longitudinal stress of the hydrocode is consistently slightly higher than for the transverse stress. A substantial portion of the difference in both cases lies within the standard deviation of the hydrocode simulations. The speed of both the elastic precursor and the growth of the plastic wave are reproduced well for all three variables.

The standard deviation of the temperature is a greater fraction of the average value than for the stresses. If the reader refers back to figure 3.5 in chapter 3, it can be seen that even for identical simulations with only different random seeds in the velocity ensemble, there is a spread of temperatures both at and after the onset of plasticity. It is therefore appropriate that each regression model learns a slightly different governing set of rules within this uncertainty. This equally explains the peak in the standard deviation for the longitudinal and transverse stress, which occurs at the onset of plasticity. This is an appropriate outcome due to the process's inherent stochasticity.

The temperature matches slightly less closely than the stresses, though the peak difference is less than 30 K (less than 15 K when compared to the upper range of the standard deviation.) The hydrocode consistently produces slightly lower temperatures than the MD. Considering the variability of temperature to different compression conditions, and the difficulty in precise temperature measurements in experiments, this is a highly acceptable result.

A quadratic ramp of the same rise time to the same final speed was also tested. The results are shown in figure 4.8. Again, the hydrocodes are broadly effective

in reproducing the spatial profiles of the longitudinal and transverse stresses and the temperature. In this case, the longitudinal stress is slightly overestimated for a greater portion of the profile, with a peak difference of more than 4 GPa. The transverse stress has a similar peak difference, though this is more localised - over the whole profile, it has comparatively high fidelity. The temperature also matches the MD profile more closely, with a peak difference of approximately 35 K. There is also a sign of instability in the temperature calculation close to the piston. However, overall these results deonstrate the ability of the hydrocode to respond appropriately to different compression pathways to the same state.

4.5 Sources of error

While both cases exhibit good agreement, it is still important to consider the sources of the differences. The first source is the error introduced by the small-box approximation itself, which though demonstrated in chapter 3 to reasonably reproduce the evolution of a piston-ramped sample, nonetheless does not do so exactly. As noted in chapter 3, the major sources of error behind the approximation likely include its inability to factor in the flow of dislocations from adjacent regions. Additionally, while the onset of plasticity can be influenced by void size, it cannot be precisely predicted as it is a stochastic process.

The second source of error is the neural network; although the validation error of the time derivatives was minimized, its final value was not zero for any model. In other words, even though the neural network's predictions were very close to the true values, they nonetheless did not match exactly. Even in the case where the neural network reached zero validation error, a question remains about the representational capability of our chosen data set. We describe our data set design principles in section 4.2.1 but emphasise that this data set could be expanded or optimised to be more representative of the relevant physics.

The remaining uncertainty in the neural network model calculation - arising both from the training error and any error associated with 'gaps' in the data set - would

therefore be propagated through each time step of the model. At this stage, another source of uncertainty is introduced. In the hydrocode, the stresses and temperature are computed by multiplying their rates of change (values returned by the neural network) by the current time step. In other words, a linear approximation is made to assign the new value. As long as the time step is small, this should result in little error; nonetheless, the error still exists and can similarly be propagated throughout the simulation. This propagation is a likely source of the slight overestimation of the longitudinal stress and the slight underestimation of the temperature, which in both cases is seen to extend over the whole profile.

The fourth source - of difference, rather than error - is the NEMD simulation which, like the UUC MD simulations, is subject to the stochasticity of plasticity. Therefore there may be a range of possible onsets and final temperatures affected by the particularities of the initialisation and void size and distribution.

4.6 The role of voids as a plasticity parameter

The above results were achieved using a void radius value of 6 angstroms, an adjustable parameter. To choose this value, the same simulation was run with different values for the void radius. The result of using different values for the void radius the hydrocode is shown in the left column of figure 4.9. These are once again each the averages of 50 simulations using 50 different regression models. These are shown side-by-side with results of MD piston simulations with varying void sizes.

The impact of the void radius is as expected: a smaller value leads to the suppression of plasticity and overcompression in the elastic regime. This is accompanied by a greater rise in temperature when plasticity does eventually begin. As the void size becomes greater, this elastic overcompression becomes less pronounced and the plastic wave grows earlier, and results in a lower final temperature. This general behaviour is likewise seen in MD piston simulations, shown on the right. Greater void sizes leads to a convergence in behaviour, echoing the MD results found in chapter 1, figure 3.10.

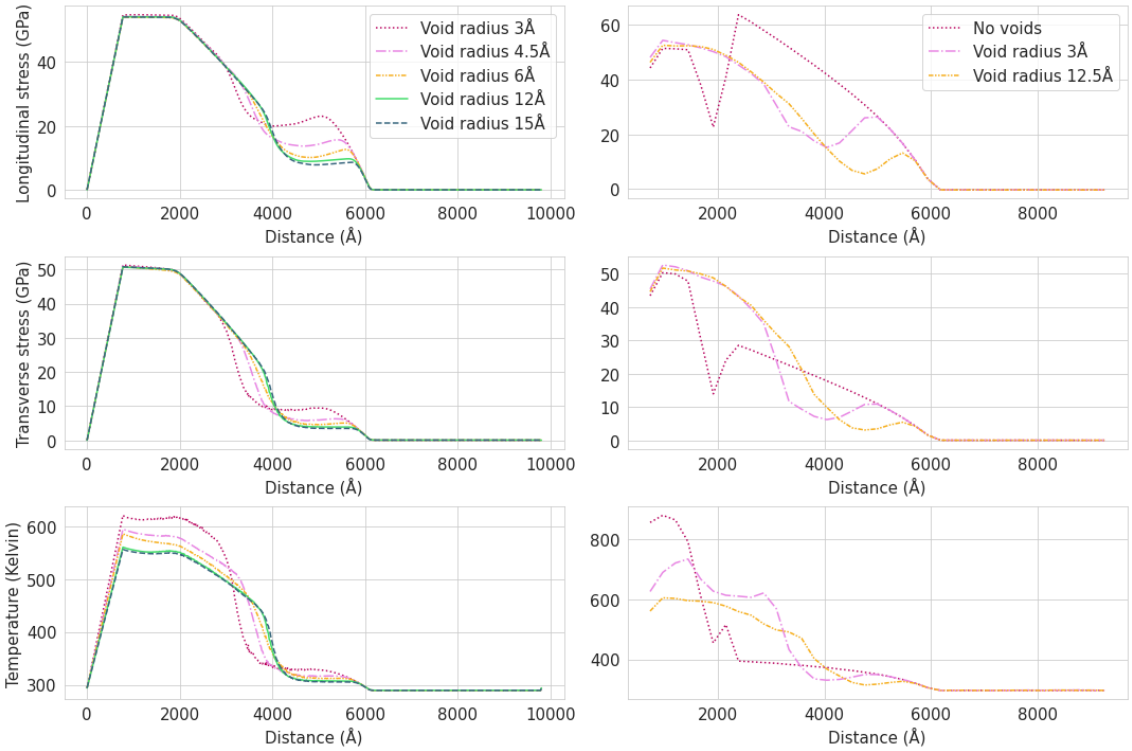

Figure 4.9: On the left: stress and temperature profiles from the hydrocode simulation of the linear ramp compression of copper using different values for the void radius parameter. On the right: the same profiles for an MD piston simulation of linear ramp compression using different void radii, each spaced approximately 313 angstrom apart. All are snapshots at 120 ps into compression.

The void radius parameter therefore appears to be an effective parameter for promoting or inhibiting plasticity. However, it is important to emphasise once more - as we noted in chapter 3 - that the introduction of a single void, of which only the size is varied, is a simplified substitute for the full range of possible dislocation nucleation and dynamics during plastic deformation. However, the benefit of this overall approach is that further initial defect configurations could be added as inputs to the neural network regression model, provided they were adequately supplied by the MD training set. This could be as simple (such as the single spherical void used in this work) or as complex as required. For example, multiple defects whose characteristics and relative positions could be specified. This would certainly increase the complexity of the neural network regression model and must be supported by a robust training set but there is no discernible barrier to the development of this approach.

4.7 Conclusion of chapter 4

A novel methodology for simulating the ramp compression of an elastic-plastic material has been presented. This methodology uses an MD data set of uniform uniaxial compressions to train a neural network regression model, which is then used in a one-dimensional hydrocode to compute the evolution of stress and temperature in response to the rate of strain on the previously computed state. This simulation tool is effective at reproducing the stress and temperature spatial profiles of an MD simulation of a piston-driven ramp compression wave at a fraction of the computational cost: while recognising that there is a large upfront computational cost of data set generation, the 50 hydrocode simulations required 16.7 CPU hours compared to the 1500 CPU hours required by the NEMD piston simulation. Sources of error could likely be further minimised by further development of the MD data set and greater optimisation of the neural network model, though at this stage improvements are likely to be incremental.

While this project focused on ramp compression of copper in the [1 1 1] orientation, this methodology is expected to extend to other orientations and fcc metals - the only requirement is that a researcher constructs a data set that comprehensively covers their region of interest, and ensures sufficient optimisation of their neural network model through the tuning of hyperparameters. Furthermore, while this methodology as been tested for the elastic-plastic transition of copper, it has not yet been tested on phase transformations (either diffusional or diffusionless) which could be an interesting application, particularly with regards to the study of metastable states. To assess whether this methodology is appropriate for their chosen application, a researcher need only consider if their local physics of interest can be adequately reproduced by a small uniform uniaxially compressed sample on their desired time scale.

Chapter 5

Conclusion

5.1 Summary of work presented

This thesis presents a novel methodology for the multi-scale simulation of ramp compression, which uses the following assumption: for sufficiently low strain rates, the local ramp compression of a single Lagrangian element can be approximated in molecular dynamics by uniform, uniaxial compression (UUC). In chapter 3, we present a thorough investigation of this assumption in the case of the elastic-plastic ramp compression of single-crystal copper. In order to have a baseline against which to compare the assumption, this first involved conducting an NEMD simulation of ramp compression of copper using the piston method with a linear velocity profile. It was found that compression of a single-crystal with no initial defects resulted in an extremely high elastic limit of approximately 60 GPa followed by a dramatic pullback in stress. Since a defect-free material is a somewhat unique and unusual case, spherical voids were seeded which served to significantly reduce the elastic limit in line with most experimental results.

From this baseline NEMD simulation the thermodynamic data of two thin Lagrangian elements were recorded. To test the assumption, two separate molecular dynamics simulations were then run, whose dimensions were controlled such that they were compressed along the z-direction to the same extent and at the same rate as the Lagrangian elements of the NEMD simulation. The thermodynamic evolution of these simulations was then compared to that of the Lagrangian elements. Similarly to the NEMD simulation, when the materials were defect-free they

experienced a very high elastic limit followed by a pullback. Therefore they were also seeded with an initial void; different void sizes were tested. This size was found to be a simple yet effective parameter for influencing the onset of plasticity.

It was found that for a specific void size, the UUC simulation reproduced the thermodynamic evolution of the first Lagrangian element (the closest element to the piston) very effectively. The second Lagrangian element was also reproduced effectively, with slightly greater deviation in the temperature and transverse stress. This may indicate that by the point it reaches this second element, the ramp has steepened to the extent that the UUC approximation has become less applicable. Dislocation analysis was conducted to assess the similarity between dominant plasticity mechanisms between the two techniques. It was found that the distribution of dislocation types, their magnitude, and the onset of their nucleation were similar for a certain void size.

To go beyond the results of chapter 3, a multi-scale model was proposed to enable simulation of ramp propagation for an arbitrary piston velocity profile. Chapter 4 presents the development of the three major components of this model: 1) a one-dimensional hydrocode, 2) an MD data set of UUC simulations, and 3) a neural network, trained on the MD data set, to act as the equation of state/constitutive model for the hydrocode.

The one-dimensional hydrocode was based on the formulation by Wilkins, as described in background section 2.3.7. The major change to this formulation was replacing the stress and pressure calculation with a call to the neural network. Therefore the equation of state and von Mises yield condition were no longer needed. The user input to the hydrocode includes the grid spacing, an initial density, an initial sound speed estimate (which needn't be very accurate) and a void radius parameter. The void radius acts as a tuneable plasticity parameter.

The MD data set was constructed to provide the neural network with training data that covered a sufficient range of compression scenarios within the regime of interest. This included compressions along linear, quadratic, cubic, radical and

sigmoid pathways, with different values used for the final volume, the time period, and the initial void radius for each simulation.

The neural network was trained on the relative volume and the scaled stresses, temperature and void radius contained in the MD data set, which was split into a training and validation set. Its output was the time derivative of the longitudinal and transverse stress, and the time derivative of the temperature. While more sophisticated hyperparameter optimisation was not conducted, the network was found to return an acceptably low training and validation error. To account for local minima, 50 independent but identical neural networks were trained on 50 slightly different data sets using different randomly initialised weights.

The trained neural networks were integrated with the hydrocode and used to run 50 independent hydrocode simulations of a sample driven by a piston linearly ramped from rest to 1100 ms^{-1}. The results were then compared to an NEMD simulation of the same scenario. The hydrocode successfully reproduced the elastic-plastic transition, with the same elastic limit as the NEMD simulation when accounting for void radius adjustment. The void radius itself was found to be a highly effective parameter for influencing the onset of plasticity, with the added benefit of being a physically-based mechanism. Reasonably good agreement was found between the profiles of longitudinal stress, the transverse stress, and the temperature, with the hydrocode appearing to predict slightly higher longitudinal stress and slightly lower temperature than the MD. Similar results were found for the case of a quadratically ramped piston.

5.2 Opportunities for future work

There are several opportunities for further exploration and optimisation. As noted in Chapter 3, there is room to explore and fully characterise the relationship between the role of voids in the UUC simulations and the piston-ramped NEMD simulations, to further understand any differences in nucleation and multiplication mechanisms. Alternative initial defect types and configurations could also be explored.

Furthermore, since chapter 3 investigated a specific case - the ramp compression of single-crystal copper in the $\langle 1\,1\,1 \rangle$ orientation - it is clearly necessary to investigate the applicability of this method to different materials. There is no obvious reason why this approximation should not work effectively for other fcc metals with no phase changes. It may also be possible to reproduce a diffusional phase change, as the void technique could play a similar role in the nucleation and growth of a new phase. However, to capture the volume change associated with a phase change, it may be necessary that the UUC be pressure-driven rather than volume-driven. Whether a UUC simulation could reproduce a diffusionless phase change is less certain still, as the role of voids in influencing its onset is even less analogous to the elastic-plastic case. Of course, this is speculation; the reproduction of both diffusional and diffusionless phase changes ought to be investigated.

In chapter 4, we note several areas for improvement and refinement. Firstly, the one-dimensional hydrocode; this is a simple formulation developed by Wilkins in 1963. [117] There may be more effective, efficient or sophisticated formulations now in existence. Provided the coupling to the neural network remained the same, other aspects of the hydrocode could be adapted or improved without disruption to the main technique.

The optimisation of neural networks is a commonly discussed challenge in machine learning. The architecture, the activation function, the error function, the learning rate and the length of training are all known as hyperparameters. While a heuristic approach is often adopted in setting these parameters, as we have done in this thesis, there are several automated optimisation techniques that could be used. [114] [115] [116] An investigation into optimising the neural network using these techniques could be very useful; manual design of the network can be very time-consuming, which reduces the ease of use of the general methodology. An optimised network can return lower-error predictions using a simpler network architecture. Since the feedforward computation is the most computationally expensive step in the hydrocode simulation, a reduced network architecture could provide significant

improvements in runtime.

In the context of this application, the training data set itself could also be considered a type of hyperparameter. As noted in chapter 4, another sparser data set may work equally well to the one presented, which could substantially reduce the computational cost of training. This would certainly be important to consider if multiple data sets for different materials or compression scenarios were developed. It would therefore be reasonable to determine most cost-effective approach to constructing the MD data set.

Finally, the methodology could and should be expanded to include shock wave formation. As noted in the background, ramp waves eventually steepen into shock waves. While the UUC approximation clearly is not appropriate for shocks, we described several existing equilibrium MD techniques in section 2.4.6. Simulations of one of these types could be integrated into the MD data set to enable the neural network to make accurate stress and temperature predictions in response to sudden changes in strain rate.

5.3 Closing remarks

We have presented a novel multi-scale modelling approach to simulating ramp compression for an elastic-plastic material, utilising a neural network emulator. It is hoped that this methodology, or parts of it, may be explored, adopted and further developed to aid the planning or interpretation of experiments as an alternative to current techniques. The challenges, limitations, and opportunities have hopefully been sufficiently documented in this thesis to be of use to any future investigators.

Bibliography

[1] RD Rawlings, JM Alexander, JC Anderson, and KD Leaver. *Materials Science*. Thomas Nelson and Sons Ltd, 1974. 7, 40, 41, 44

[2] Derek Hull and David J Bacon. *Introduction to dislocations*, volume 37. Elsevier, 2011. 7, 42, 43

[3] Brad Lee Holian and Peter S Lomdahl. Plasticity induced by shock waves in nonequilibrium molecular-dynamics simulations. *Science*, 280(5372):2085–2088, 1998. 8, 66

[4] Giles Kimminau, Paul Erhart, Eduardo M Bringa, Bruce Remington, and Justin S Wark. Phonon instabilities in uniaxially compressed fcc metals as seen in molecular dynamics simulations. *Physical Review B*, 81(9):092102, 2010. 8, 84, 85

[5] Damian C Swift, JH Eggert, Damien G Hicks, Sebastien Hamel, Kyle Caspersen, Eric Schwegler, Gilbert W Collins, Nadine Nettelmann, and GJ Ackland. Mass–radius relationships for exoplanets. *The Astrophysical Journal*, 744(1):59, 2011. 13

[6] Marius Millot, Natalia Dubrovinskaia, Ana Černok, Stephan Blaha, Leonid Dubrovinsky, David G Braun, Peter M Celliers, Gilbert W Collins, Jon H Eggert, and Raymond Jeanloz. Shock compression of stishovite and melting of silica at planetary interior conditions. *Science*, 347(6220):418–420, 2015. 13

[7] F Coppari, RF Smith, J Wang, M Millot, D Kim, JR Rygg, S Hamel, JH Eggert, and TS Duffy. Implications of the iron oxide phase transition

on the interiors of rocky exoplanets. *Nature Geoscience*, 14(3):121–126, 2021. 13

[8] Matija Ćuk and Sarah T Stewart. Making the moon from a fast-spinning earth: A giant impact followed by resonant despinning. *science*, 338(6110):1047–1052, 2012. 13

[9] Richard G Kraus, Sarah T Stewart, DC Swift, CA Bolme, RF Smith, S Hamel, BD Hammel, DK Spaulding, DG Hicks, JH Eggert, et al. Shock vaporization of silica and the thermodynamics of planetary impact events. *Journal of Geophysical Research: Planets*, 117(E9), 2012. 13

[10] Richard G Kraus, Seth Root, Raymond W Lemke, Sarah T Stewart, Stein B Jacobsen, and Thomas R Mattsson. Impact vaporization of planetesimal cores in the late stages of planet formation. *Nature Geoscience*, 8(4):269–272, 2015. 13

[11] Sarah Stewart, Erik Davies, Megan Duncan, Simon Lock, Seth Root, Joshua Townsend, Richard Kraus, Razvan Caracas, and Stein Jacobsen. The shock physics of giant impacts: Key requirements for the equations of state. In *AIP conference proceedings*, volume 2272, page 080003. AIP Publishing LLC, 2020. 13

[12] Russell J Hemley and Neil W Ashcroft. The revealing role of pressure in the condensed matter sciences. *Physics Today*, 51(8), 1998. 13

[13] Paul F McMillan. New materials from high-pressure experiments. *Nature materials*, 1(1):19–25, 2002. 13

[14] Klaus Thoma, F Schäfer, Stefan Hiermaier, and E Schneider. An approach to achieve progress in spacecraft shielding. *Advances in Space research*, 34(5):1063–1075, 2004. 13

[15] Zhi-Liang Wang, Yong-Sheng Liu, and R.F. Shen. Stress–strain relationship of steel fiber-reinforced concrete under dynamic compression. *Construction and Building Materials*, 22(5):811–819, 2008. 13

[16] Zhidan Rong, Wei Sun, and Yunsheng Zhang. Dynamic compression behavior of ultra-high performance cement based composites. *International Journal of Impact Engineering*, 37(5):515–520, 2010. 13

[17] David P. Kendall. A Short History of High Pressure Technology From Bridgman to Division 3. *Journal of Pressure Vessel Technology*, 122:91–99, 2017. 13

[18] Leonid Dubrovinsky, Saiana Khandarkhaeva, Timofey Fedotenko, Dominique Laniel, Maxim Bykov, Carlotta Giacobbe, Eleanor Lawrence Bright, Pavel Sedmak, Stella Chariton, Vitali Prakapenka, et al. Materials synthesis at terapascal static pressures. *Nature*, 605(7909):274–278, 2022. 13

[19] Yanming Ma, Mikhail Eremets, Artem R. Oganov, Yu Xie1, Ivan Trojan, Sergey Medvedev, Andriy O. Lyakhov, Mario Valle, and Vitali Prakapenka. Transparent dense sodium. *Nature Letters*, 458:182–185, 2009. 14

[20] Mikhael I. Eremets, Alexander G. Gavriliuk, Ivan A. Trojan, Dymitro A. Dzivenko, and Reinhard Boehler. Single-bonded cubic form of nitrogen. *Nature Materials*, 3:558–563, 2004. 14

[21] Natalia Dubrovinskaia, Leonid Dubrovinsky, Natalia A Solopova, Artem Abakumov, Stuart Turner, Michael Hanfland, Elena Bykova, Maxim Bykov, Clemens Prescher, Vitali B Prakapenka, et al. Terapascal static pressure generation with ultrahigh yield strength nanodiamond. *Science advances*, 2(7):e1600341, 2016. 14

[22] Burkhard Militzer, François Soubiran, Sean M. Wahl, and William Hubbard. Understanding Jupiter's interior. *Journal of Geophysical Research: Planets*, 121:1552–1572, 2016. 14

[23] Earl F O'Bannon III, Zsolt Jenei, Hyunchae Cynn, Magnus J Lipp, and Jason R Jeffries. Contributed review: Culet diameter and the achievable pressure of a diamond anvil cell: Implications for the upper pressure limit of a diamond anvil cell. *Review of Scientific Instruments*, 89(11):111501, 2018. 14

[24] Thomas S. Duffy and Raymond F. Smith. Ultra-High Pressure Dynamic Compression of Geological Materials. *Frontiers in Earth Sciences*, 7, 2019. 14

[25] Thomas J Ahrens. 6. shock wave techniques for geophysics and planetary physics. In *Methods in Experimental Physics*, volume 24, pages 185–235. Elsevier, 1987. 14

[26] LC Chhabildas and LM Barker. Dynamic quasi-isentropic compression techniques: applications to aluminum and tungsten. Technical report, Sandia National Labs., Albuquerque, NM (USA). Thermomechanical and Physical Div., 1986. 14

[27] Jonathan Lind, Brian J Jensen, Matthew Barham, and Mukul Kumar. In situ dynamic compression wave behavior in additively manufactured lattice materials. *Journal of Materials Research*, 34(1):2–19, 2019. 14

[28] RJ Trainor, JW Shaner, JM Auerbach, and NC Holmes. Ultrahigh-pressure laser-driven shock-wave experiments in aluminum. *Physical Review Letters*, 42(17):1154, 1979. 14

[29] Marc A Meyers, F Gregori, BK Kad, MS Schneider, DH Kalantar, BA Remington, G Ravichandran, T Boehly, and JS Wark. Laser-induced shock compression of monocrystalline copper: characterization and analysis. *Acta Materialia*, 51(5):1211–1228, 2003. 14

[30] Damian C Swift and Randall P Johnson. Quasi-isentropic compression by ablative laser loading: Response of materials to dynamic loading on nanosecond time scales. *Physical Review E*, 71(6):066401, 2005. 14

[31] M Keith Matzen, MA Sweeney, RG Adams, JR Asay, JE Bailey, GR Bennett, DE Bliss, DD Bloomquist, TA Brunner, RB e Campbell, et al. Pulsed-power-driven high energy density physics and inertial confinement fusion research. *Physics of Plasmas*, 12(5):055503, 2005. 14

[32] RG Kraus, J-P Davis, CT Seagle, DE Fratanduono, DC Swift, JL Brown, and JH Eggert. Dynamic compression of copper to over 450 gpa: A high-pressure standard. *Physical Review B*, 93(13):134105, 2016. 14, 17, 81

[33] N Ozaki, WJ Nellis, T Mashimo, Muhammad Ramzan, Rajeev Ahuja, Thanayut Kaewmaraya, T Kimura, M Knudson, K Miyanishi, Y Sakawa, et al. Dynamic compression of dense oxide (gd3ga5o12) from 0.4 to 2.6 tpa: Universal hugoniot of fluid metals. *Scientific reports*, 6(1):1–9, 2016. 14

[34] H Jarmakani, JM McNaney, B Kad, D Orlikowski, JH Nguyen, and MA Meyers. Dynamic response of single crystalline copper subjected to quasi-isentropic, gas-gun driven loading. *Materials Science and Engineering: A*, 463(1-2):249–262, 2007. 14

[35] JD Painter, BC Fitzmaurice, Michael Goff, GJ Appleby-Thomas, DC Wood, and Tom Pinto. On a novel graded areal density solution to facilitate ramp wave generation in plate-impact studies. In *AIP Conference Proceedings*, volume 1793, page 060017. AIP Publishing LLC, 2017. 14

[36] M Dargaud and P Forquin. A shockless plate-impact spalling technique, based on wavy-machined flyer-plates, to evaluate the strain-rate sensitivity of ceramic tensile strength. *Journal of Dynamic Behavior of Materials*, 8(1):73–88, 2022. 14

[37] RF Smith, JH Eggert, R Jeanloz, TS Duffy, DG Braun, JR Patterson, RE Rudd, J Biener, AE Lazicki, AV Hamza, et al. Ramp compression of diamond to five terapascals. *Nature*, 511(7509):330–333, 2014. 14

[38] Emma E McBride, A Krygier, A Ehnes, Eric Galtier, M Harmand, Z Konôpková, HJ Lee, H-P Liermann, Bob Nagler, Alexander Pelka, et al. Phase transition lowering in dynamically compressed silicon. *Nature Physics*, 15(1):89–94, 2019. 15

[39] N Amadou, T De Resseguier, E Brambrink, T Vinci, A Benuzzi-Mounaix, G Huser, G Morard, F Guyot, K Miyanishi, N Ozaki, et al. Kinetics of the iron α- phase transition at high-strain rates: Experiment and model. *Physical Review B*, 93(21):214108, 2016. 15

[40] Bruce A Remington, Patrick Allen, Eduaro M Bringa, Jim Hawreliak, Darwin Ho, K Thomas Lorenz, Hector Lorenzana, James M Mcnaney, Marc A Meyers, Stephen W Pollaine, et al. Material dynamics under extreme conditions of pressure and strain rate. *Materials Science and Technology*, 22(4):474–488, 2006. 15

[41] Bruce A Remington, Robert E Rudd, and Justin S Wark. From microjoules to megajoules and kilobars to gigabars: Probing matter at extreme states of deformation. *Physics of Plasmas*, 22(9):090501, 2015. 15

[42] Bob Nagler, Brice Arnold, Gary Bouchard, Richard F Boyce, Richard M Boyce, Alice Callen, Marc Campell, Ruben Curiel, Eric Galtier, Justin Garofoli, et al. The matter in extreme conditions instrument at the linac coherent light source. *Journal of synchrotron radiation*, 22(3):520–525, 2015. 15

[43] Ulf Zastrau, Karen Appel, Carsten Baehtz, Oliver Baehr, Lewis Batchelor, Andreas Berghäuser, Mohammadreza Banjafar, Erik Brambrink, Valerio Cerantola, Thomas E Cowan, et al. The high energy density scientific

instrument at the european xfel. *Journal of synchrotron radiation*, 28(5), 2021. 15

[44] Despina Milathianaki, S Boutet, GJ Williams, A Higginbotham, D Ratner, AE Gleason, M Messerschmidt, Marvin M Seibert, DC Swift, P Hering, et al. Femtosecond visualization of lattice dynamics in shock-compressed matter. *Science*, 342(6155):220–223, 2013. 15, 83, 89

[45] AE Gleason, CA Bolme, HJ Lee, B Nagler, E Galtier, D Milathianaki, J Hawreliak, RG Kraus, JH Eggert, DE Fratanduono, et al. Ultrafast visualization of crystallization and grain growth in shock-compressed sio2. *Nature communications*, 6(1):1–7, 2015. 15

[46] TD Swinburne, MG Glavicic, KM Rahman, NG Jones, J Coakley, DE Eakins, TG White, V Tong, D Milathianaki, GJ Williams, et al. Picosecond dynamics of a shock-driven displacive phase transformation in zr. *Physical Review B*, 93(14):144119, 2016. 15

[47] MG Gorman, AL Coleman, Richard Briggs, RS McWilliams, Andreas Hermann, David McGonegle, CA Bolme, AE Gleason, Eric Galtier, HJ Lee, et al. Recovery of metastable dense bi synthesized by shock compression. *Applied Physics Letters*, 114(12):120601, 2019. 15

[48] Surinder M Sharma, Stefan J Turneaure, JM Winey, and YM Gupta. Transformation of shock-compressed copper to the body-centered-cubic structure at 180 gpa. *Physical Review B*, 102(2):020103, 2020. 16

[49] Jian Sun, Dennis D Klug, and Roman Martoňák. Structural transformations in carbon under extreme pressure: Beyond diamond. *The Journal of chemical physics*, 130(19):194512, 2009. 16

[50] Elliot Snider, Nathan Dasenbrock-Gammon, Raymond McBride, Mathew Debessai, Hiranya Vindana, Kevin Vencatasamy, Keith V Lawler, Ashkan

Salamat, and Ranga P Dias. Room-temperature superconductivity in a carbonaceous sulfur hydride. *Nature*, 586(7829):373–377, 2020. 16

[51] Davide Castelvecchi. Stunning room-temperature-superconductor claim is retracted. *Nature*, 2022. 16

[52] KT Lorenz, MJ Edwards, AF Jankowski, SM Pollaine, RF Smith, and BA Remington. High pressure, quasi-isentropic compression experiments on the omega laser. *High Energy Density Physics*, 2(3-4):113–125, 2006. 16

[53] Nourou Amadou, Erik Brambrink, Thibaut De Rességuier, Adamou Ousmane Manga, Almoustapha Aboubacar, Björn Borm, and Anaïs Molineri. Laser-driven ramp compression to investigate and model dynamic response of iron at high strain rates. *Metals*, 6(12):320, 2016. 16

[54] J.H. Eggert, R.F. Smith, D.C. Swift, R.E. Rudd, D.E. Fratanduono, D.G. Braun, J.A. Hawreliak, J.M. McNaney, and G.W. Collins. Ramp compression of tantalum to 330gpa. *High Pressure Research*, 35(4):339–354, 2015. 17, 81

[55] Sirus K Han, Raymond F Smith, Donghoon Kim, June K Wicks, J Ryan Rygg, Amy Lazicki, Jon H Eggert, and Thomas S Duffy. Polymorphism of gold under laser-based ramp compression to 690 gpa. *Physical Review B*, 103(18):184109, 2021. 17, 81

[56] Jue Wang, Raymond F Smith, Jon H Eggert, Dave G Braun, Thomas R Boehly, J Reed Patterson, Peter M Celliers, Raymond Jeanloz, Gilbert W Collins, and Thomas S Duffy. Ramp compression of iron to 273 gpa. *Journal of Applied Physics*, 114(2):023513, 2013. 17, 81

[57] C. T. Seagle, J.-P. Davis, M. R. Martin, and H. L. Hanshaw. Shock-ramp compression: Ramp compression of shock-melted tin. *Applied Physics Letters*, 102(24):244104, 2013. 17, 81

[58] Robert E Rudd and Jeremy Q Broughton. Concurrent coupling of length scales in solid state systems. *Computer simulation of materials at atomic level*, pages 251–291, 2000. 17

[59] Nathan R Barton, Joel V Bernier, Robert Becker, Athanasios Arsenlis, Robert Cavallo, Jaime Marian, Moono Rhee, H-S Park, Bruce A Remington, and RT Olson. A multiscale strength model for extreme loading conditions. *Journal of applied physics*, 109(7):073501, 2011. 17

[60] Ling Wu, Nanda Gopala Kilingar, Ludovic Noels, et al. A recurrent neural network-accelerated multi-scale model for elasto-plastic heterogeneous materials subjected to random cyclic and non-proportional loading paths. *Computer Methods in Applied Mechanics and Engineering*, 369:113234, 2020. 17

[61] Alexander E Mayer, Mikhail V Lekanov, Natalya A Grachyova, and Eugeniy V Fomin. Machine-learning-based model of elastic—plastic deformation of copper for application to shock wave problem. *Metals*, 12(3):402, 2022. 17

[62] Alexander E Mayer, Vasiliy S Krasnikov, and Victor V Pogorelko. Homogeneous nucleation of dislocations in copper: Theory and approximate description based on molecular dynamics and artificial neural networks. *Computational Materials Science*, 206:111266, 2022. 18

[63] A. P. Thompson, H. M. Aktulga, R. Berger, D. S. Bolintineanu, W. M. Brown, P. S. Crozier, P. J. in 't Veld, A. Kohlmeyer, S. G. Moore, T. D. Nguyen, R. Shan, M. J. Stevens, J. Tranchida, C. Trott, and S. J. Plimpton. LAMMPS - a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales. *Comp. Phys. Comm.*, 271:108171, 2022. 18, 62

[64] Alexander Stukowski and Karsten Albe. Extracting dislocations and non-dislocation crystal defects from atomistic simulation data. *Modelling and*

Simulation in Materials Science and Engineering, 18(8):085001, 2010. 18, 68

[65] 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):015012, 2009. 18, 68

[66] Mark L Wilkins. *Computer simulation of dynamic phenomena*. Springer Science & Business Media, 1999. 18, 45, 47, 51, 52, 103

[67] François Chollet et al. Keras. `https://keras.io`, 2015. 18, 114

[68] Tobias Hermann. frugally-deep. `https://github.com/Dobiasd/frugally-deep`, 2022. 18

[69] Cyril Cayron. Continuous atomic displacements and lattice distortion during fcc–bcc martensitic transformation. *Acta Materialia*, 96:189–202, 2015. 26

[70] JS Bowles and CM Wayman. The bain strain, lattice correspondences, and deformations related to martensitic transformations. *Metallurgical Transactions*, 3(5):1113–1121, 1972. 26

[71] Qi An, Sheng-Nian Luo, Li-Bo Han, Lianqing Zheng, and Oliver Tschauner. Melting of cu under hydrostatic and shock wave loading to high pressures. *Journal of Physics: Condensed Matter*, 20(9):095220, 2008. 26

[72] AG Cullis, NG Chew, HC Webber, and David J Smith. Orientation dependence of high speed silicon crystal growth from the melt. *Journal of crystal growth*, 68(2):624–638, 1984. 26

[73] Timothy C Germann, Brad Lee Holian, Peter S Lomdahl, and Ramon Ravelo. Orientation dependence in molecular dynamics simulations of shocked single crystals. *Physical review letters*, 84(23):5351, 2000. 26

[74] William Thornton Read. *Dislocations in crystals*. McGraw-Hill, 1953. 42

[75] Marc A Meyers. *Dynamic behavior of materials*. John wiley & sons, 1994. 45, 46, 49, 50

[76] JD Johnson. The sesame database. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 1994. 51

[77] Kevin W Cassel. *Matrix, Numerical, and Optimization Methods in Science and Engineering*. Cambridge University Press, 2021. 56

[78] John VonNeumann and Robert D Richtmyer. A method for the numerical calculation of hydrodynamic shocks. *Journal of applied physics*, 21(3):232–237, 1950. 57

[79] 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. 62

[80] F Ercolessi, M Parrinello, and E Tosatti. Simulation of gold in the glue model. *Philosophical magazine A*, 58(1):213–226, 1988. 62

[81] Murray S Daw, Stephen M Foiles, and Michael I Baskes. The embedded-atom method: a review of theory and applications. *Materials Science Reports*, 9(7-8):251–310, 1993. 63

[82] SM Foiles, MI Baskes, and Murray S Daw. Embedded-atom-method functions for the fcc metals cu, ag, au, ni, pd, pt, and their alloys. *Physical review B*, 33(12):7983, 1986. 63

[83] Wataru Shinoda, Motoyuki Shiga, and Masuhiro Mikami. Rapid estimation of elastic constants by molecular dynamics simulation under constant stress. *Physical Review B*, 69(13):134103, 2004. 66

[84] Mark E Tuckerman, José Alejandre, Roberto López-Rendón, Andrea L Jochim, and Glenn J Martyna. A liouville-operator derived measure-preserving

integrator for molecular dynamics simulations in the isothermal–isobaric ensemble. *Journal of Physics A: Mathematical and General*, 39(19):5629, 2006. 66

[85] Brad Lee Holian and Galen K. Straub. Molecular Dynamics of Shock Waves in Three-Dimensional Solids: Transition from Nonsteady to Steady Waves in Perfect Crystals and Implications for the Rankine-Hugoniot Conditions. *Physical Review Letters*, 43, 1979. 66

[86] Brad Lee Holian. Molecular dynamics comes of age for shockwave research. *Shock Waves*, 13:489–495, 2004. 66

[87] EM Bringa, K Rosolankova, RE Rudd, BA Remington, JS Wark, M Duchaineau, DH Kalantar, J Hawreliak, and J Belak. Shock deformation of face-centred-cubic metals on subnanosecond timescales. *Nature materials*, 5(10):805–809, 2006. 66

[88] VV Zhakhovskiĭ, SV Zybin, K Nishihara, and SI Anisimov. Shock wave structure in lennard-jones crystal via molecular dynamics. *Physical review letters*, 83(6):1175, 1999. 67

[89] R Ravelo, BL Holian, TC Germann, and PS Lomdahl. Constant-stress hugoniostat method for following the dynamical evolution of shocked matter. *Physical Review B*, 70(1):014103, 2004. 67

[90] J-B Maillet, Michel Mareschal, Laurent Soulard, Ramon Ravelo, Peter S Lomdahl, Timothy C Germann, and Brad Lee Holian. Uniaxial hugoniostat: A method for atomistic simulations of shocked materials. *Physical Review E*, 63(1):016121, 2000. 67

[91] Evan J Reed, Laurence E Fried, and JD Joannopoulos. A method for tractable dynamical studies of single and double shock compression. *Physical review letters*, 90(23):235503, 2003. 67

[92] Alexey V Bolesta, Lianqing Zheng, Donald L Thompson, and Thomas D Sewell. Molecular dynamics simulations of shock waves using the absorbing boundary condition: A case study of methane. *Physical Review B*, 76(22):224108, 2007. 67

[93] J Dana Honeycutt and Hans C Andersen. Molecular dynamics study of melting and freezing of small lennard-jones clusters. *Journal of Physical Chemistry*, 91(19):4950–4963, 1987. 67

[94] Alexander Stukowski. Structure identification methods for atomistic simulations of crystalline materials. *Modelling and Simulation in Materials Science and Engineering*, 20(4):045021, 2012. 68

[95] Christopher M Bishop et al. *Neural networks for pattern recognition*. Oxford university press, 1995. 69, 71, 76

[96] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016. 71, 72

[97] Hrushikesh Mhaskar, Qianli Liao, and Tomaso Poggio. When and why are deep networks better than shallow ones? In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017. 71

[98] Andrew Ng. Adam optimization algorithm, 2017. 74

[99] A Higginbotham, J Hawreliak, EM Bringa, G Kimminau, N Park, E Reed, BA Remington, and JS Wark. Molecular dynamics simulations of ramp-compressed copper. *Physical Review B*, 85(2):024112, 2012. 81, 84

[100] James Coakley, Andrew Higginbotham, David McGonegle, Jan Ilavsky, Thomas D Swinburne, Justin S Wark, Khandaker M Rahman, Vassili A Vorontsov, David Dye, Thomas J Lane, et al. Femtosecond quantification of void evolution during rapid material failure. *Science advances*, 6(51):eabb4434, 2020. 83, 89

[101] EM Bringa, JU Cazamias, P Erhart, J Stölken, N Tanushev, BD Wirth, RE Rudd, and MJ Caturla. Atomistic shock hugoniot simulation of single-crystal copper. *Journal of Applied Physics*, 96(7):3793–3799, 2004. 83

[102] Takahiro Hatano. Dislocation nucleation in shocked fcc solids: Effects of temperature and preexisting voids. *Physical review letters*, 93(8):085501, 2004. 90, 95

[103] LP Davila, P Erhart, EM Bringa, MA Meyers, VA Lubarda, MS Schneider, R Becker, and M Kumar. Atomistic modeling of shock-induced void collapse in copper. *Applied Physics Letters*, 86(16):161902, 2005. 90, 95

[104] Sirirat Traiviratana, Eduardo M Bringa, David J Benson, and Marc A Meyers. Void growth in metals: Atomistic calculations. *Acta Materialia*, 56(15):3874–3886, 2008. 90, 95

[105] Takahiro Hatano and Hideki Matsui. Molecular dynamics investigation of dislocation pinning by a nanovoid in copper. *Physical Review B*, 72(9):094105, 2005. 95

[106] KJ Zhao, CQ Chen, YP Shen, and TJ Lu. Molecular dynamics study on the nano-void growth in face-centered cubic single crystal copper. *Computational Materials Science*, 46(3):749–754, 2009. 95

[107] Yizhe Tang, Eduardo M Bringa, and Marc A Meyers. Ductile tensile failure in metals through initiation and growth of nanosized voids. *Acta materialia*, 60(12):4856–4865, 2012. 95

[108] Qingshan Dong, Zhongwen Yao, Peyman Saidi, and Mark R Daymond. Effect of pre-existing dislocations on the formation of dislocation loops: Pure magnesium under electron irradiation. *Journal of Nuclear Materials*, 511:43–55, 2018. 95

[109] Jianqiao Hu, Xuan Ye, Xiaoming Liu, and Zhen Chen. The effects of initial void and dislocation on the onset of plasticity in copper single crystals. *Journal of Applied Physics*, 126(16):165104, 2019. 95

[110] P Rosakis, AJ Rosakis, G Ravichandran, and J Hodowany. A thermodynamic internal variable model for the partition of plastic work into heat and stored energy in metals. *Journal of the Mechanics and Physics of Solids*, 48(3):581–607, 2000. 98

[111] Alexander E. Mayer, Mikhail V. Lekanov, Natalya A. Grachyova, and Eugeniy V. Fomin. Machine-learning-based model of elasticmdash;plastic deformation of copper for application to shock wave problem. *Metals*, 12(3), 2022. 102

[112] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323. JMLR Workshop and Conference Proceedings, 2011. 114

[113] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 114

[114] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, 24, 2011. 115, 127

[115] Matthias Feurer and Frank Hutter. Hyperparameter optimization. In *Automated machine learning*, pages 3–33. Springer, Cham, 2019. 115, 127

[116] Tong Yu and Hong Zhu. Hyper-parameter optimization: A review of algorithms and applications. *arXiv preprint arXiv:2003.05689*, 2020. 115, 127

[117] Mark L Wilkins. Calculation of elastic-plastic flow. Technical report, California Univ Livermore Radiation Lab, 1963. 127