2018-01-01

Massively Parallel Computational Studies Of Material Response At High Strain Rate Deformation

Jayalath Abeywardhana Mudiyanselage Madawa Abeywardhana
University of Texas at El Paso, abeywardhanajamm@gmail.com

MASSIVELY PARALLEL COMPUTATIONAL STUDIES OF MATERIAL RESPONSE AT HIGH STRAIN RATE DEFORMATION

JAYALATH ABEYWARDHANA MUDIYANSELAGE MADAWA ABEYWARDHANA

Doctoral Program in Computational Science

APPROVED:

Ramon Ravelo, Ph.D., Chair

Jorge Lopez, Ph.D.

Devesh Misra, Ph.D.

Charles H. Ambler, Ph.D.
Dean of the Graduate School

To my parents and family

♡

MASSIVELY PARALLEL COMPUTATIONAL STUDIES OF MATERIAL RESPONSE AT HIGH STRAIN RATE DEFORMATION

by

JAYALATH ABEYWARDHANA MUDIYANSELAGE MADAWA ABEYWARDHANA

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at El Paso

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

Doctoral Program in Computational Science

THE UNIVERSITY OF TEXAS AT EL PASO

August 2018

Table of Contents

Appendix

List of Figures

List of Tables

Acknowledgements

I would like to acknowledge the support from Air Force Office of Scientific Research under AFOSR Award No. FA9550-12-1-0476, Department of Energy under contract DE-AC52-06NA25396 and the Computational Science Program at University of Texas at El Paso.

I am grateful to my supervisor, Dr. Ramon Ravelo for his valuable time, patience and guidance, which directed me through my research. I am also thankful for the help from Dr. Tim Germann, Dr. Edward Kober, Dr. Saryu Fensin for mentoring me and providing computational resources.

Abstract

Large scale molecular dynamics (MD) simulations are now commonly utilized to study materials at extreme conditions: high pressure and/or temperatures and ultra-high strain rates of deformation. A variety of emerging architectures such as general purpose graphics processing units (GP-GPU) and many-integrated core (MIC) architecture as well as new execution models have changed the traditional approach of high-performance computing. ExMatEx, the DoE initiative for enabling exa-scale (10^{18} flops) performance in scientific applications, has developed several proxy applications to facilitate co-design of novel algorithms and hardware by software developers and microchip vendors.

We have used CoMD, a proxy application for classical MD, to investigate load balancing of shock wave simulation problems in various platforms as well as how best to improve performance of embedded-atom method (EAM) force evaluation kernels in GPUs. We also have implemented quasi-isentropic (QI) compression and expansion model in Los Alamos MD code SPaSM. QI uniaxial compression is achieved by incorporating a strain rate function in the position and velocity equations of motion. In this new formalism the change in internal energy is exactly equal to the work done during the compression or expansion.

Large-Scale molecular dynamics (MD) simulations comprising 4 to 34 million atoms were performed to systematically study the poorly understood interplay between initial dislocation density and strain rate on deformation twinning in bcc metals. Using tantalum as a test case, for which a large body of experimental data exist. The atomic interactions were modeled employing an embedded-atom method (EAM) potential of Ta, we examined both compressive and tensile deformation at strain rates in the range of $10^8 - 10^{11} s^{-1}$. At these high-strain rates, twin nucleation thresholds can clearly be measured. Under both expansion and compression, deformation twinning increases with strain rate for strain-rates

$> 10^9 s^{-1}$.

Chapter 1

Introduction

A large number of scientific disciplines have their own motive for studying materials in extreme environments of very high fluxes of energetic particle and photon, temperature, pressure, compression and strain rate. For an example, exploring novel energy technologies and improving capabilities and efficiency of existing energy sources[1], planetary [2, 3] and stellar [4] sciences, defense and shielding technologies are only a few of the beneficiaries of such studies.

A variety of loading conditions has been experimentally employed for understanding dynamic behavior of materials in these extreme conditions. Quasi-hydrostatic compression in diamond-anvil-cell (DAC) experiments, Isentropic ramp compression and shock compression are few of the compression techniques that have been extensively used. Depending on the method data can be obtained along different thermodynamic paths. Typically, Quasi-hydrostatic experiments provide a complete isothermal in a range pressure[5, 6], while a shock[7] or isentropic compression[8] experiment only provides a single single data point.

These experiments have been pivotal for understanding the dynamic response of solids at high pressure, specially the equation of state (EOS) and material strength. Accurate EOS data of metals are extremely important since they are important to establish accurate high-pressure scales as well as to validate electronic structure calculations. Strength data are useful for developing and validating constitutive (strength) models.

These data can reliably gathered only until the material is stable in it's natural phase.

Tantalum: a body centered cubic (bcc), and copper: a face centered cubic (fcc) are two metals that has been extensively studied for material strength and obtaining appropriate constitutive relations.

Tantalum is a refractory metal with high melting temperature and strength. Tantalum is a ideal pressure standard due to it's simple phase diagram and highly stable bcc phase allowing high pressure data to be obtained without interference of phase transformations. Quantum molecular dynamics (QMD) simulations has demonstrated that the bcc structure is the most stable phase at least up to 100GPa [9], Highly stable bcc phase of Ta allows it to be a perfect candidate for studying plasticity under extreme loading conditions without intervention of phase changes. However, tantalum exhibits twinning and intricate dislocation dynamics under high strain loading which complicate the analysis of it's strength[10, 11, 12].

It is well established knowledge that strength of many materials increases with plastic strain rate $\dot{\epsilon}_p$. Before the emerging of laser shock experiments, access to such higher measurements of $\dot{\epsilon}_p$ through experiments were limited to $\sim 10^5 s^{-1}$. Figure 1.1 shows strength data of copper obtained from quasi-static and shock experiments as well as two constitutive models: Zerilli-Armstrong model[13] (dashed black line) and PTW (Preston, Tonks, and Wallace) model [14] (green diamonds). Experimental data were gathered form Follansbee *et al.* [15, 16], Tong *et al.* [17], Tanner *et al.* [18], Meyers *et al.* [19], Schneider *et al.* [20] and Murphy *et al.* [21].

At very high strain rates ($> 10^8 s^{-1}$) thermal activation mechanisms of plastic deformation are thought to be less important than dislocation motion opposed by phonon drag. As seen in figure 1.1 most models agree in the thermally activated regime, where model parameters are calibrated against experimental data. However, at high strain rates, where recently obtained high strain rate ($\dot{\epsilon}_p > 10^7 s^{-1}$) laser shock strain rates inhabit, experimental data significantly deviate from constitutive models.

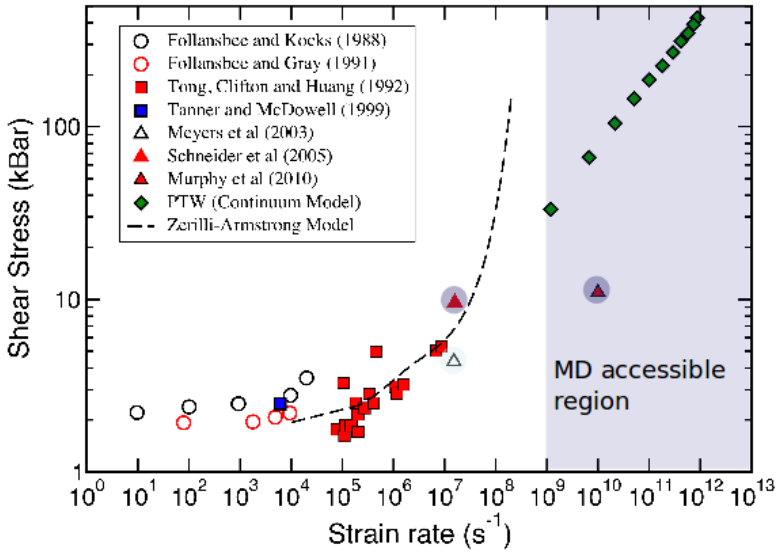


Figure 1.1: Strength of copper as a function of strain rate.

Dynamic response of materials under extreme strain rates have been studied using non-equilibrium molecular dynamics (NEMD) simulations of shock wave propagation[22, 23, 24, 25, 26, 27]. These simulations, however, include the complete process of generation of shock wave, to propagation in the material until it reaches a steady state, and finally emerge at the free surface. The sample dimension in the direction of shock-wave propagation must be greater than the natural thickness of the shock wave, as well long enough to allow the shock wave to reach a steady-wave profile. Sufficient cross-sectional area is also required to study realistic transverse flow. As a result, shock wave simulations require significantly large crystal samples, and longer period of execution time to reach the steady state of the sample even before any useful statistics can be gathered. Thus, these simulations are notoriously computationally expensive.

The strain rate of a shock wave can be given by equation 1.1 where u_p is the particle

velocity and λ is width of the shock[28]. Thus, for an example, to achieve an steady shock profile with an strain rate of $1.0 \times 10^7 s^{-s}$ where the shock velocity is 1.0kms^{-1} the length of the length of the sample has to be in the order of hundreds of micrometers. The number of atoms in a bcc sample with dimension of 40nm $\times$ 40nm $\times$ 100μm will be of the order of tens of billions. Such simulations would consume significant portion of computational resources available to scientific community that uses supercomputing facilities.

$$\dot{\epsilon} = \frac{u_p}{\lambda} \tag{1.1}$$

Therefore, the strain rates accessible through NEMD shock simulations with current resources in high performance computer (HPC) facilities are limited to the area highlighted in the figure 1.1. In this region compression/loading times are of the order of picoseconds and duration of simulation barely reach nanoseconds for large scale MD.

Equilibrium MD methods have been developed for reproducing final state of a shock compressed materials[29, 30]. Dynamic evolution of structural defect and thermodynamic quantities can be probed for longer periods since these methods focus on the final shocked stated rather than the full propagation of the shock wave.

Many of the new computer architectures and execution models which can potentially improve capabilities of conventional MD codes to explore the lower strain rate regime for longer time periods have also been introduced along with new computational techniques. A co-design approach for exploiting the features of these emerging computer architectures to accomplish exascale computing capabilities for studying materials under extreme conditions was initiated by ExMatX.[31].

My research work primarily involved in investigating performance optimizations of characteristic MD kernels on different computer architectures and execution models using proxy

MD code CoMD (Co-Designed Molecular Dynamics) as well as developing new a technique for investigating dynamic response of materials at high strain rate in a massively parallel MD code SPaSM (Scalable Parallel Sort-range MD)[32, 33, 34], and use this new method for studying strain rate dependence of material strength and twinning in tantalum.

The structure and organization of my dissertation is as follows:

In chapter 2, a background of dynamic response and plastic deformation of materials under high strain rates along with a review of experimental techniques and computer simulations that scientist have been using to study high strain rate behavior of materials is given.

Chapter 3 is comprised of a review of classical MD formalism: including MD solvers and their accuracy, stability and efficiency, ensembles and evaluation of atomic interaction. An in-depth review of quasi-isentropic compression, the method we implemented in SPaSM, is also given in this chapter.

Chapter 4 includes a description of parallelization techniques of massively parallel MD code, a review of SPaSM and CoMD. The results gathered by using CoMD as a test environment for load balancing and performance evaluation of MD codes are also given in this chapter.

Chapter 5 a description of methods that we employed for measuring dislocation densities and twin volume fractions in simulated bcc samples is included.

Chapter 6 description of the masively parallel MD simulations that we performed with defective samples up to 34 million atoms in order to understand inter play between dislocations and twins at high strain rates is included.

Finally, the summary and conclusions of this research are presented in Chapter 7.

Chapter 2

Background

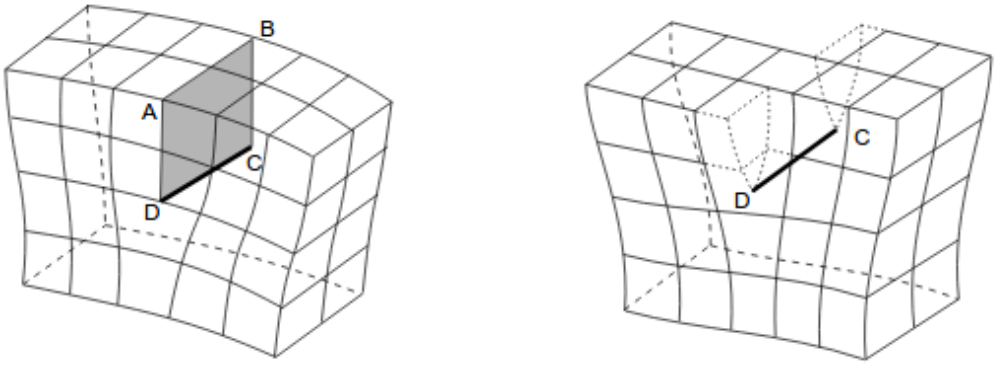
2.1 Dynamic Response of Materials

2.1.1 Yield of Materials

The ability of solids to resist failure or plastic deformation (irreversible change in shape) under the action of external loads is a measure of the strength of the material. The applied loads may be axial (tensile or compressive), or shear. The external forces on the material will cause internal stress that eventually lead material to change its geometry. For small external loads the material deformation is elastic, meaning that, as soon as the external load is removed the material structure returns to its equilibrium initial configuration.

2.1.2 Dislocation Slip

Dislocations in solids are abrupt changes in the regular ordering of atoms, along a line (dislocation line). Two types of dislocations can be observed in ordered lattices: edge and screw dislocations. Edge dislocations can be formed by adding an extra partial plane in between existing full lattice planes. The dislocation line is at the end of the extra plane as seen in figure 2.1(a) for a simple cubic lattice. In an edge dislocation, the Burgers vector is perpendicular to the dislocation line (Burgers vector, is the distance between initial and final decoherence between atomic spacing needed to close a loop around the dislocation line). Screw dislocations are a result of displacing planes relative to each other through shear. Thus, the Burgers vector is parallel to the dislocation line. Figure 2.1 (b) shows left-handed screw dislocation in a simple cubic lattice. In both figure 2.1 (a) and (b),



(a) (b)

Figure 2.1: Edge (a), and screw dislocations in a simple cubic lattice. These images were obtained from [35].

DC represents the dislocation line. Atoms near the dislocation line experience extremely larger displacements from their equilibrium positions. As a result, theory of elasticity or the Hooke's law would not hold for atoms near the dislocation core.

In a perfect lattice, the shear stress required to plastically deform a single crystal by sliding a plane of atoms on top of an adjacent plane was calculated by Frenkel in 1926. Since lattice points in a single crystal are periodical, shearing force required to displace a plane of atom was assumed to be periodical in the lattice spacing of the plane and can be expressed by equation 2.1.

$$\tau = \frac{Gb}{2\pi a}\sin\frac{2\pi x}{b} \tag{2.1}$$

Where, τ is the applied shear stress, G is the shear modulus, b is the lattice spacing in the slip direction, a is the distance between atomic planes and x is the shear translation of the slip plane from its equilibrium. Thus, *theoretical critical shear stress* (τ_{th}) that is required to move a plane of atoms from one equilibrium lattice site to adjacent lattice site is the maximum value of τ and can be given by:

$$\tau_{th} = \frac{Gb}{2\pi a} \tag{2.2}$$

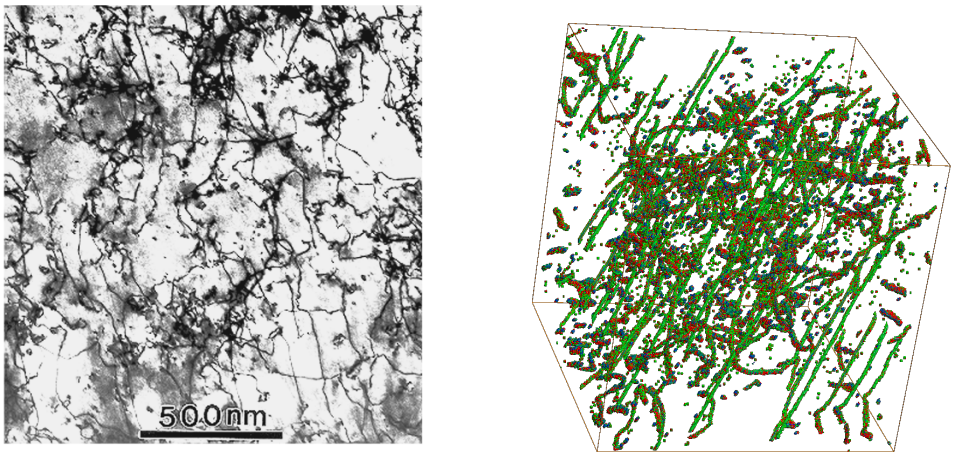
However, calculated theoretical values were orders of magnitude higher than the experimental values. Orowan, Polanyi and Taylor independently described these huge discrepancies by the presence of dislocations in 1934.

In experiments, Transmission Electron Microscopy (TEM) has been most widely used technique for identifying dislocations in deformed samples. Dislocations can be seen as dark lines in bright-field TEM (or bright lines in dark-field TEM) due to bended atomic planes near the dislocation core diffract electron beam instead of transmitting it through the sample. Figure 2.3 (a) shows bright-field TEM image of dislocation structures observed in polycrystalline tantalum sample shocked to 15 GPa [36]. Dislocations in simulated samples can be identified by Central Symmetry Parameter (CSP) [37] or Dislocation Extraction Algorithm (DXA) [38, 39]. Figure 2.3 (b) shows dislocation structures in a MD simulated tantalum sample compressed up to 20 GPa.

Slip Systems

Two types of dislocation movement can occur in solids: dislocation glide - when dislocation moves in the plane which contains both its line and Burgers vector, and dislocation climb - when dislocation moves out of the glide surface, and thus in a normal direction to its Burgers vector.

Propagation of a dislocation can easily occur on closed packed planes (higher planar density) in closed packed directions (higher linear density). Thus, in closed packed crystal structures dislocations move in planes with highest density of atoms. Direction of the movement of the dislocations is typically the direction with highest linear density of atoms. These dislocation glide planes and directions are called slip planes and slip directions respectively, and together they form a slip system in the crystal structure. The most dense



(a) (b)

Figure 2.2: (a): Bright-field TEM image showing dislocation structures in a polycrystalline Ta sample shocked at 15 GPa. (adapted from Hsiung *et al.* [36]). (b): Dislocation structures observed in a MD simulation (using Central Symmetry Parameter) of single crystal tantalum compressed up to 20 GPa.

lattice plane variant in bcc is {110} with two different slip directions (<111>) in each plane. With the six different {110} type planes, bcc structures are capable of hosting twelve {110}<111>slip systems. Further, In bcc structures, closed packed slip direction <111>can lie on {211} and {321} planes introducing two new variants of slip systems each containing twelve and twenty four respectively. However, bcc structure lacks a truly closed packed plane similar to {111} planes in fcc. Thus, bcc slip systems are activated by heat or phonon drag.

Resolved Shear Stress

Resolved shear stress along a slip direction on a particular slip plane can be calculated by equation 2.3. Where, F is the applied shear force on a area of A. ϕ and λ are angles between shear force and slip plane normal, and angle between shear force and slip direction

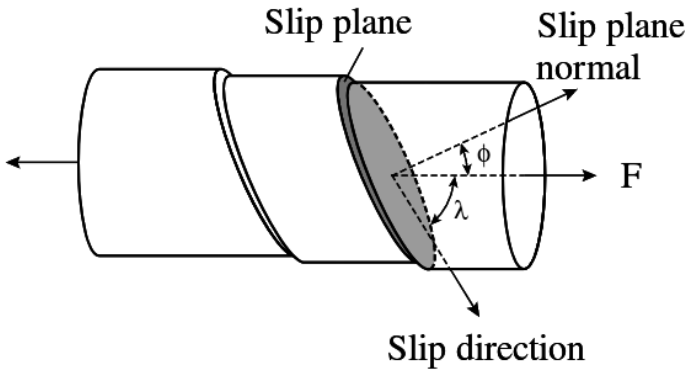


Figure 2.3: Geometry of a slip in a cylindrical crystal. This image was obtained from Hull *et al.* [35].

respectively as shown in figure 2.3. For initiating a dislocation slip on a particular slip system τ_{rss} has to be greater than the *critical resolved shear stress* (CRSS) of the slip system. Geometrical factor $\cos\phi\ \cos\lambda$ is called the Schmid factor.

$$\tau_{rss} = \frac{F}{A}\cos\phi\ \cos\lambda \tag{2.3}$$

Dislocation slip is a relief mechanism of accumulated shear on the lattice structure. The rate at which a material plastically deforms depends on the mobility of the dislocations. Thus, by hindering dislocation motion the material can be strengthened and become brittle.

The density of the defects ρ in the system and the mobility of the dislocations ν can be associated with the plastic strain rate according to Orowan equation:

$$\nu = \frac{\dot{\epsilon}M}{\eta\rho b} \tag{2.4}$$

Where M is a geometrical factor, η is a scaling parameter of order unity and b is the

Burgers vector. According to the equation 2.4, for a fixed mobility, the density of dislocation is linearly proportional to the strain-rate, thus, for faster compression rates, more defects will be generated in the material, causing dislocations to move slower. This can be explained by considering defects as energy barriers for dislocation motion.

Peierls-Nabarro Stress

Peierls in 1940 showed the size of a dislocation and shear stress required to move a dislocation across it's glide plane in a simple cubic lattice can be determined by using suitable approximations and considering the periodic structure of the glide plane. Nabarro *et al.* [40] corrected some of the error of the Peierls model and derived the minimum shear stress required to cause a single dislocation to move continuously through the lattice (equation 2.5).

$$\tau_{PN} = Ge^{-2\pi a/b(1-\nu)} \tag{2.5}$$

Where, G is the shear modulus, ν is the Poisson's ratio, a is interplanar spacing, and b is interatomic spacing.

2.1.3 Twinning

Twinning has been seen in bcc, fcc, as well as in hcp metals. In bcc structures the {112} plane is the most frequent twinning plane and <111> is the most frequent twinning direction reported in samples exposed to a variety of mechanical conditions. In bcc metals, $^{a}/_{2}$ <111>, where a is the lattice constant, screw dislocations has a three-fold symmetry under zero stress. Sleeswyk *et al.* [41] proposed, due to it's three-fold symmetry, $^{a}/_{2}$ <111> screw dislocations can dissociate in to three $^{a}/_{6}$ <111> partials each situated on {112} planes which intersect along the axis of the original screw dislocation. However, when

stress is applied, this three-fold symmetry is unstable and partials can rearrange to form a stable configuration that develops in to a three layer twin in the {112} plane with highest resolved stress. The dislocation reaction related to the nucleation of three layer twin can be given by equation 2.6.

$$ {}^{a}/_{2} < 111 > \rightarrow 3 \times {}^{a}/_{6} < 111 > \tag{2.6} $$

2.2 Plastic Deformation in Metals

Dislocation slip and twinning are competing mechanisms of plastic deformation in single crystal metals. Typically, at low temperature, stacking fault energy (SFE) determines the stress required for dislocation slip and twinning. Higher stacking fault energies increase the stress required for twinning and promote plastic deformation by dislocation slip. Contrarily, lower stacking fault energies increase the tendency of deformation by twinning by reducing the twinning stress. stacking fault energy can be changed by alloying, thus change the twinning stress.

More uniform dislocation arrangements are observed in shock experiments compared to quasi-static compression experiments. Materials with high stacking fault energy often nucleate twins above a threshold stress during shock compression, while they may not nucleate twins under quasi-static compression other than at very low temperature[42]. Dislocation slip is much more sensitive to the strain rate and temperature compared to twinning. However, under very high strain rates where dislocation slip is not sufficient enough to relax the accumulating shear stress, twinning becomes a favorable mechanism.

2.2.1 Plastic Deformation in fcc Metals

In fcc metals, {111} type planes have the highest planar density and <110> directions have the highest linear atomic density. Thus, $^{a}/_{2}$ <110> {111} is the most prominent slip system in fcc structures. Dislocations in fcc structures are classified according to their Burgers vector: perfect dislocations ($^{a}/_{2}$ <110>), Shockley partials ($^{a}/_{2}$ <112>), Frank partials ($^{a}/_{3}$ <111>), Stair-Rod ($^{a}/_{6}$ <110>), and Hirth ($^{a}/_{3}$ <100>) dislocations.

Stacking fault energy is a key factor for determining propensity of twinning in fcc metals. For example, Aluminum and Nickel, two metals with relatively high stacking fault energies, typically deform by dislocation slip at low temperature, while silver, a fcc metal with low stacking fault primarily deform by twinning. Low temperature and high strain rates also promote twinning in fcc metals. Typically, deformation twins in fcc metals are known to be nucleated by the glide of partial dislocations with the same Burgers vector on successive planes[43]. This collectively produces macroscopic strain to accommodate the imposed strain. Stacking faults are another plastic deformation mechanism in fcc crystals and consist of atoms arranged in a hcp-like coordination.

2.2.2 Plastic Deformation in bcc Metals

Majority of lattice dislocations observed in bcc metals have the minimum possible Burgers vectors, $^{a}/_{2}$ <111>, while dislocations with Burgers vectors of types a<100> and a<110> are occasionally formed by reactions. However, only dislocations with $^{a}/_{2}$ <111> Burgers vectors are believed to be contributing to plastic deformation of bcc metals [44].

Bcc transition metals exhibit a complex plastic deformation behavior due to it's inherent lattice properties. Typically, the yield and flow stresses of these materials increase dramatically at lower temperature and highly sensitive to the applied strain rate. Also

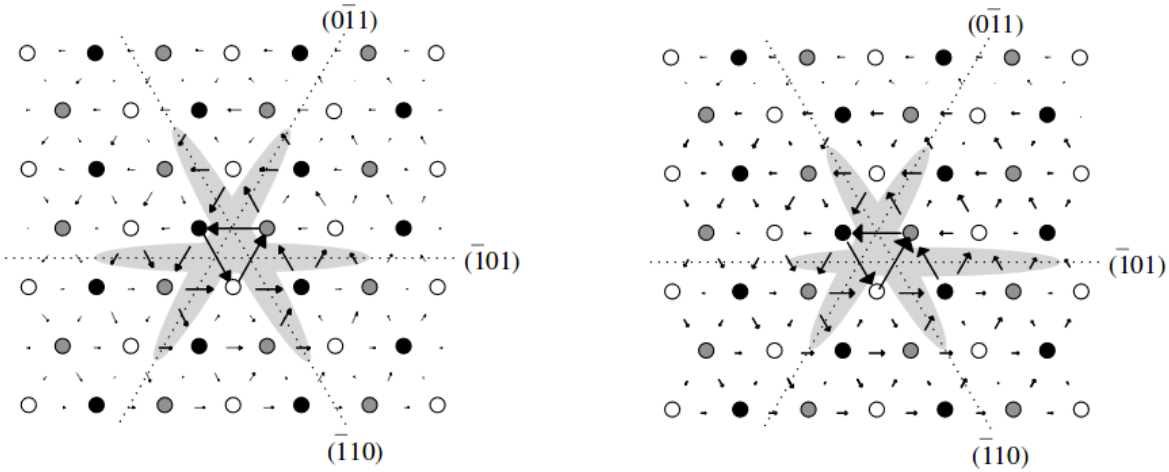
at lower temperature the bcc materials tend to become brittle due to lower dislocation activities caused by higher lattice friction (Peierls stress).

At ambient pressure, low temperature and high strain rates screw dislocations are the dominant structural defects in the plastic deformation of bcc metals due to their intrinsic core properties. Screw dislocations with Burgers vectors $^{a}/_{2} < 111 >$ have non-planar cores, which are capable of splitting in to several non-parallel crystallographic planes. Woodward *et al.* [45] and Duesbery *et al.* [46] showed that isolated screw dislocation cores spread symmetrically onto three conjugate {110} planes using both *Ab-initio* and atomistic simulations respectively.

As a result of these non-planar core structure of screw dislocations, at low temperature, bcc metal does not follow the Schmid law which states that CRSS for glide of a dislocations depend only on the stress components parallel to the slip system. Ito *et al.* [47] and Gröger *et al.* [48] showed that the glide of the screw dislocation in bcc metals is significantly affected not only by the parallel stress components but also by the non-parallel stress components to the slip direction. These non-Schmid stresses can change both lattice friction and the glide plane by altering the dislocation core structure. This behavior is quite different from other closed packed structures which follow the Schmid law.

Figure 2.4 (a) shows structure of a relaxed $^{a}/_{2}$ <111> screw dislocation in bcc molybdenum using differential displacement map (DDM) projected onto (111) plane. The differently colored circles represents three adjacent (111) planes. In DDM, the relative displacements of two neighboring atoms parallel to the Burgers vector are represented by the lengths of the arrows. The three longest arrows at the center of the figure correspond to the relative displacement vector $^{a}/_{6}$[111] which collectively define a Burgers circuit and with a $^{a}/_{2}$ <111> Burgers vector, the total Burgers vector of the dislocation.

Transformed core of the $^{a}/_{2}[111]$ screw dislocation under shear stress applied in the (101) plane is shown in figure 2.4 (b). This transformation was elastic for the applied shear stress $\sigma <$ CRSS, and was plastic meaning that atoms retained in their transformed positions even after the shear stress was removed for $\sigma >$ CRSS. Both figure 2.4 (a) and (b) was obtained from [48]. The three-fold directional spreading is known as the polarization of the dislocation core. Yang *et al.* [49] studied this polarization of $^{a}/_{2}$ <111> screw dislocation in tantalum, molybdenum, and vanadium as a function of applied pressure.



(a) (b)

Figure 2.4: (a) Core of a $^{a}/_{2}$ <111> screw dislocations in molybdenum. (b) Transformed Core of a $^{a}/_{2}$ <111> screw dislocations in molybdenum under applied shear stress. These images were obtained from [48] .

However, edge type dislocations of the bcc lattice, exhibit different core structures. Yamaguchi *et al.* [50] demonstrated that $^{a}/_{2}$ <111> non-screw dislocations have planar cores confined on to single {110} plane in an unstressed bcc structure using atomistic simulations. Vitek *et al.* [51] showed the cores of non-screw dislocations remained confined in to a {110} without changing drastically under an external stress.

2.3 Materials Under Extreme Strain Rates

High strain and strain rates can be imposed on materials by shock compression or high rate ramp (magnetic or plasma piston) drivers. Solid materials undergo rapid and dramatic structure changes when a shock wave is propagating through. These dynamic evolution of materials are difficult to achieve by static compression methods due to the longer time scale allows to dissipate stress by atomic rearrangements and temperature to equilibrate.

2.3.1 Shock Compression

A shock wave is a type of abrupt disturbance propagating through a medium. When a wave moves faster than the speed of sound in the propagating medium, the wave can be considered a shock wave. Similar to an ordinary wave, a shock wave carries energy and momentum. It is characterized by an abrupt, change in pressure, temperature and density in the medium from uncompressed to compressed regimes.

Depending on the characteristic propagation of elastic and plastic waves in a shocked material, three types of shocks have been observed. A weak shock wave uniaxially, elastically and reversibly compresses the material in the direction of the applied stress. Somewhat stronger shocks break into an elastic shock followed by an irreversible plastic shock which nucleate planar defects and dislocations behind the elastic shock wave. At even higher compression, the plastic wave overtakes the elastic shock producing a single shock front.

Typically, shock waves are generated by explosions, or collisions of macroscopic bodies. With careful diagnostics of thermodynamic quantities, shock waves can be used to obtain equation of state (EOS) information in regimes of extreme pressure and temperature which are inaccessible otherwise [3, 52, 53] . To obtain an accurate equation of state, three main conditions have to be fulfilled: a uniform shock front over a large distance, a steady state

pressure in the sample during the measurements and a low level of preheating in the material ahead of the shock wave[2]. EOS data of a material obtained using a single shock is called Hugoniot data and plays a major role in strength and plasticity models.

For one-dimensional steady flow, the conservation of mass, momentum, and energy across a planar shock front connecting the initial (pre-shock) and final (shocked) states leads to the Rankine-Hugoniot relations[30]:

Mass:

$$\rho_0 u_s = \rho(u_s - u_p) \tag{2.7}$$

Where, u_p is the particle velocity and u_s is the shock velocity.

Momentum:

$$P_{zz} = P_0 + \rho_0 u_s u_p \tag{2.8}$$

Energy:

$$E_H = E_0 + \frac{1}{2}(P_{zz} + P_0)(V_0 - V) \tag{2.9}$$

The subscript 0 in these equations stands for the quantities in the pre-shock state. P_{zz} is the normal component of the stress tensor to the direction of the shock propagation.

The total volumetric strain of the shock is given by the equation 2.10 and the strain rate at the shock front by equation 2.11.

$$\epsilon = \frac{u_p}{u_s} \tag{2.10}$$

$$\dot{\epsilon} = \frac{\epsilon u_s}{\lambda} = \frac{\epsilon u_p}{\lambda} \tag{2.11}$$

Where λ is width of the shock.

Shock waves in solid targets are usually generated by using high explosives, impact of a projectile (driven by a compressed gas delivery system or by strong magnetic fields) or by using high energetic lasers. Ultra high pressure shock wave experiments were also attempted in 1965 by Russian Nuclear Center in Arzamas using underground nuclear explosions[54]. Nuclear driven shock compression experiments have achieved ultra high pressures up to 1TPa[52].

Shock compressed samples can be recovered for post-mortem analysis of the plastically deformed microstructures. Johnson *et al.* [55] introduced *in situ* X-ray diffraction analysis for material under shock compression. Figure 2.5 shows schematic of the experimental setup for *in situ* X-ray diffraction analysis.

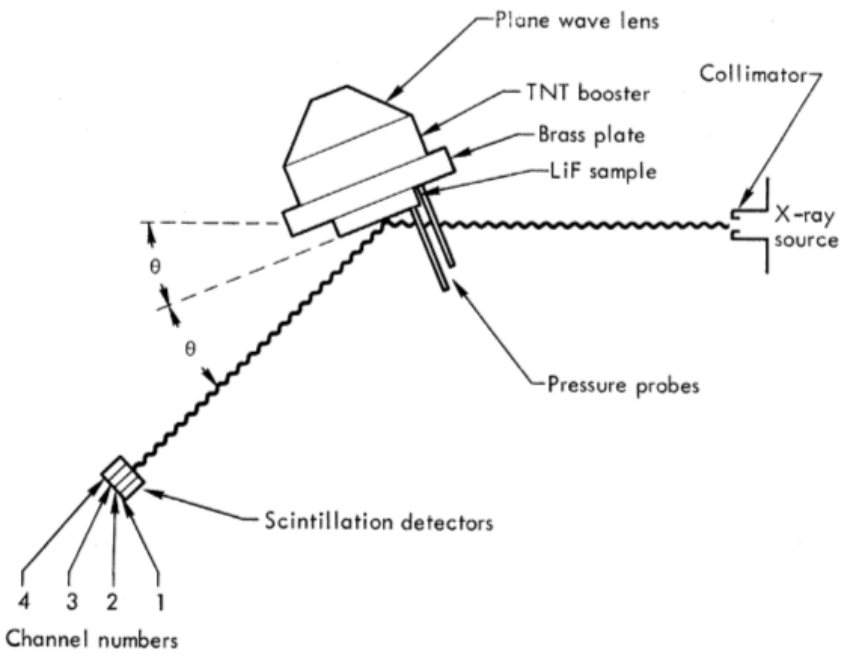


Figure 2.5: Schematic of the experimental setup for *in situ* X-ray diffraction analysis. This figure was adapted from reference [55].

Flyer Plate Shock Experiments

Rice *et al.*[56] used explosives such as TNT (trinitrotoluene) or a mixture of TNT and RDX (Cyclotrimethylenetrinitramine) in contact with the metal target to generate shock waves for his experiments in 1958. This method was known as the 'free surface'method. Figure 2.6 shows a position-time plot of the shock waves and free surface of the target. After the detonation of the explosive, a shock wave propagates through the sample and reach the free surface. This shock wave is represented by line AB in the figure 2.6. Line AC represents a wave reflected from interface of the explosive and the sample. Line AD represents the velocity of this interface which is equal to the particle velocity of the sample. Line BF represents the velocity of the free surface which moves away from the sample with twice the particle velocity, while unloading wave that travels in to sample from the free surface is denoted by line BE. This method can be used for pressures up to few tens of GPa with the free surface approximations which states that the velocity of the unloading wave is approximately twice the particle velocity. However, this method can not be used for porous samples since the velocity of the unloading wave is considerably less than the particle velocity.

In the same year, Al'Tshuler *et al.* [57] used a different set up by adding a extra plate (driver plate) in between explosives and the sample under investigation. The dynamic adiabat and equation of state of the extra plate have to be known to plot it's rarefaction and deceleration adiabats. This method is known as the reflection method or the momentum transfer method. Figure 2.7 shows position-time diagram of this method. Line CD is the shock wave in the driver plate (A). Line DF represents the shock wave in the target sample (B). The reflected wave back in to medium A at the interface of A and B is shown by line DE. Line DG is the trajectory of the interface between A and B. Figure 2.7 was obtained from reference [57].

The time duration of shock pulses generated by explosively-driven flyer plates are typ-

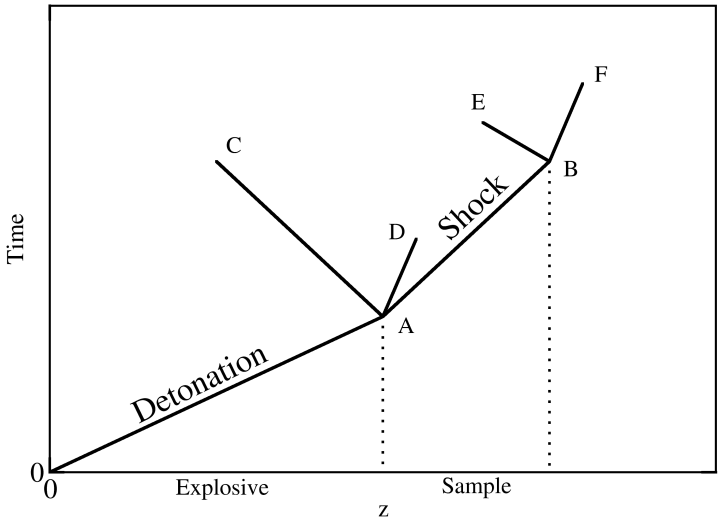


Figure 2.6: Position time diagram of the shock waves and free surface of a target used for free surface method shock experiment.

ically in the range from 1μs to 3μs, while a typical shock pulse in gas gun experiments ranges from 0.1μs to 1μs. Shock velocities that range from 200ms^{-1} to 10kms^{-1} can be achieved by flyer plate impacts.

There are significant amount of shock compression experiments have been conducted on tantalum both using explosively-driven flyer plates[58, 59, 36, 60] and gas gun drivers[61, 53, 62, 63]. These experiments have contributed enormous wealth of data for understanding and modeling dynamic behavior of tantalum pressures up to 100GPa and strain rates up to $1.0 \times 10^6 s^{-1}$. However, for experimental studies beyond these limits high energy laser shocks or magnetic ramp loading techniques are necessary.

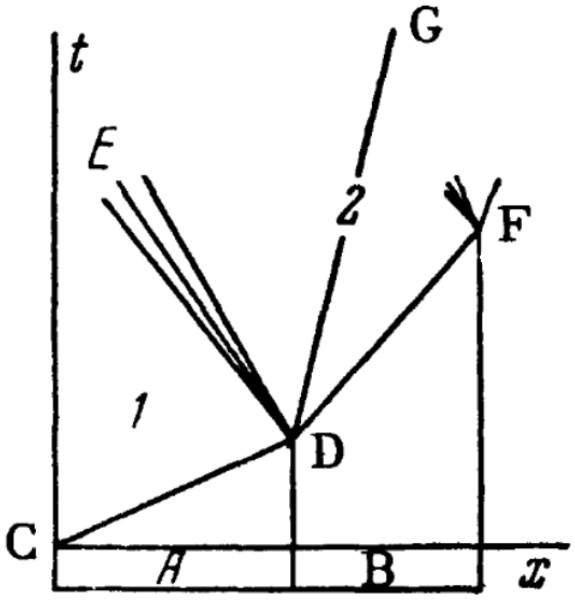


Figure 2.7: Position time diagram of the shock waves and boundary surface of a target used for reflection method shock experiment. This image was obtained from reference [57].

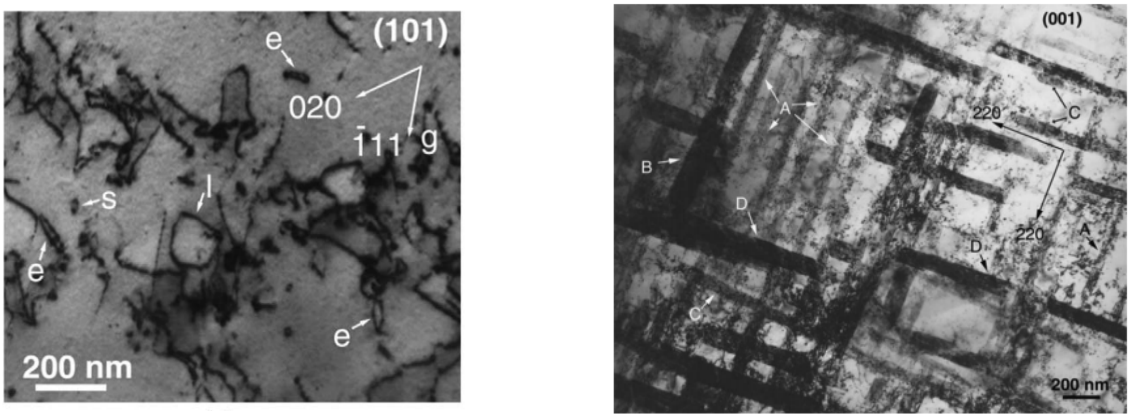
Laser Shock Experiments

Shock waves in solids and liquids, induces by laser pulses that last for few nanoseconds have been employed for studying high pressure phenomena as early as 1960s[64, 65, 66]. Typical strain rate of a laser induced shock wave is in the regime $> 1.0 \times 10^7 s^{-1}$. The laser pulse vaporizes the surface material of the target at extremely high rate generating a pressure wave that propagate in to the material. Anderholm *et al.* [67] demonstrated that increased shock pressure can be achieved by covering the surface of the target with a laser-transparent overlay which enables the confinement of the vapor at the energy deposition.

High energy laser beam cause target surface to heat rapidly forming a plasma. A shock pulse that propagate in to the target is generated by the thermal expansion of this plasma. The heat generated by the laser pulse is limited to first few atomic planes of sample and

rapidly diffuse due to very short period of the laser pulse: typically few nanoseconds.

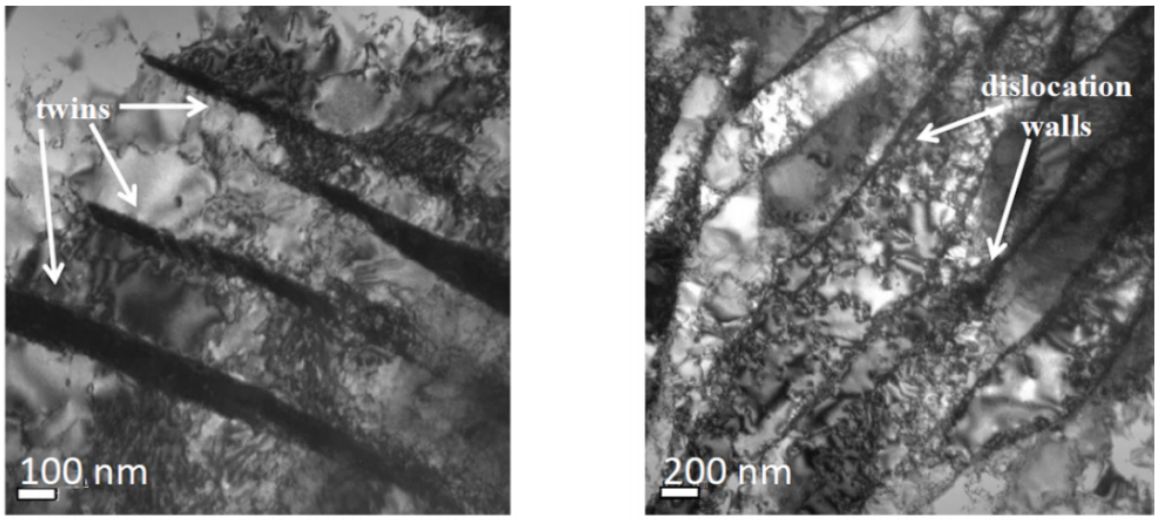
In recent years, Laser shock experiments have been used extensively for studying strength and plastic deformation fcc metals and alloys under high pressure and strain rates[19, 20, 68]. Figure 2.8 (a) and (b) show TEM images of dislocation loops in a 40J laser shocked Cu sample and stacking faults observed in a 200J laser shocked Cu sample observed by Schneider *et al.* [20].



(a) (b)

Figure 2.8: Dislocation loops in a 40J laser shocked Cu sample (a) and stacking faults observed in a 200J laser shocked Cu sample (b). These TEM images were adapted from reference [20].

Followed by successful studies on fcc materials, laser shock experiments were extended to explore strength and plastic deformation of bcc materials at high strain rates. For example, tantalum single crystals[10, 12], nanocrystals [69, 11] and it's alloys were extensively studied. Florando *et al.* [69] demonstrated that deformation twins can be suppressed by higher initial dislocation densities using laser shocks on nanocrystalline tantalum. Figure 2.9 (a) and (b) show twins in tantalum nanocrystalline sample, dislocation walls observed in a cold-rolled tantalum nanocrystalline sample observed in recovered samples by Florando[69].



(a) (b)

Figure 2.9: Twins in tantalum nanocrystalline sample (a) and dislocation walls observed in a cold-rolled tantalum nanocrystalline sample (b) were observed in recovered samples from laser shock experiments by Florando [69].

The National Ignition Facility (NIF) at Lawrence Livermore National Laboratory[70] and the OMEGA Laser Facility at the University of Rochester's Laboratory for Laser Energetics are two of the major high energy laser facilities in the United States that contributed for most of the studies conducted in recent years. The NIF is the world's largest and most powerful laser system for experiments studying high energy density (HED) science. The OMEGA includes 60 laser beams that focus up to 40kJ of energy onto a target that measures less than 1 millimeter in diameter approximately within few nanoseconds[71].

2.3.2 Isentropic (Shockless Ramp) Compression

Shock compression of materials are always accompanied with high temperature where the free energy has a significant thermal component. Isothermal EOS at lower temperatures are often inferred from shock data using thermal pressure models and significantly vary at

higher pressure [72]. Shockless ramp compression has been widely used to counter this issue with shock compression and obtain isothermal EOS data of materials at lower temperatures.

Edwards *et al.* [73] introduced a shockless compression method using lasers that can achieve pressures up to 50GPa and strain rated in the range from $1.0\times10^{6}s^{-1}$ to $1.0\times10^{8}s^{-1}$. In this method, a reservoir is placed in front of the sample of interest. The reservoir and the sample were separated by a vacuum gap. A laser-driven shock wave is used to pressurize the reservoir and turn it in to a weakly ionized plasma which traveled through the vacuum piled up against the sample generating smoothly increasing load on the sample. Figure 2.10 shows a schematic of the shockless ramp compression method at three different stages: t_1 before the shock wave in reservoir reach it's rear surface, t_2 the ionized plasma of the reservoir start stagnating against the target material, t_3 the reservoir continues to pile up and pressure wave in the sample start to steepen. In this method, the peak pressure on the sample can be varied by changing the laser energy and pulse length, gap size, and reservoir density.

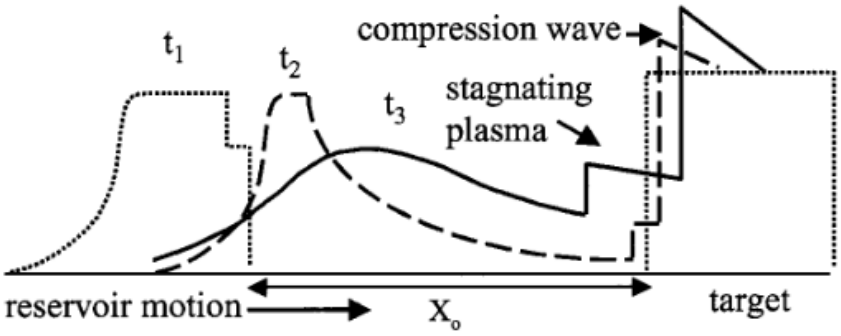


Figure 2.10: Schematic of the shockless ramp compression method at three different stages: t_1 reservoir is shocked with a laser, t_2 ionized plasma of the reservoir starts stagnating against the sample, t_3 compression wave in the sample. This image was obtained from reference [73] .

Laser-driven shockless ramps has been successfully used for studying material strength[74], deformation mechanism[75] and EOS[8] of solids at high pressure and strain rates.

Hall *et al.* [76, 77] demonstrated the ability of Z accelerator of Sandia National Laboratories to produce quasi-isentropic compression of solids. The Z accelerator is a low inductance pulsed power generator which is can store 20 MJ of electrical energy. When the accelerator is discharging, an electrical current pulse of 27 MA with a rise-time of 100ns to 300ns can be created. These current pulses generate time-varying magnetic fields between anode and cathode continuously loading the target material[78]. The Z accelerator has been widely used for study strength[79, 80] and EOS[81] of tantalum.

2.3.3 Computer Simulations

Atomistic simulations of shock wave propagation in solid materials have been extensively used to study material strength[22, 25, 26] and plastic deformation[23, 24, 27, 82] under extreme strain rates. Figure 2.11 shows shock induced twinning in tantalum single crystals at various shock velocities.

A wide variety of different codes have been developed to study materials behavior, focusing on different time and length scales. These codes range from quantum MD simulations to macro-scale continuum codes that employ the finite volume method. Figure 2.12 shows some of the many codes the scientific community is currently using for studying materials at extreme conditions and which structural scales these codes are focused on. Due to change in level of focus, the underlying algorithms vary significantly among these codes. As a result, performance and scalability of these codes drastically change depending on the underlying computer architecture and programming model used in the code.

MD simulation of a steady shock wave is computationally expensive due to large system

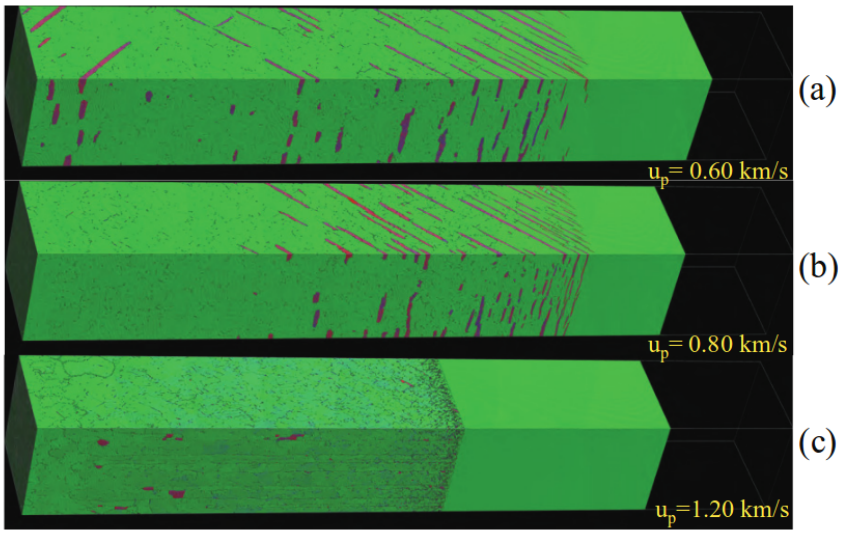


Figure 2.11: Shock induced twinning in tantalum single crystals at various shock velocities. This figure was adapted from reference [22].

sizes and longer time period required to achieve steady state. Execution time of these codes can expand well beyond millions of CPU hours depending on the size of the problem and the probing time scale. Optimal execution of these production level codes is important, as they can be computationally expensive and use much of the computing resources available from a supercomputing facility. Even a small fraction of improvement to the calculation of a timestep in these codes would save a significant amount of execution time, since these codes calculate hundreds of thousand or perhaps millions of timesteps for large scale simulations. And, as a result, saving money and energy available for their studies will benefit researchers. The Exascale Co-Design Center for Materials in Extreme Environments (ExMatEx) has been established by US Department of Energy with the goal of investigating algorithmic (software) and hardware space in order to make exascale performance (10^{18} operations per second) of these production codes a possibility.

For large-scale production level MD simulations, the number of timesteps can easily be in the regime of millions, thus sometimes these simulations needed to be run for weeks

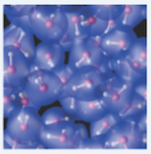
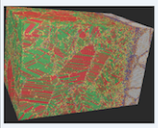
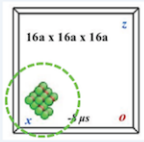
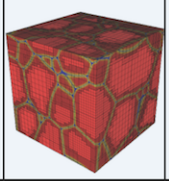
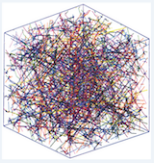
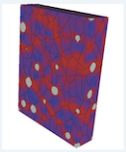
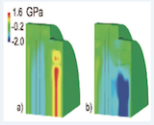
Ab-initio	MD	Long-time	Phase Field	Dislocation	Crystal	Continuum
Inter-atomic forces, EOS	Defects and interfaces, nucleation	Defects and defect structures	Meso-scale multi-phase evolution	Meso-scale strength	Meso-scale material response	Macro-scale material response
						
Code: Qbox/ LATTE Motif: Particles and wavefunctions, plane wave DFT, ScaLAPACK, BLACS, and custom parallel 3D FFTs Prog. Model: MPI + CUBLAS/CUDA	Code: SPaSM/ ddcMD/CoMD Motif: Particles, explicit time integration, neighbor and linked lists, dynamic load balancing, parity error recovery, and *in situ* visualization Prog. Model: MPI + Threads	Code: SEAKMC Motif: Particles and defects, explicit time integration, neighbor and linked lists, and *in situ* visualization Prog. Model: MPI + Threads	Code: AMPE/GL Motif: Regular and adaptive grids, implicit time integration, real-space and spectral methods, complex order parameter Prog. Model: MPI	Code: ParaDis Motif: "segments" Regular mesh, implicit time integration, fast multipole method Prog. Model: MPI	Code: VP-FFT Motif: Regular grids, tensor arithmatic, meshless image processing, implicit time integration, 3D FFTs. Prog. Model: MPI + Threads	Code: ALE3D/ LULESH Motif: Regular and irregular grids, explicit and implicit time integration. Prog. Model: MPI + Threads

Figure 2.12: Different flavors of scientific codes, corresponding proxy applications, and the degree of scale they are focused on. This figure was obtained from[31].

in supercomputing facilities. Given the longer time periods of execution, the possibility of being interrupted before the end of the simulation is higher for production level MD simulations. Thus, saving the status of the simulations in appropriate time intervals is important, so that probability of losing computed results will be minimum.

In-situ diagnostic and post processing of saved sample configurations and gathering statistics is also important for validating the reliability of the simulations and extracting material response to applied constraints. Central symmetry parameter[37] (CSP) and orientation imaging map[83] (OIM) are two of the popular *In-situ* diagnostic methods employed in atomistic simulations due to their low computational cost. These two methods provide insight to dynamic evolution of the atomic structures. Computationally expensive methods, such as common neighbor analysis[84] (CNA), polyhedral template matching [85]

(PTM) and Dislocation extraction algorithm[39, 38] (DXA), are commonly used for post processing of saved samples.

Chapter 3

Classical Molecular Dynamics

Solving equations of motion (Newtonian equations) for a set of atoms or molecules to determine future states of an atomic system is coined as Classical Molecular Dynamics. With the development of the digital computers, the evaluation of Newtonian equations was found to be easily handled by the digital computers. Thus, the era of MD Simulations was initiated. The first ever MD simulation for a system of hard spheres was performed by Alder and Wainwright in 1957 for several hundreds of particles [86]. After several years, A. Rahman successfully simulated a system of 864 liquid Argon atoms on a CDC 3600 digital computer[87] by using a Lennard-Jones potential for modeling atomic interactions.

With the initial attempts being successful, scientists studied systems with multiple elements. As a result, MD simulations were applied to a range of different molecules from basic small diatomic molecules (for example CO, N_2, and H_2O) to large molecules such as proteins[88]. The results obtained from these simulations were compared to result of experiments which led to the validation of the underlying models of the MD simulations and finally the validated models were performed and considered as a reference for theoretical studies of the molecular systems.

Computer programs can be implemented to solve Newtonian equations of motion for particles and simulate classical phenomenon of materials. Generally, in MD simulations ordinary differential equations (ODE) are solved to predict trajectory of the system.

3.1 MD Solvers

Equation 3.2 is the differential form of the Newton's second law and the same as equation 3.1. The general approach for solving ODEs is using finite difference methods. The idea behind these methods is to calculate quantities of interest (in this case, particle positions, velocities, forces, and other dynamical information) at a later time $(t + \delta t)$ given the exact values of those quantities at current time step (t). Thus, equations are solved on a step-by-step basis. Selection of δt will depend on the method of solution, and on the vibrational frequency of the particles in the system. For accurate calculations of MD, must be significantly smaller than the time scale of vibrational period of the atoms.

$$f = ma = -\nabla(u) \tag{3.1}$$

$$\dot{v} = \frac{f}{m} \tag{3.2}$$

$$\dot{r} = v \tag{3.3}$$

The general scheme of MD simulations can be specified step-wise:

- Predict the position, velocities, at time , using the values of these quantities and forces at time .
- Redistribute particles into the correct domains.
- Evaluate forces from the new positions.
- Calculate new velocities from the forces.
- Calculate any variables of interest, such as the energy, pressure, order parameters, and, perform visualizations and saving system configurations if necessary.

There are several solvers that are used for solving ODEs related to MD. As a rule of thumb, these solvers or integrators should possess the following qualities.

- Should be fast, and consume little memory.
- Should allow longer time steps.
- Should duplicate the classical trajectory of particles.
- Should satisfy the known conservation laws for energy, momentum, and enthalpy if necessary.
- Should be time-reversible.
- Should be simple and easy to program.

3.1.1 Verlet Integration

Verlet integration [89] is a numerical method used to solve differential form of Newton's equations of motion. This method is a direct solution for the second order equations such as equation 2.1 and most widely used for classical MD integrators.
The Taylor expansions of $r(t+\delta t)$ and $r(t-\delta t)$:

$$r(t+\delta t) = r(t) + \delta t\nu(t) + (1/2)\delta t^2 a(t) + ... \tag{3.4}$$

$$r(t-\delta t) = r(t) - \delta t\nu(t) + (1/2)\delta t^2 a(t) - ... \tag{3.5}$$

By summing equation 3.4 and 3.5 together, velocity term can be eliminated,

$$r(t+\delta t) = 2r(t) - r(t-\delta t) + \delta t^2 a(t) + O(\delta t^4) \tag{3.6}$$

The velocities can be obtained using the central difference theorem and can be given as in equation 3.7. As a result, to calculate velocity at time t , the position of the particle

at time $t + \delta t$ need to be available. Velocities are not necessary to calculate the trajectory, but they are useful for estimating kinetic energy and temperature of the system.

$$\nu(t) = \frac{r(t + \delta t) - r(t - \delta t)}{2\delta t} \tag{3.7}$$

It is important to notice that the local error of the equation 3.6 is of the order of δt^4 while the local error of equation 3.7 is of the order of δt^2. Further, the important property of the Verlet algorithm is that it is properly centered (i.e.$r(t + \delta t)$ and $r(t - \delta t)$ play symmetric role around the evaluation point in time for the velocities $\nu(t)$), which make it time-reversible.

3.1.2 Leap-frog Method

Modifications to the Verlet method have been introduced later to improve numerical imprecision of instantaneous velocity calculation. This new schemes is called the leap-frog method, and the particle velocity is evaluated at half-integer time steps. The equations for calculating positions and velocities are given in equation 3.8 and 3.9 respectively.

$$r(t + \delta t) = r(t) + \delta t \times \nu\left(t + \frac{1}{2}\delta t\right) \tag{3.8}$$

$$\nu\left(t + \frac{1}{2}\delta t\right) = \nu\left(t - \frac{1}{2}\delta t\right) + \delta t \times a(t) \tag{3.9}$$

Figure 3.1 shows a work flow of a leap-fog method. Verlet and leap-frog methods require only a single force evaluation for advancing positions from $r(t)$ to $r(t+\delta t)$. Other numerical methods such as Runge-Kutta, and Nordsieck method [90] are also viable options, however they require multiple force evaluations to be performed per timestep.

3.1.3 Accuracy, Stability, and Efficiency of the Solver

It is obvious that none of the integration algorithms would generate essentially exact solution over the timesteps of the calculation. Due to the discrete progression of numerical

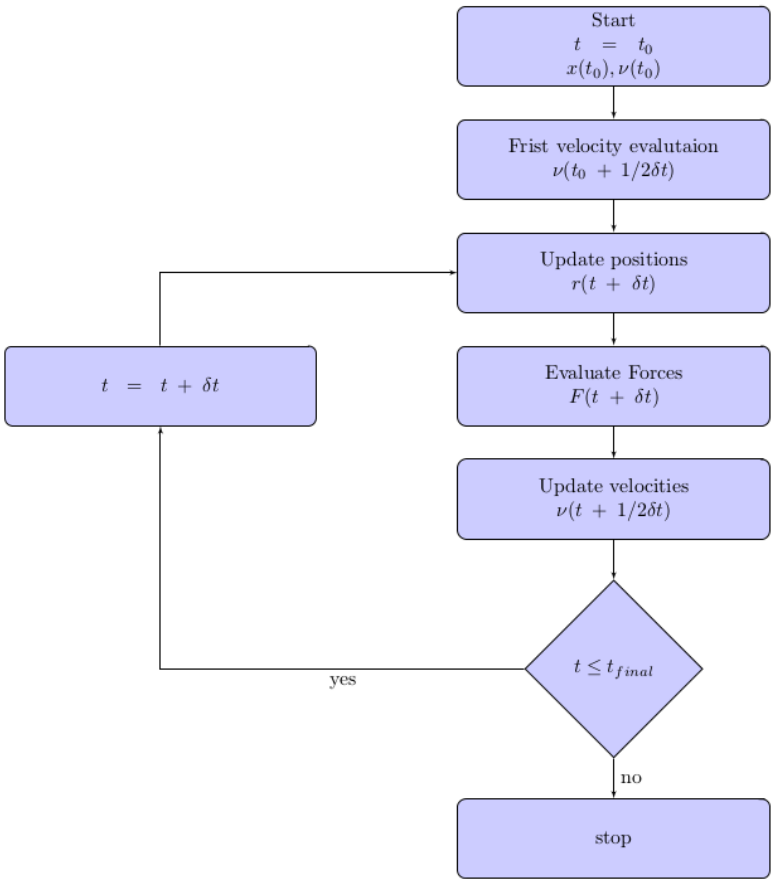


Figure 3.1: Work flow of a leap-frog method.

solvers any small perturbation applied to a classical trajectory will cause it to diverge from its original over the timesteps of the calculation.

Accuracy and stability of numerical schemes depends on how small are the differences between the calculated values of positions and velocities from the values given by the differential equation.

Accuracy of a solver is effected by two sources of errors: round off error arises by representing numbers in a computer using limited number of bits, truncation errors caused by rewriting continuous variables by discrete set of terms. Typically, round off errors are smaller compared to truncation errors given the solver is stable. The truncation error is measured by the order of the numerical solver. The accuracy of the leap-frog method can be evaluated as follows:

By eliminating ν from equation 3.8 and 3.9,

$$\frac{r(t+\delta t) - 2r(t) + r(t-\delta t)}{\delta t^2} = a(t) \tag{3.10}$$

If R is the solution to the differential equations 3.8 and 3.9,

$$\frac{d^2R}{dt^2} = a \tag{3.11}$$

If truncation error at timestep n is ϵ^n, by substituting R in to equation 3.10,

$$\frac{R(t+\delta t) - 2R(t) + R(t-\delta t)}{\delta t^2} = a(t) - \epsilon^n \tag{3.12}$$

Equation 3.12 can be rewritten by expanding $R(t+\delta t)$ and $R(t-\delta t)$,

$$\frac{d^2R}{dt^2} + \frac{\delta t^2}{12}\frac{d^4R}{dt^4} + \text{h.o.t} = a(t) - \epsilon^n \tag{3.13}$$

Where derivative are evaluated at timestep n. By subtracting equation 3.13 from equation 3.11,

$$\epsilon^n = -\frac{\delta t^2}{12}\frac{d^4R}{dt^4} + \text{h.o.t.} \tag{3.14}$$

Therefore, leap-frog method is second order accurate ($\epsilon^n \propto \delta t^2$).

Stability of a solver is concerned with the propagation of errors. An unstable solver can grow very small truncation or round off error in to larger cumulative errors. Stability can be quantitatively measured by obtaining the equation describing the evolution of errors. The characteristic equation of a time integration scheme is obtained by formulating an eigenvalue problem of the amplification matrix $\mathbf{G}$ [91].

The amplification matrix for leap-frog method can be obtained as follows:

Rewriting equation 3.8 and 3.9

$$r^{n+1} - r^n = \nu^{n+1/2}\delta t \tag{3.15}$$

$$\nu^{n+1/2} - \nu^{n-1/2} = \times a^n \delta t \tag{3.16}$$

Where, superscript n represents value of the variable at nth timestep. Error propagation equations are obtained by assuming,

$$r = R + \epsilon_r \tag{3.17}$$

$$\nu = V + \epsilon_\nu \tag{3.18}$$

If ϵ is small enough $a(R^n + \epsilon_x^n)$ can be expanded using the Taylor series. By subtracting the error-free solution from the approximated solution,

$$\epsilon_r^{n+1} - \epsilon_r^n = \epsilon_\nu^{n+1/2}\delta t \tag{3.19}$$

$$\epsilon_\nu^{n+1/2} - \epsilon_\nu^{n-1/2} = \frac{da^n}{dr}\delta t\epsilon_r^n = -\Omega^2\delta t\epsilon_r^n \tag{3.20}$$

Thus error vector $\boldsymbol{\epsilon}$ can be written as,

$$\boldsymbol{\epsilon}^n = \begin{bmatrix} \epsilon_r^n \\ \epsilon_\nu^{n-1/2} \end{bmatrix} \tag{3.21}$$

By adding equation 3.19 and equation 3.20 together and writing them in matrix form,

$$\boldsymbol{\epsilon}^{n+1} - \boldsymbol{\epsilon}^n = \begin{bmatrix} 0 & \delta t \\ 0 & 0 \end{bmatrix} \boldsymbol{\epsilon}^{n+1} + \begin{bmatrix} 0 & 0 \\ -\Omega^2\delta t & 0 \end{bmatrix} \boldsymbol{\epsilon}^n \tag{3.22}$$

By collecting terms,

$$\begin{bmatrix} 1 & -\delta t \\ 0 & 1 \end{bmatrix} \boldsymbol{\epsilon}^{n+1} = \begin{bmatrix} 1 & 0 \\ -\Omega^2\delta t & 1 \end{bmatrix} \boldsymbol{\epsilon}^n \tag{3.23}$$

Multiplying by $\begin{bmatrix} 1 & \delta t \\ 0 & 1 \end{bmatrix}$ which is the inverse of the matrix in left hand side,

$$\boldsymbol{\epsilon}^{n+1} = \begin{bmatrix} 1 - \Omega^2\delta t^2 & \delta t \\ -\Omega^\delta t & 1 \end{bmatrix} \boldsymbol{\epsilon}^n \tag{3.24}$$

$$\boldsymbol{\epsilon}^{n+1} = \mathbf{G}\boldsymbol{\epsilon}^n \tag{3.25}$$

Where $\mathbf{G}$ is the amplification matrix. By applying similarity transformation which diagonalizes $\mathbf{G}$,

$$(S^{-1}\boldsymbol{\epsilon}^{n+1}) = (S^{-1}\mathbf{G}S)(S^{-1}\boldsymbol{\epsilon}^{n}) \tag{3.26}$$

$$\boldsymbol{\epsilon}^{'n+1} = \mathbf{G}^{'}\boldsymbol{\epsilon}^{'n} \tag{3.27}$$

Where $\boldsymbol{\epsilon}^{'} = S^{-1}\boldsymbol{\epsilon}$ and $\mathbf{G}^{'}$ is a diagonalized matrix. Thus, values on the diagonal are the eigenvalues of $\mathbf{G}$.

$$\mathbf{G}^{'} = \begin{bmatrix} \lambda_1 & & & & \\ & \lambda_2 & & & \\ & & \lambda_3 & & \\ & & & . & \\ & & & & . \end{bmatrix} \tag{3.28}$$

The error at timestep $n+1$ can be related to the original error seed $\boldsymbol{\epsilon}^{'\mathbf{1}}$ by applying equation 3.27 repeatedly.

$$\boldsymbol{\epsilon}^{'n+1} = \mathbf{G}^{'}\boldsymbol{\epsilon}^{'n} = (\mathbf{G}^{'})^2\boldsymbol{\epsilon}^{'n-1} = ... = (\mathbf{G}^{'})^n\boldsymbol{\epsilon}^{'1} \tag{3.29}$$

Where,

$$(\mathbf{G}^{'})^n = \begin{bmatrix} \lambda_1^n & & & & \\ & \lambda_2^n & & & \\ & & \lambda_3^n & & \\ & & & . & \\ & & & & . \end{bmatrix} \tag{3.30}$$

Thus, if the magnitude of any of the eigenvalues of $\mathbf{G}$ is greater than one ($|\lambda|_{\max} > 1$) the original error $\epsilon^{'1}$ will be amplified and grow exponentially.

The eigenvalues of $\mathbf{G}$ are given by roots of the equation,

$$\det(\mathbf{G} - \lambda \boldsymbol{I}) = 0 \tag{3.31}$$

Thus, for leap-frog solver,

$$\det(\mathbf{G} - \lambda \boldsymbol{I}) = \begin{vmatrix} 1 - \Omega^2 \delta t^2 - \lambda & \delta t \\ -\Omega^2 \delta t & 1 - \lambda \end{vmatrix} = 0 \tag{3.32}$$

Thus,

$$\lambda^2 - (2 - \Omega^2 \delta t^2)\lambda + 1 = 0 \tag{3.33}$$

The two characteristic solutions λ_+ and λ_- are,

$$\lambda_{\pm} = 1 - \frac{(\Omega \delta t)^2}{2} \pm \left[\frac{(\Omega \delta t)^2}{2}\right] \left[1 - \frac{4}{(\Omega \delta t)^2}\right]^{1/2} \tag{3.34}$$

As a result accuracy and stability of the solver depend on the timestep size, and truncation error of the floating point data type. Stability of the solver requires the timestep size to be much smaller compared to characteristic period of atomic vibrations. The accuracy of the solver also benefits by decreasing timestep size down to a critical timestep size, but lowering timestep size below this critical value can elevate the effect of truncation error causing accuracy of the solver to diminish.

Efficiency of the MD solver is important as number of atoms used in modern day MD simulations can involve up to billions of atoms requiring much larger computational resources than ever before. Also, depending on the studying phenomena probing time span of the MD simulations can be as large as nanoseconds. An efficient solver will advance the system with reasonably less operations, thus execution time spent on updating configuration of atoms in the system is lower. Decreasing the timestep size in order to achieve higher accuracy negatively effect the efficiency of the solver as higher number of timesteps are going to be required to reach the final state of the system. Thus selecting timestep size for a MD simulation without compromising neither accuracy nor efficiency is crucial.

Methods such as Runge-Kutta require force evaluation which is the most expensive set of operations in a MD solver to be performed multiple times. However, accuracy of these methods are generally superior to leap-frog method allowing them to employ a larger timestep size. Thus, lower temperature simulations where atomic vibrational frequency is much smaller can be more efficient by using a higher timestep size without compromising accuracy.

3.2 Ensembles

An ensemble is a collection of all microstates of a system that is with constrains that characterize a system macroscopically. There are three ensembles that are commonly used: microcanonical (NVE) as mentioned above, canonical (constant NVT) where T is a constant temperature in absolute scale, and isothermal (constant NPT) where P is a fixed pressure. For each ensemble, the mentioned extensive variables are kept fixed. In principle, by ensemble averaging, any other thermodynamical quantity can be derived. One can calculate various dynamic as well as structural and energetic properties from the average or the fluctuations of these quantities over the ensemble generated.

3.2.1 Microcanonical Ensemble (NVE)

In micorcanonical ensembles, the total energy of the system and the number of particles in the system are kept constant; each of the members of the ensemble are required to have the same number of particles and total energy. This implies that the system must remain totally isolated; the system is not allowed to exchange energy or particles with its environment in order to stay in statistical equilibrium.

Equations of motion of NVE can be given by equation 3.35 and 3.36 which resembles to equations 3.3 and 3.2 respectively. In equations 3.35 and 3.36 α, represents the direction in the Cartesian coordinate system and i represents the atom index.

$$\dot{r}_{\alpha i} = v_{\alpha i} \tag{3.35}$$

$$\dot{v}_{\alpha i} = \frac{F_{\alpha} i}{m_i} \tag{3.36}$$

3.2.2 Canonical Ensemble (NVT)

In a Canonical ensemble, the total energy is allowed to vary among the members of the ensemble, while the number of particles is fixed. Instead of the energy, the temperature is fixed. The canonical ensemble is appropriate for describing a closed system which is in, or has been in, weak thermal contact with a heat bath. In order to be in statistical equilibrium the system must remain totally closed (unable to exchange particles with its environment), and may come into weak thermal contact with other systems that are described by ensembles with the same temperature. For MD simulations with the NVT ensembles, a modulation of temperature of a system is required and is done via thermostat.

Thus, these ensembles allow the user to obtain desired average temperature of the system. Nose-Hoover thermostat[92] is the most widely used thermostat for NVT ensembles and the one used in SPaSM.

The position update of NVT is the same as in microcanonical ensemble (equation 3.35). However, NVT uses a thermodynamic friction coefficient ζ_T that evolves with time in order to minimize the difference between statistical temperature and the instantaneous kinetics of the system. Thus, equation 3.2 is modified to equation 3.37.

$$\dot{v}_{\alpha i} = \frac{F_{\alpha i}}{m} - \nu_T \zeta_T v_{\alpha i} \tag{3.37}$$

Such that,

$$\dot{\zeta_T} = \nu_T \left(\frac{T}{T_0} - 1 \right) \tag{3.38}$$

Where, ν_T is the heat flow coupling rate (frequency) that provide the time scale for the heat flow. This modification introduces damping fluctuation to the instantaneous temperature T such that it desired temperature T_0 is reached eventually.

3.2.3 Isothermal-isobaric Ensemble (NPT)

In an isothermal-isobaric ensemble, the number of atoms, pressure and the temperature are kept constant. The temperature is kept constant by using a heat flow and the pressure is kept fixed by adjusting volume of the system. The equations of motion of NPT can be expressed as follows:

$$\dot{r}_{\alpha i} = v_{\alpha i} + \nu_P \eta_\alpha r_{\alpha i} \tag{3.39}$$

$$\dot{v}_{\alpha i} = \frac{F_{\alpha i}}{m_i} - (\nu_P \eta_\alpha + \nu_T \zeta_T) v_{\alpha i} \tag{3.40}$$

$$\dot{L}_\alpha = \nu_P \eta_\alpha L_\alpha \tag{3.41}$$

$$\dot{\zeta_T} = \nu_T \left(\frac{T}{T_0} - 1 \right) \tag{3.42}$$

$$\dot{\eta_\alpha} = \frac{\nu_P}{B_0} (\sigma_{\alpha\alpha} - P_{\alpha\alpha}) \tag{3.43}$$

Where, ζ_T is the thermodynamical friction coefficient that regulate temperature with a coupling frequency of ν_T, and η_α is the stain rate in direction of α with a coupling frequency of ν_P. The internal stress tensor $\sigma_{\alpha\beta}$ can be given by equation 3.44.

$$\sigma_{\alpha\beta} = \frac{(\sum m_I (\upsilon_{\alpha i} \upsilon_{\beta i}) + \sum r_{\beta i} F_{\alpha i})}{V} \tag{3.44}$$

3.2.4 Constant Volume Hugoniostat (NVHug)

Maillet *et al.* [29] introduced an alternative equilibrium method called "uniaxial Hugoniostat" for reproducing final state of a shock compressed material. In this method, the sample is instantaneously, uniaxially, and homogeneously compressed to the final volume of the corresponding to the shock pressure at the very beginning of the simulation followed by a relaxation of the internal energy on the shock Hugoniot using a time-reversible integral feedback.

The position update of the NVHug method is same as in microcanonical and canonical ensembles (thus, same as equation 3.35). The velocity update of the NVHug method can be expressed by equation 3.45, where ν_H is the heat flow coupling frequency which provide time scale of internal energy relaxation similar the NVT method and ζ_H is a dimensionless heat flow variable that change with time.

$$\dot{v}_{\alpha i} = \frac{F_{\alpha i}}{m_i} - \nu_H \zeta_H v_{\alpha i} \tag{3.45}$$

$$\dot{\zeta_H} = \frac{\nu_H}{B_0 V_0}(E - E_H(t)) \tag{3.46}$$

Since the internal energy of sample strongly depend on the compressibility of the sample, ζ_H has been normalized by bulk modulus B_0 and volume V_0 at zero pressure. In this method, internal energy of the sample is regulated by adjusting kinetic temperature.

3.2.5 Constant Pressure Hugoniostat ($NP_{zz}Hug$)

Due to uniaxial compression is applied instantaneously in NVHug, the initial temperature and stress tend to be unrealistically large at high strains. Ravelo et al [30] eliminated this shortcoming of the constant volume hugoniostat method by replacing instantaneously compression with a shock-less ramp compression. The modified MD equations of motion for this new hugoniostat method are given in Ravelo et al [30] and [93].

$$\dot{r}_{\alpha i} = v_{\alpha i} + \nu_P \eta_\alpha r_{\alpha i} \tag{3.47}$$

$$\dot{v}_{\alpha i} = \frac{F_{\alpha i}}{m_i} - (\nu_P \eta_\alpha + \nu_H \zeta_H) v_{\alpha i} \tag{3.48}$$

$$\dot{L}_\alpha = \nu_P \eta_\alpha L_\alpha \tag{3.49}$$

$$\dot{\zeta_H} = \frac{\nu_H}{B_0 V_0}(E - E_H(t)) - \beta_H \zeta_H \tag{3.50}$$

$$\dot{\eta_\alpha} = \frac{\nu_P}{B_0}(\sigma_{\alpha\alpha} - P_{\alpha\alpha}) - \beta_P \eta_\alpha \tag{3.51}$$

Where, $\sigma_{\alpha\alpha}$ is internal stress tensor, η_α is the stain rate in direction of α (by convention, uniaxial compression is performed in z-direction, therefore, $\eta_z \neq 0$ and $\eta_x, \eta_y = 0$), ν_H and ν_P are coupling frequency of the heat flow (ergostat given by equation 3.50) and strain-rate (barostat given by equation 3.51). β_H and β_P represents the damping coefficients of ergostat and barostat. In this formalism, $E_H(t)$ is given by equation 3.52.

$$E_H(t) = E_0 + \frac{1}{2}(\sigma_{\alpha\alpha}(t) - P_0)(V_0 - V(t)) \tag{3.52}$$

3.2.6 Quasi-Isentropic Compression (QIC)

Ravelo *et al.* [93] derived QIC method for MD, a different method than the Hugoniostat methods, by introducing a pre-defined strain rate function. In this method, the system is dynamically compressed along an isentrope resulting an increase in internal energy caused only by the work done on the system in contrast to heat flow employed in Hugoniostat methods. The equations of motion of QIC can be expressed as follows:

$$\dot{r}_{\alpha i} = v_{\alpha i} + \dot{\epsilon}_\alpha r_{\alpha i} \tag{3.53}$$

$$\dot{v}_{\alpha i} = \frac{F_{\alpha i}}{m_i} - \dot{\epsilon}_\alpha v_{\alpha i} \tag{3.54}$$

$$\dot{L}_\alpha = \dot{\epsilon}_\alpha L_\alpha \tag{3.55}$$

$$\dot{\epsilon}_\alpha = f(t, \tau) \tag{3.56}$$

Where, $\dot{\epsilon}_\alpha$ is true strain rate in the direction of compression modeled by a predetermined continuous strain rate function $f(t, \tau)$. t is time and τ is compression time. Figure

3.2 shows a set of possible strain rate functions that can be implemented easily.

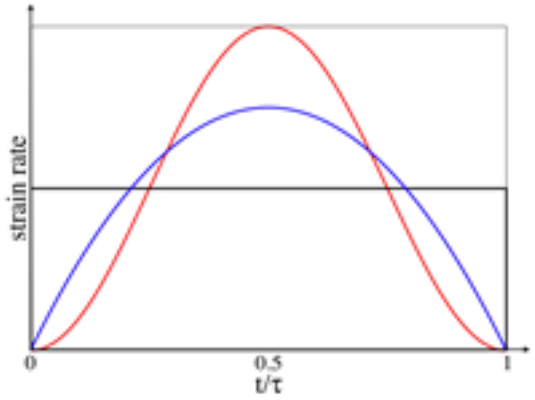


Figure 3.2: Different strain rate functions that can be used to model strain rate function. Black line represents constant strain rate, blue line represents parabolic strain rate function, and, red line represents strain rate function modeled with $sin^2\theta$.

The equations 3.53 to 3.55 can be used to show that the rate of change of internal energy with time is equal to the negative of the work done by the barostat.

Starting from the first law of thermodynamics:

$$\dot{E} = \dot{Q} - \dot{W} = \sum_i m_i \nu_i \dot{\nu}_i + \sum_{i,j} \frac{\partial U}{\partial r} \dot{r} = \sum_i^N m_i \nu_i \dot{\nu}_i - \sum_i^N F_i \dot{r}_i \tag{3.57}$$

By using equation 3.54 for substituting $\dot{\nu}$,

$$\dot{E} = \sum_i^N m_i \nu_i \left(\frac{F_i}{m_i} - \dot{\epsilon} \nu_i \right) - F_i \dot{r}_i \tag{3.58}$$

$$\dot{E} = \sum_i^N \nu_i F_i - m_i \dot{\epsilon} \nu_i^2 - F_i \dot{r}_i \tag{3.59}$$

By using equation 3.53 for substituting ν,

$$\dot{E} = \sum_{i}^{N} [(\dot{r_{\alpha i}} - \dot{\epsilon_\alpha} r_{\alpha i}) F_{\alpha i} - m_i \dot{\epsilon_\alpha} \nu_{\alpha i}^2 - F_{\alpha i} \dot{r_{\alpha i}}] \tag{3.60}$$

$$\dot{E} = -\dot{\epsilon_\alpha} \sum_{i}^{N} [r_{\alpha i} F_{\alpha i} + m_i \nu_{\alpha i}^2] \tag{3.61}$$

The stress tensor is given by,

$$\sigma_{\alpha\beta} = \frac{\sum_{i}^{N} [r_{\beta i} F_{\alpha i} + m_i \nu_{\alpha i} \nu_{\beta i}]}{V} \tag{3.62}$$

By substituting stress tensor in equation 3.61

$$\dot{E} = -\epsilon_\alpha V \sigma_{\alpha\alpha} = -\sigma_{\alpha\alpha} \dot{V} = -\dot{W} \tag{3.63}$$

Equation 3.63 shows that $\dot{Q} = 0 \rightarrow T\dot{S} = 0$. Thus, the entropy of the system is unchanged by the QIC. This property of QIC will allow studying the correlation between temperature differences and the plastic work done by the dislocation motion (relaxation). Figure 3.3 shows $\dot{E}$ and $-\dot{W}$ as a function of time in a QIC simulation where a defective sample is compressed 25% with t_{rise} of 500ps. It is clearly noticeable that entropy of the system is preserved

The change of entropy (dS) of a system is given by equation 3.64, where T is the absolute temperature and dQ is the amount of heat released or absorbed. However, due to the dissipation of heat caused by plastic deformation entropy can be changed in small quantities (for example, small spikes in $\dot{E}$ curve in figure 3.3), which is negligible. Hence, with the QIC (approximately constant entropy), higher strains and strain rates can be achieved at significantly lower temperatures compared to shock wave simulations with the same amount of strain.

$$dS = -\frac{dQ}{T} \tag{3.64}$$

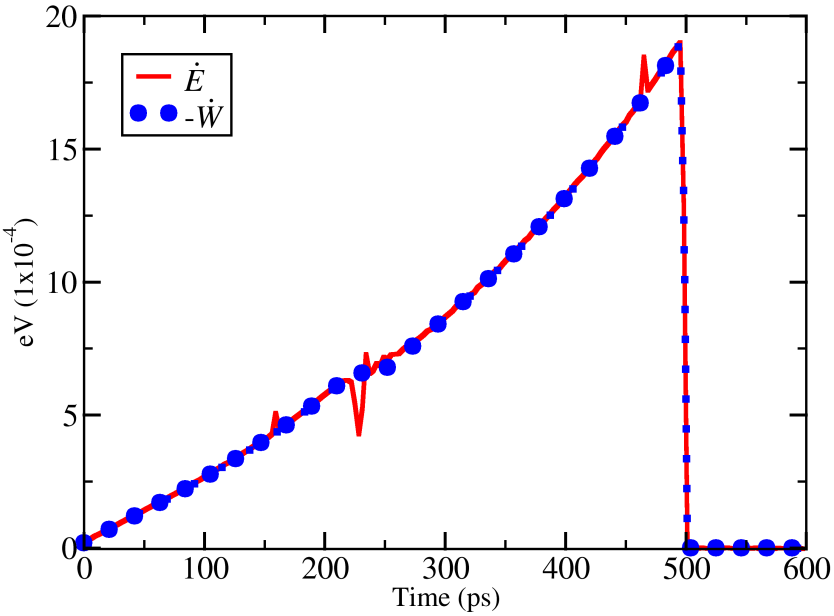


Figure 3.3: $\dot{E}$ and $-\dot{W}$ as a function of time in a QIC simulation where a defective sample is compressed 25% with t_{rise} of 500ps.

Strain rate in above equations are expressed in true strain rate which is defined by equation 3.81. Depending on the strain rate function employed true strain rate need to be calculated every time step through out the compression.

Strain Rate

By definition, strain rate is the rate of change of length with respect to a reference length scale. However, depending on the reference length scale, whether it is initial length (L_0) or the current length ($L(t)$), strain rate is identified as *engineering strain rate* or *true strain rate* respectively. The QIC implementation in SPaSM is equipped with four different strain rate functions: constant engineering strain rate, constant true strain rate, parabolic true

strain rate, and squared sinusoidal true strain rate.

Following calculations show how true strain rate and engineering strain rate is calculated for a given strain (ϵ) in SPaSM.

Since the strain is applied during n timesteps:

$$L_n = L_0(1 + \epsilon) \tag{3.65}$$

Where L_0 and L_n are original and final length of the simulation box.

Definition of true strain rate:

$$\dot{\epsilon}_{true} = \frac{L_{i+1} - L_i}{L_i} \times \frac{1}{\delta t} \tag{3.66}$$

Where $i = 0, 1, 2, ...n$, δt is the timestep, and L_i is the length of simulation box at i^{th} timestep. Thus, $n \times \delta t =$ compression time (t_{rise}).

Using equation 3.66, L_{i+1}:

$$L_{i+1} = L_i(1 + \dot{\epsilon}_{true} \times \delta t) \tag{3.67}$$

Thus, L_n:

$$L_n = L_0(1 + \dot{\epsilon}_{true} \times \delta t)^n \tag{3.68}$$

By substituting for L_n/L_0 from equation 3.65:

$$(1 + \dot{\epsilon}_{true} \times \delta t)^n = \epsilon + 1 \tag{3.69}$$

$$ln(1 + \dot{\epsilon}_{true} \times \delta t) = \frac{ln(\epsilon + 1)}{n} \tag{3.70}$$

By expanding $ln(1 + \dot{\epsilon}_{true} \times \delta t)$:

$$\dot{\epsilon}_{true} \times \delta t - \frac{(\dot{\epsilon}_{true} \times \delta t)^2}{2} + ... = \frac{ln(\epsilon + 1)}{n} \tag{3.71}$$

Since $\dot{\epsilon}_{true}$ and $\delta t \ll 1$, ϵ_{true} can be expressed by equation 3.72:

$$\dot{\epsilon}_{true} = \frac{ln(\epsilon + 1)}{n \times \delta t} = \frac{ln(\epsilon + 1)}{t_{rise}} \tag{3.72}$$

Definition of engineering strain rate:

$$\dot{\epsilon}_{eng} = \frac{L_{i+1} - L_0}{L_0} \times \frac{1}{i \times \delta t} \tag{3.73}$$

Thus,

$$\dot{\epsilon}_{eng} = \frac{L_n - L_0}{L_0} \times \frac{1}{n \times \delta t} = \frac{L_n - L_0}{L_0} \times \frac{1}{t_{rise}} \tag{3.74}$$

By substituting for $L_n - L_0$ from equation 3.65, ϵ_{eng} can be expressed by equation 3.75:

$$\dot{\epsilon}_{eng} = \frac{\epsilon}{t_{rise}} \tag{3.75}$$

In this study, all the simulations were performed under constant engineering strain rates. Thus, the true strain rate which appears in the QIC equations of motion (equations 3.53 and 3.54) changes with time according to the equation 3.80.

Following condition has to be satisfied for ϵ_{eng} to be constant.

$$\dot{\epsilon}_{eng} = \frac{L_1 - L_0}{L_0 \times \delta t} = \frac{L_2 - L_0}{L_0 \times 2\delta t} = \frac{L_i - L_0}{L_0 \times i\delta t} = \frac{\epsilon}{t_{rise}} \tag{3.76}$$

Thus, L_i:

$$L_i = L_0 \left(1 + \frac{\epsilon}{t_{rise}} \times i\delta t\right) \tag{3.77}$$

By using equation 3.77 to substitute L_{i+1} and L_i in to equation 3.66, $\dot{\epsilon}_{true}$ can be written as:

$$\dot{\epsilon}_{true} = \frac{L_0 \left(1 + \frac{\epsilon}{t_{rise}}(i+1)\delta t\right) - L_0 \left(1 + \frac{\epsilon}{t_{rise}} i\delta t\right)}{L_0 \left(1 + \frac{\epsilon}{t_{rise}} i\delta t\right)} \times \frac{1}{\delta t} \tag{3.78}$$

$$\dot{\epsilon}_{true} = \frac{\frac{\epsilon}{t_{rise}}\delta t}{\left(1 + \frac{\epsilon}{t_{rise}} i\delta t\right)} \times \frac{1}{\delta t} \tag{3.79}$$

By substituting $\dot{\epsilon}_{eng}$ from equation 3.75:

$$\dot{\epsilon}_{true} = \frac{\dot{\epsilon}_{eng}}{1 + \dot{\epsilon}_{eng} \times t} \tag{3.80}$$

Where, $t = i \times \delta t$.

Calculating Simulation Box Length Under Constant Engineering Strain Rate

Definition of true strain rate:

$$\dot{\epsilon}(t) = \frac{\dot{L}(t)}{L(t)} \tag{3.81}$$

Thus, true strain rate at $t = t + \frac{\delta t}{2}$:

$$\dot{\epsilon}(t + \tfrac{\delta t}{2}) = \frac{\dot{L}(t + \frac{\delta t}{2})}{L(t + \frac{\delta t}{2})} \tag{3.82}$$

From figure XX:

$$\dot{L}(t + \tfrac{\delta t}{2}) = \frac{L(t + \delta t) - L(t)}{\delta t} \tag{3.83}$$

also:

$$L(t + \tfrac{\delta t}{2}) = \frac{L(t + \delta t) + L(t)}{2} \tag{3.84}$$

By substituting 3.83 and 3.84 in 3.82:

$$\dot{\epsilon}(t + \tfrac{\delta t}{2}) = \frac{2 \times (L(t + \delta t) - L(t))}{\delta t \times (L(t + \delta t) + L(t))} \tag{3.85}$$

Thus, updated length of the simulation box:

$$L(t + \delta t) = \frac{\left(1 + \frac{\dot{\epsilon}(t + \frac{\delta t}{2})}{2} \delta t\right)}{\left(1 - \frac{\dot{\epsilon}(t + \frac{\delta t}{2})}{2} \delta t\right)} L(t) \tag{3.86}$$

Quasi-Isentropic expansion can also be achieved employing the same set of equations mentioned above by using expansive strain rate which is opposite in sign instead of compressive strain rate.

Implementation of QIC in MD

Following tasks were performed for incorporating the QIC equations of motion in SPaSM.

- Position and velocity updating routines of the leap-frog solver was modified to solve equation 3.53 and 3.54 respectively.
- Three new input parameters were introduced to collect final strain, compression time, and strain rate function from the user.
- Depending on the selected strain rate function, the value of $\dot{\epsilon}$ changes with time. Thus, $\dot{\epsilon}$ need to be calculated at every timestep. A new function called "setDeformation" was introduced for this purpose.
- Equation 3.57 assumes that center of mass of the system is at the origin. By default, SPaSM generate lattice in the positive octant. To shift the center of mass to the origin, x,y, and z coordinates of center of mass of the system is calculated and subtracted from atom positions.

The C source code of key routines that are required for implementing QIC is given in appendix A.

3.3 Inter-Atomic Interactions

Models for calculating interactions between atoms in MD simulations have been evolved and improved over time. In the early era of MD Lennard-Jones potential was popular for modeling atomic-interactions, since it was well suited for the computing machines available during that time. LJ interaction model is a pairwise additive force model and does not account for the changes in local electron density. However, in metallic crystals, valance electrons are shared among neighboring ions. Therefore, the inter-atomic interaction in metals

is mostly effected by change in local electron density due to neighboring atoms rather than pairwise additive force. Holian *et al.* [94] demonstrated plastic behavior in metals can be described better using a many body interaction model rather than solely using a pairwise model.

Later, more accurate but simple methods for modeling inter-atomic interactions were introduced for metallic systems by taking local electron density into the account. All these models are based on the many body energy function, but build on different physical arguments (tight binding model, effective medium theory, etc.). Followings are a few of these many body models:

- Embedded Atom Method (EAM) [95, 96, 97]
- Finnis Sinclair [98]
- Glue Model [99]

For large-scale MD simulations, EAM and Finnis Siclair potentials are often used due to their low computational cost. However, due to radial approximation of bond strengths without considering angular dependence, these potentials are well suited for metals with sp-bond or metals with filled or nearly filled d-electron bands such as the noble metals. Strong directional bonds exist in metals with partially filled d bands, thus, atomic interactions of these metals have to be treated by considering explicit angular force components. Later, quantum-based interatomic potentials were developed using tight binding theory[100, 101] and generalized pseudopotential theory[102] by including angular force contributions.

Atomic interaction models are generated to approximate actual interactions between atoms of single element type, alloys, or mixtures. As a result, their performance depends on the choice of parameters in the training process. Different models developed for the same element can generate completely different results in MD simulations. Zhang *et al.* [24] and

Ravelo *et al.* [22] demonstrated that different empirical interatomic potentials produce different deformation structures under non-equilibrium MD simulations of shock compression.

Thus, one has consider accuracy and limitations of the model in temperature and pressure domains when selecting a interaction model for MD. NIST Interatomic Potential Repository Project [103] maintains an archive of model potential in standardized data format for a number of elements, alloys, and compounds.

3.3.1 EAM

EAM is a semi-empirical radial-force method, based on Density Functional Theory (DFT), where the potential energy of any structural arrangement of nuclei is consider as a function of the positions of the atoms and the local electron density. At a low computational cost, this method can describe important mechanical properties of metallic systems. Thus, EAM is widely used for modeling atomic interactions in metals large scale atomic simulations. In this method, each atom is considered to be influenced by homogeneous and spherical electron gas of neighboring atoms and calculate energy required to embed the central atom in to this electron cloud. The potential energy of each individual atoms (E_i) is summed together to calculate total potential energy of the system.

$$E_{tot} = \sum_{i}^{N} E_i \tag{3.87}$$

The potential energy per atom i is given by,

$$E_i = F(\bar{\rho}_i) + \frac{1}{2} \sum_{i \neq j} \phi(r_{ij}) \tag{3.88}$$

where $F(\bar{\rho}_i)$ is the energy required to embed the atom i into a homogeneous electron gas of density ρ_i , and is a non-linear function of local electron densities imposed by the neighboring atoms (generally, within a given range of distance). $\phi(r_{ij})$ is a short-range

pair-wise potential (similar to Lennard-Jones) between atom i and it's neighboring atoms j and solely depend on r_{ij}: separation between atoms i and j. The electron density function (ρ_i) can be expressed by;

$$\bar{\rho_i} = \sum_{i \neq j} \varpi(r_{ij}) \tag{3.89}$$

Where $\varpi(r_{ij})$ is a spherically symmetric function.

Generally, EAM model potentials are generated by fitting a selected training set of experimental (*e.g.* elastic constants, vacancy formation energy and Grüneisen parameter) and quantum MD data (*e.g.* stress tensor, unrelaxed surface energy, and unstable stacking fault energy) in to a parameterized functional form. The selection of the training set depend on the material and thermodynamical conditions under investigation.

It can be argued that due to the fact that EAM assumes electron density of an metallic iron is spherically symmetric, mechanical properties of metal which have directional bonding associated with partially filled d bond can not be expected to be predicted by EAM. However, Ravelo *et al.* [22] demonstrated that by carefully selecting high pressure properties in the training set, EAM can be used to predict mechanical properties of bcc Ta under shock loading.

Discretized tabular format is used for storing EAM model potentials. For example, the pair-wise potential $\phi(r_{ij})$ is stored in a two column table with each row corresponding to a different r_{ij} value. Thus, an EAM potential that model a single element would be comprised of three tables each for $\Phi(r_{ij})$, $\varpi(r_{ij})$, and $F(\bar{\rho})$. An EAM potential that model interaction between more than one type of element should have additional table to account for heterogeneous interactions. For example, seven tables are required to model a binary alloy (three tables for pair-wise interactions: $\phi_{AA}(r_{ij})$, $\phi_{BB}(r_{ij})$, and $\phi_{AB}(r_{ij})$, two tables

for electron density function: $\varpi_A(r_{ij})$ and $\varpi_B(r_{ij})$, and two tables for embedding energy function: $F_A(\bar{\rho}_i)$, and $F_B(\bar{\rho}_i)$).

EAM Model Potentials for Tantalum

Ravelo *et al.* [22] developed two EAM model potentials (Ta1, and Ta2) for tantalum and evaluated them by comparing simulated Hugoniot elastic limit (HEL), melt curve, and defect structures under shock compression against experimental data. Both of these model potentials have been used to produce mechanical properties of Ta at the same degree of accuracy along with similar morphology of defect structures and densities. bcc to hcp phase transition has been predicted to occur at pressures of 463 GPa and 325 GPa for Ta1 and Ta2 respectively. Liu *et al.* [104] reported that bcc is the most stable structural phase in the pressure range of 0 GPa to 300 GPa for Ta2 potential. However they predicted a bcc to pnma phase at about 110 GPa for Ta1 potential by investigating the enthalpy differences of each phase.

Ta1 and/or Ta2 potential have been widely used in computational studies for investigating material strength [80, 105], plastic deformation under shock compression and nanoindentation [106, 107], high pressure melting curve [104] of tantalum up to 300 GPa. Slight differences between the two potentials are known to present due to different descriptors used as training set. As a result, Ta2 is claimed to represent thermal pressures accurately than Ta1.

The parameters of Ta2 potential functions are given in table 3.1. In this study Ta2 potential in tabular form of 5000 points (for each tables) was employed for modeling atomic interactions of tantalum. Thus, the potential tables occupied approximately 234.4 KB in memory. Cut-off distance (r_{max}) for atomic interactions was 5.5819 Å, meaning that neighboring atoms beyond this point does not exert any force on the central atom. Intermediate

values were obtained by employing a linear interpolation function.

Table 3.1: Parameters of the EAM function of Ta2 potential.

Parameter	Ta2
E_c (eV)	8.1
α	4.95
a_0 (Å)	3.304
f_3	-0.035744
f_4	-0.020879
U_0 (eV)	1.1094
r_1 (Å)	2.7826
α_P	4.6463
β_3	0.160
β_4	0.060
r_s (Å)	2.8683
s	8.00
r_c (Å)	5.5819
a_1 (Å^{-s})	-0.039742
a_2 (Å$^{-(s+1)}$)	0.038568
a_3 (Å$^{-(s+2)}$)	-0.012714
a_4 (Å$^{-(s+3)}$)	0.0014196
p	5.9913
q	8.0
ρ_0	0.074870

3.4 Evaluation of Thermodynamic Quantities

Pressure

The virial equation has been the standard method of calculating pressure in MD simulations. Louwerse *et al* [108] demonstrated that virial equation (equation 3.90) does not hold when periodic boundary conditions are used. However, Louwerse also showed that for systems described with pair-additive force fields the pairwise version of virial equation (equation 3.91) still hold simply due to errors in the flawed derivation cancel out.

$$P = \frac{Nk_BT}{V} + \frac{1}{3V}\langle\sum_{i=1}^{N}\mathbf{r}_i.\mathbf{f}_i\rangle \tag{3.90}$$

Where k_B is the Boltszmann's constant, $\mathbf{f}_i$ is the resulting force acting on atom i due to all other atoms, and $\langle\rangle$ represents statistical average over time.

$$P = \frac{Nk_BT}{V} + \frac{1}{6V}\langle\sum_{i=1}^{N}\sum_{j\neq i}^{N}\mathbf{r}_{ij}.\mathbf{f}_{ij}\rangle \tag{3.91}$$

Where, f_{ij} is the force on atom i due to jth atom and $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$.

Thompson *et al.* [109] derived three distinct forms of virial equations to calculate pressure and stress tensor of group of atoms confined under periodic boundary conditions. All of these forms are claimed to be produce accurate results regardless of the potential function.

The global stress tensor $\mathbf{P}$ can be given by:

$$\mathbf{P}V = \langle\sum_{i=1}^{N} m_i\mathbf{v}_i\mathbf{v}_i + \mathbf{W}(\mathbf{r}^N)\rangle \tag{3.92}$$

Where, $\mathbf{W}(\mathbf{r}^N)$ is expressed by:

$$\mathbf{W}(\mathbf{r}^N) = \sum_g \sum_{i=1}^{N} \mathbf{r}_i \mathbf{F}_i \tag{3.93}$$

Here, g represents the group or the subdomains.

The kinetic energy term in equation 3.92 is effected by the velocity of the center of mass. For eliminating this effect, the center of mass velocity of the simulation box was subtracted from atomic velocities before using in equation 3.92.

Temperature

Temperature was calculated as:

$$T = \frac{1}{n} \sum_{i=0}^{N} \frac{m}{3k_B N_A} \left(v_{i,x}^2 + v_{i,y}^2 + v_{i,z}^2 \right) \tag{3.94}$$

Where m is atomic mass of tantalum, N_A is the Avogadro's number, $v_{i,x}$, $v_{i,y}$, $v_{i,z}$ are x, y and z velocity components of ith atom.

Chapter 4

Massively Parallel Molecular Dynamics

In this chapter, MD codes that are extensively employed in massively parallel supercomputer as well as a co-design approach to improve key performance characteristic kernels of these code to adapt emerging computer architectures and execution models will be discussed along with results obtained from my study of performance evaluation of MD kernels using a proxy application.

Since, Alder and Wainwright's first ever MD simulation of few hundreds of atoms in 1957 capabilities efficiency of computer hardware, instruction execution models, grew immensely allowing simulating billions of atoms [110, 111] a possibility by early 2000's. Germann *et al.* [112] conducted the First ever trillion atom simulation on the BlueGene/L system at Lawrence Livermore National Laboratory using Lennard-Jones particles. BlueGene/L could theoretically perform 590×10^{12} floating point operations (Flops) compared to 1000 Flops in the early vacuum tube machines that Alder and Weinwright used. At the moment, fastest supercomputer in the world, Sunway TaihuLight in National Supercomputing Center in Wuxi China has a theoretical peak performance of 93 PFlops [113].

Dynamic response of material has been extensively studied using variety of massively parallel and highly optimized large scale classical MD codes such as SPaSM (Scalable Parallel Sort-range MD)[32, 33, 34], ddcMD[111, 114], and LAMMPS[115]. However, classical MD schemes are notorious for not reaching anywhere near to the peak performance of a

supercomputer due to it's inherent features such as atomic interaction calculation, redistribution of atoms. Most time consuming task of all of these implementations is evaluation of atomic interactions. Improvements of the underlying hardware alone is not sufficient, but revolutionary advances in underlying algorithms, communication schemes, and data distributions method are also need to be utilized to harness the maximum potential of the supercomputers with hundreds of thousand of cores.

4.1 Parallelization Methods

Modern day supercomputers carry millions of processor cores[113] and continue to grow in numbers. A typical MD code spend about 90% of the execution time on calculating atomic interaction[116, 117]. Thus, depending on the type of atomic interaction, whether they are simple short-range or hybrid of short-range and long-ranged ironic interactions, all of the parallel MD codes employ workload distribution strategies along with Message Passing Interface (MPI) in order to speed up the force evaluation.

4.1.1 Atom Decomposition (AD)

In this method atoms are distributed among compute tasks regardless of position in the coordinate system. Thus each processor calculates forces on a subset of atoms. Typically, this decomposition method is suitable for systems of particles where a given particle interact with every other or most of the particles in the system. As a result, the entire array of the atom data has to be stored in each of the compute tasks leading to poor scaling. Figure 4.1 (a) shows a schematic of the atom distribution method where atoms as well as the atom data array are colored according to the processor which evaluate forces of that particular atom.

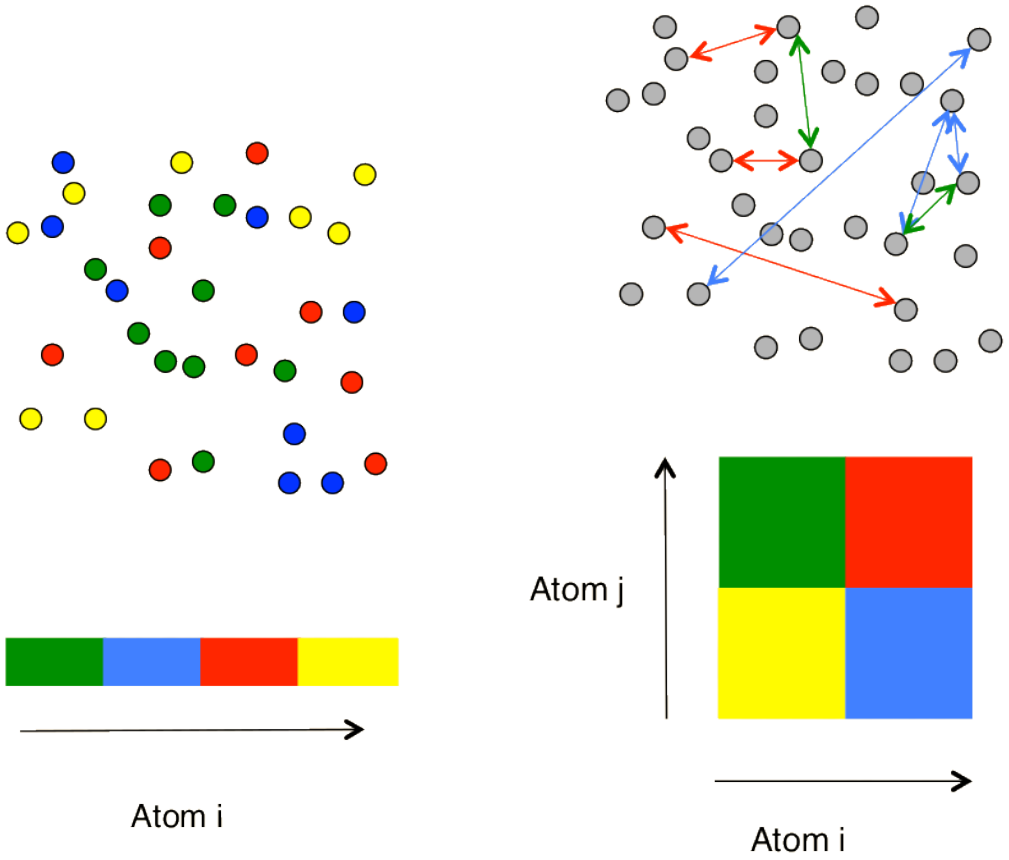


Figure 4.1: Schematics of atom decomposition (a) and force decomposition (b).

4.1.2 Force Decomposition (FD)

In this method a subset of the entire force evaluation is distributed among compute tasks. Thus, in contrast to AD, only a subset of particle data array is stored in a compute task leading to better scaling. Figure 4.1 (b) shows a schematic of the force distribution method. In figure 4.1 (b), different colors of arrows between atoms represent the processor which calculate them as well as different colors in atom data array, typically a 2D array, illustrates in which processor atoms are stored.

Both of these decomposition methods resemble a Lagrangian gridding in a simulation

of fluids where the grid cells change with the fluid. Further, the assignments of atoms to processors remain fixed throughout the entire simulation, as a result, redistribution of particles is not necessary.

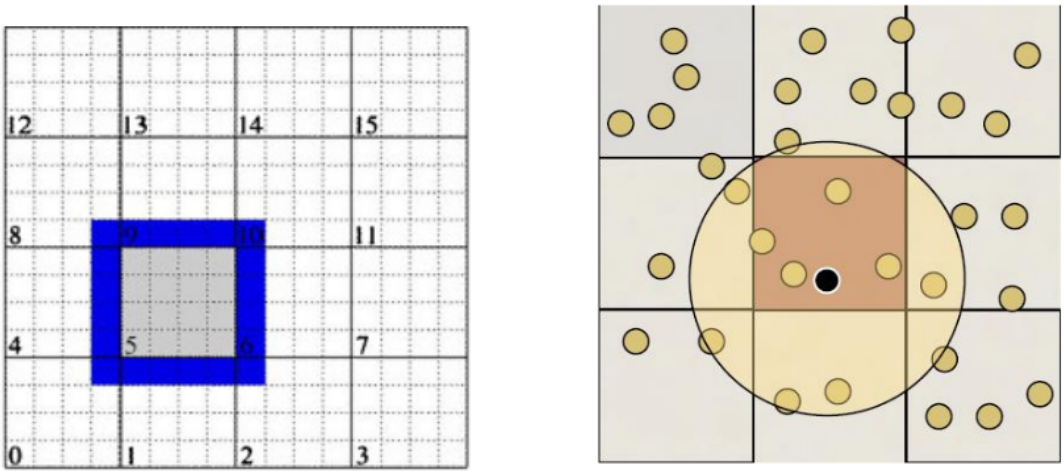
4.1.3 Spatial Decomposition

In spatial decomposition, the workload is parallelized by assigning calculations on different sections of the simulation box into computing cores. The grid defined by boundaries of the compute tasks that confine the sub-domains are generally fixed (Eulerian sub-domains) and atoms are allowed to move through it. Thus, redistribution of atoms are necessary. This initial decomposition guarantees each compute core perform equal amount of calculations assuming the atomic density is homogeneous throughout the simulation box. Spatial-decomposition is well suited for systems that can be modeled by a short-range potential, with approximately uniform spatial distribution of atoms and relatively low densities. Figure 4.2(a) shows an example of initial decomposition of a 2D simulation box among 16 compute cores.

In short range classical MD simulations atomic interactions beyond third neighbor shell is negligible. Thus, interaction between two atoms separated by more than a certain value of cutoff (r_{max}) is not calculated: i.e if atom i interacts with atom j the inequality given in 4.1 has be satisfied where r_{ij} is the distance between atom i and atom j and r_{max} is a property of employed atomic interaction model respectively.

$$r_{ij} \leq r_{max} \tag{4.1}$$

The value of r_{max} is used for dividing the sub-domains into smaller cubical cells with edges greater than r_{max}. These smaller domains are called *link-cells* and facilitate optimal



(a) (b)

Figure 4.2: Initial decomposition of a simulation box among compute nodes (a)[119], and Indra-core link-cell decomposition (b)[116].

execution of force evaluation kernels by reducing brute force search for neighboring atom in to a smaller local region[118]. Figure 4.2(b) shows an example of two dimensional link-cell decomposition in 2D. In a three dimensional link-cell formulation atoms in a particular cell can find all of it's interacting neighbors within the it's own cell and 26 neighboring link-cells.

r_{max} of the interaction model act as a lower bound for link-cell size ($l_{link-cell}$):

$$l_{link-cell} > r_{max} \tag{4.2}$$

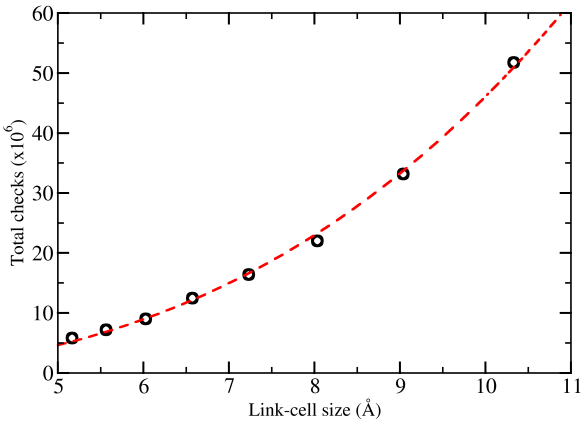
However, average number of atoms in a link-cell will cubically increase with increasing link-cell size. As a result, number of atoms that has to search through in the selection process of interacting neighbors will also increase cubically hindering computational performance of the simulations. Figure 4.3 shows total number of conditional evaluations of inequality 4.1 performed by a compute task for selecting interacting neighbors (a), and the execution time of a timestep (b) as a function of link-cell size. These plots were generate by test simulations of 1.5 million copper atoms distributed among 48 compute cores.

Atomic interactions were modeled by an EAM potential[120] with a r_{max} of 4.95 Angstroms.

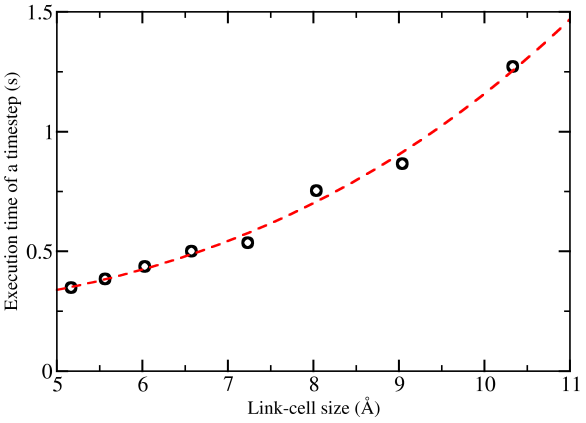
A compute task will require cells that reside in neighboring compute tasks to evaluate forces of atoms belong to it's outer shell of link-cells. As a result, additional steps of communication between compute tasks have to be introduced in to the force evaluation routines. For example, the compute task that is sub-domain number 5 in figure 4.2(a) is assigned will require atom information from cells colored in blue. Typically, this blue region is called halo region of compute task 5 and copy of the cells in this region is stored in compute node 5 to avoid frequent communication. However, some halo communications are still unavoidable: derivatives of embedding function of neighboring atoms are needed for calculating atomic forces in EAM. Halo regions are always updated after advancing atomic positions by the MD solver. Number of halo cells a compute task need to store depends on the dimensions of the it's sub-domain and can be minimized by cubical decomposition of sub-domains.

Fault Tolerance

Despite the sophisticated technology and considerable effort of microchip engineers employed in designing and fabricating, compute cores are still susceptible to hardware error. The mean time between failures (MTBF) is a measurement of this error and is roughly about 2 or 3 years in a typical CPU used in desktop computers. However, due to immense number of processors, interconnects, and other components, MTBF can be a few minute in a large scale supercomputer[111]. The most frequent hardware error in a typical supercomputer is transient parity errors on the L1 cache line. This error can be mitigated by triggering a writes through of the L1 data cache directly to a lower level memory subsystem, which correct the corrupted L1 memory line by reading uncorrupted data from L3 cache at the cost of reduced performance. An application assisted error recovery methods, Which transfer control to an interrupt handler when an parity error is detected, can also



(a)



(b)

Figure 4.3: Total number of checks performed by a compute task for selecting interacting neighbors (a), and execution time of a timestep (b) as a function of link-cell size.

be implemented in to the software.

4.2 SPaSM

Peter S. Lomdal and Dave Beasley developed SPaSM, a massively parallel MD code in the 90's at Los Alamos National Laboratory. Since then, SPaSM has been successfully used for studying scientific and engineering research areas such as shock compression[26, 23], solid-solid phase transitions[121, 122], dislocation intersection [123], and crack propagation[124]. By considering each particle is a person and replacing atomic interaction by stochastic rules of disease transmission between humans, the core features of the code was later adapted to study a non-MD related problem: spread of diseases using an agent-based simulation model[125]. SPaSM also possess the ability to visualize, filter and analyze large system sizes on-the-fly.

SPaSM employs special decomposition to effectively distribute workload among processors using MPI along with link-cells to facilitate efficient force evaluation. Thus, section 4.1.3 provides important information on SPaSM's parallelization strategy.

Recent optimizations of the code Blue-Gene/L have allowed simulations of a trillion of atoms using LJ potentials[112]. Typically, performance of MD is constrained by memory and floating-point operations. The early implementations of SPaSM addressed this issue by frequent inter-processor communications of small messages to conserve memory and by avoiding redundant computations. However, the most prominent constrain of a production level MD simulation is wall-clock time. A redesigned implementation of SPaSM for hybrid supercomputers which uses accelerated processing unit with large number of parallel processing units for evaluating atomic interactions and conventional CPU for particle update, checkpointing and communication routines has achieved approximate speedup of ten

times[119].

4.3 Proxy Applications

In recent years, High Performance Computing (HPC) has been introduced with a variety of new hardware architectures. Two major type of these architectures can be identified as one being stand-alone architectures such as Intel Many Integrated Core (MIC)/Intel Xeon Phi Knights Landing, and accelerators that work side by side with a CPU, such as General Purpose Graphical Processing Units (GPGPU or GPU). The general CPU architecture also keeps on evolving with advances of underlying technologies. Optimum execution of scientific applications on these new architectures and as well as in a typical CPU is essential since the current generation's supercomputing facilities are equipped with novel hardware devices.

The parameter space that the performance of a large scale MD code depends on is overwhelmingly large and can be related to software/algorithm or hardware/architecture. Evaluating new algorithms and hardware using sophisticated production level MD codes can be time consuming and expensive. Thus, mini applications that mimic the key algorithmic complexity are widely used for testing purposes.

These mini applications are called proxy applications and serve as frameworks in which the hardware-software interactions can be easily explored, since these simplified applications are more amenable to testing and analysis than existing production applications and available to external collaborators. As a result, both the hardware vendors and software developers can co-optimize their products to ensure that future machines are able to provide the Exascale[126] performance.

4.3.1 CoMD

CoMD is a proxy application that represents the typical workload and use cases of short range classical MD and represents the fundamental workflow in such simulations from problem setup, equilibration, time integration, and, analysis/visualization. CoMD was mainly based on the production codes such as SPaSM and ddcMD with the intention of algorithmic improvements and optimizations will eventually be incorporated into the classical MD codes. The short range atomic interactions can be calculated either by Lennard-Jones pair potential or by EAM. By default, CoMD solves a NVE ensemble of copper single crystal using verlet integration. CoMD was a part of the Mantevo Project [127] which received the 2013 R&D 100 award. CoMD has been implemented for wide range of computer architectures such as MPI, OpenMP, OpenCL and CUDA.

Similar to other production MD codes, CoMD also spends most of the execution time in atomic force evaluation. In the case of CoMD, 90% of the execution time is spent on the EAM force evaluation. CoMD has been vastly used for studying performance and energy consumption of MD code on different computer architectures[128, 129].

Domain Decomposition

Spatial decomposition is employed in CoMD. The user is required to specify the number of available compute tasks and their distribution along each Cartesian axis. Thus, each compute task will be assigned a section of the initial lattice (a cube or a cuboid) and perform integration of equations of motion only for the atoms that reside inside the local sub-domain. In CoMD each compute task keeps information of atoms that reside in it's halo region.

The local sub-domain of the simulation cell that belong to a given compute task is then

divided in to link-cells. The linkcell could be a cube or cuboid depending on the dimension of the local domain. The number of link-cell in a particular direction is calculated by dividing the length of local domain by the cutoff distance in interatomic potential and rounding to the lower integer.

Data Layout

Two different techniques can be used to store atom data in computer memory; structure of arrays (SoA), and array of structure (AoS). the optimum data layout mainly depends on the memory access pattern of the underling computer architecture. In SoA method, atoms'unique data fields are stored in separate arrays. For example, in SoA, x-coordinates of all the atoms in the domain are consecutively stored in a double (or single) precision array, while in AoS, x,y and z coordinates (thus, three-fold vector of double data type) of an atom is stored in consecutive memory locations followed by the same of the next atom. An illustration of both of the data layout methods is shown in figure 4.4.

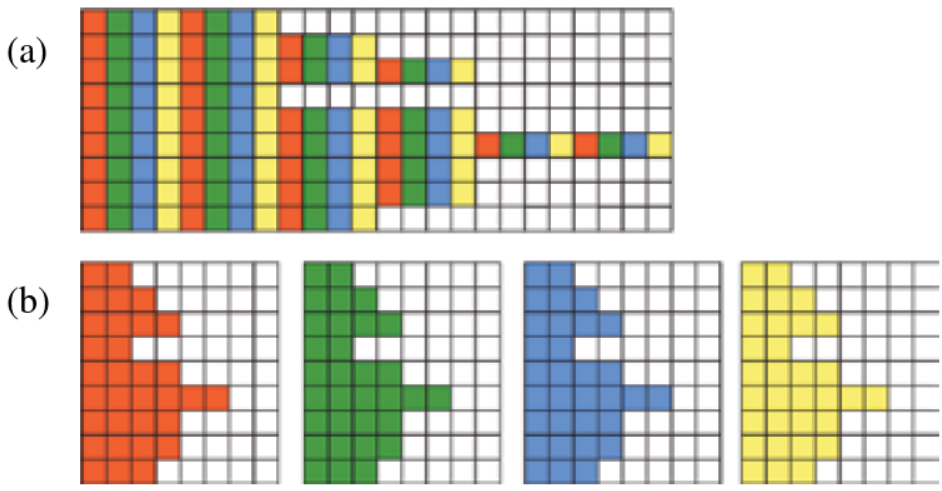


Figure 4.4: Data layout techniques in CoMD: (a) array of structures (AoS), (b) structure of arrays (SoA). This figure was adapted from reference [116].

CoMD has implemented both of these data layout methods in it's base version, therefore the best layout for a given memory access pattern can be selected at the compilation stage of the code.

GPU Implementations of CoMD

As a part of the Co-Design effort, NVIDIA has developed a CUDA implementation of CoMD which is highly optimized for NVIDIA GPUs[130]. This version is capable of achieving order of magnitude performance improvements over the basic CPU implementation of CoMD 1.1. Followings are the extra features included in the CUDA-CoMD version.

- Neighborlist to improve performance of force evaluation kernels.
- Space filling curve method was used as the access pattern for atoms instead of Cartesian access pattern. This improves the cache hit ratio when accessing the atom data.
- CUDA Priority streams for overlapping communication with force evaluation of interior atoms.
- Atom data are stored in SoA

In the GPU execution model, consecutive memory access and divergent free warps are crucial for performance improvement. The CUDA implementation of CoMD introduced an extra array for storing atom indices and used it as the base for generation of thread blocks, assuring coalesce access for atom data and warp efficiency. CUB library was used to perform parallel scan in order to generate this array of atom indices. Employing CUDA specific intrinsic function calls and texture memory for read only data such as neighborlist and potential table value, the code was able to increase the memory throughput.

Similar to the CUDA-CoMD, an OpenCL implementation of CoMD has also built upon the basic CoMD 1.1 version. However, force evaluation in the OpenCL implementation does not target a particular architecture. Thus, this version could be compiled and executed on most of the multithreaded computer architectures (such as, GPUs, Intel MICs, and, multicore CPUs given the appropriate drivers and libraries are installed).

4.4 Performance Testing Using CoMD

4.4.1 Performance Comparison Between Binary and Single Lattice

Performance of the EAM force evaluation was studied by implementing bi-crystal force evaluation routines into CoMD. A lattice with Copper and Silver atoms of type $L1_2$ was created in which Copper atoms are located in the faces of an fcc crystal and Silver atoms are located in corner of the fcc unit lattice. Figure 4.5 shows a unit cell of the $L1_2$ structure. The Cu-Ag EAM potential table comprised 4000 points for Copper and 3000 points for Silver [131]. The Cu-Ag binary EAM potential was made up of seven different tables. Four of these tables described the Cu electron density, Cu embedding function, Cu-Cu pair-wise interactions, and Cu-Ag pair-wise interactions and are comprised of 4000 data points while the rest of the tables (Ag electron density, Ag embedding function, and Ag-Ag pair-wise interactions) contains 3000 points each. CoMD data structure, which holds the potential data, has also been modified in order to accommodate extra tables. This new setup increases the size of the memory pool randomly accessed by EAM kernels. As a result, cache miss ratio is increased in the binary force evaluation.

Figure 4.6 show the force evaluation time per iteration. The data was gathered on an Intel Xeon X5355 CPU. The binary lattice spent more time on atomic interaction

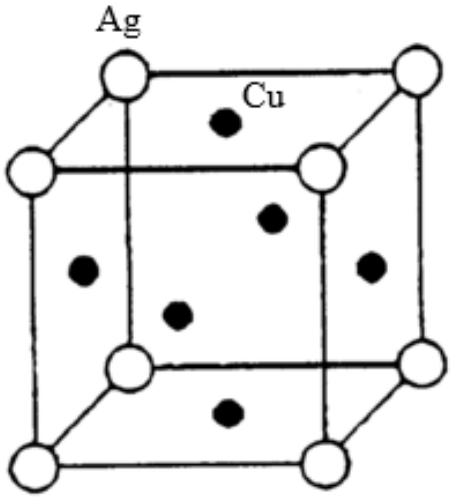


Figure 4.5: Unit cell of an $L1_2$ structure with Cu atoms in the faces and the Au atoms in the corners of the fcc structure.

calculations primarily due to the increase in the number of floating point operations in the force evaluations kernels (about 15%). Another factor, which can affect performance, is the number of atoms per linkcell. However, this will depend on the crystal structure of the binary system. In this study we tried to avoid this complication by choosing the $L1_2$ structure for the Cu-Ag system, which is structural equivalent to the mono-atomic Cu lattice in Co-MD.

4.4.2 Performance Improvement Using Neighborlists on CoMD-OpenCL

Improvements of force evaluation of OpenCL version of CoMD was studied by implementing neighborlists that store indices of neighboring atoms within a given cutoff distance which is set to be bigger than the cutoff distance of the interatomic potential. by adding a buffer (also called "skin") that holds the atoms which might interact with the central atom in future timesteps. Thus, higher skin sizes can reduces the frequency of the neighborlist updates, however lead to larger link-cell sizes and diminish performance. Atomic vibrations

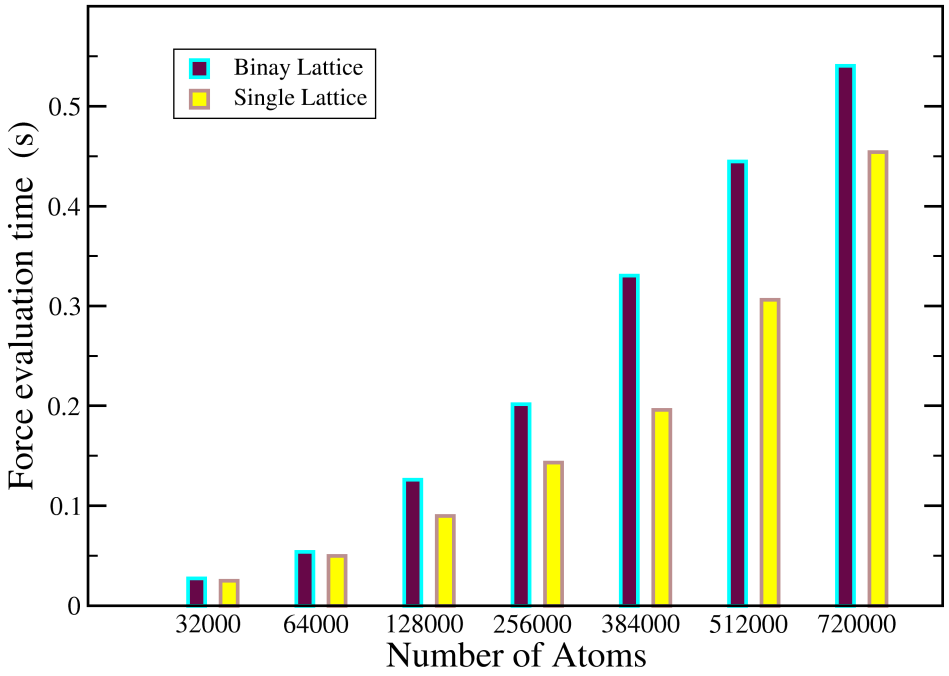


Figure 4.6: Force evaluation time for binary lattice and single lattice. It is clearly noticeable that the binary lattice spent more time for evaluation of EAM forces due to the increased number of calculation in the binary force evaluation.

are higher at higher temperatures and increased the probability of a new atom moving in to the interacting distance of the central atom. Thus, higher temperature increases the frequency of neighborlist updates. Table 4.1 summarizes the average timesteps between two consecutive neighborlist updates. In this table, "inf" represents the instances that never required a neighborlist update. It is noticeable that for a given skin size higher temperature required more frequent neighbor list updates.

The performance improvements by neighborlists was studied using NVIDIA Tesla C2070 GPUs. Figure 4.7 shows the evaluation time of EAM kernels with and without neighborlists.

Table 4.1: Average timesteps between consecutive neighborlist updates at different temperatures and relative skin sizes.

	Temperature (K)			
Relative skin size	600	1500	3000	5000
10%	30.3	17.5	11.7	9.2
20%	inf	43.5	24.5	19.2
30%	inf	inf	42.0	34

The skin size was set to be 20% of the cutoff distance of the EAM potential. According to figure 4.7 it is clearly noticeable that the neighborlist improved the performance of EAM force evaluation. The performance speedup achieved by the modified EAM kernels with neighborlist was approximately two times compared to original EAM kernels.

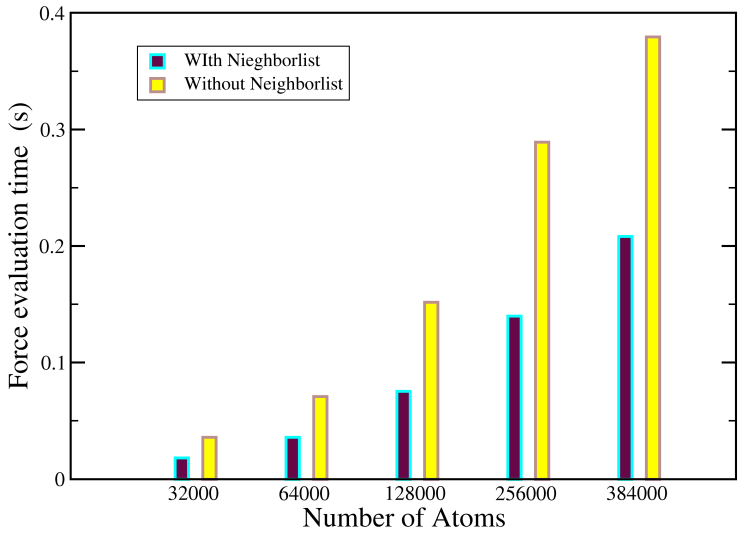


Figure 4.7: Material strength of tantalum as a function of strain rate. Yielding shear stress obtained from QIC simulations are represented by green squares and a generalized Lorentziam fit to this data is drawn in black. Von Misses stress for tantalum obtained from PTW model is given by the the red line.

Chapter 5

Extended Defect Identification

5.1 Central Symmetry Parameter (CSP)

CSP is a method for identifying structure around an atom developed by Kelchner *et al.* [37]. This method utilize the fact that in a centrosymmetric lattice a pair of equal and opposite bonds can be found extending from a central atom to it's nearest neighbors and the sum of the vectors pointing to these opposite atom pair from the central atom should be equal to zero in a perfect lattice. Value of CSP will remain zero under homogeneous elastic deformation as the paired bonds will still remain equal and opposite even though their length and direction is effected. However, atomic vibrations due to temperature will cause CSP to slightly deviate from zero while structural defect in the lattice will cause CSP higher deviations. Therefore, a proper threshold value can be use to distinguish defective central atoms from a central atom of perfect local lattice. The CSP can be calculated using equation 5.1 for an atom with n nearest neighbors.

$$CPS = \sum_{i=1}^{n/2} (\vec{r_i} + \vec{r}_{i+n/2})^2 \tag{5.1}$$

Where, $\vec{r_i}$ and $\vec{r}_{i+n/2}$ are the vectors pointing to ith pair of opposing neighbors.

CSP is a scalar value, thus identifications of different defect structures and orientations using this method is limited. Effect of thermal vibrations cause the CSP classification of defects to become less reliable at higher temperatures. Figure 5.1 shows stacking faults

identified using CSP in a 9.5% strained copper crystal. Figure 5.2 (obtained from reference [132]) shows distribution of CPS at different temperatures in a perfect fcc lattice and inside a intrinsic stacking fault in copper.

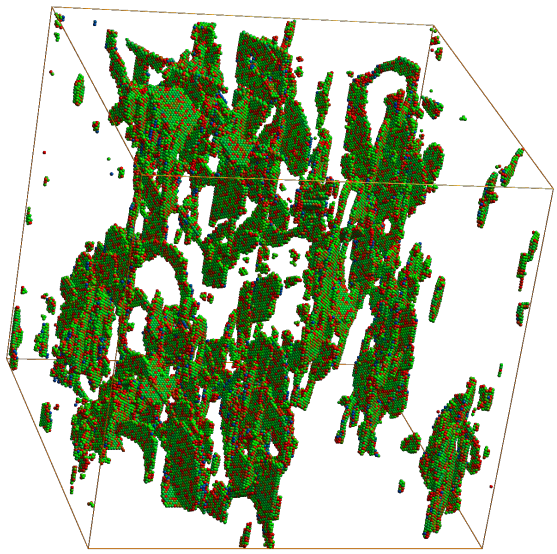


Figure 5.1: Intrinsic stacking faults in a copper crystal compressed to a strain of 9.5%.

5.2 Common Neighbor Analysis (CNA)

CNA[84, 133] is a structure analysis method that employ three characteristic indices (ijk) calculated using each of the neighbors and their bonds with the central atom:

- j: Number of common neighbors
- k: Number of bonds between common neighbors
- l: Number of bonds in longest bond chain formed by common neighbors

For computing ijk of closely packed lattices, such as fcc and hcp, a pair of atoms are considered to be bonded if they are within each others first neighbor shell. In contrast for

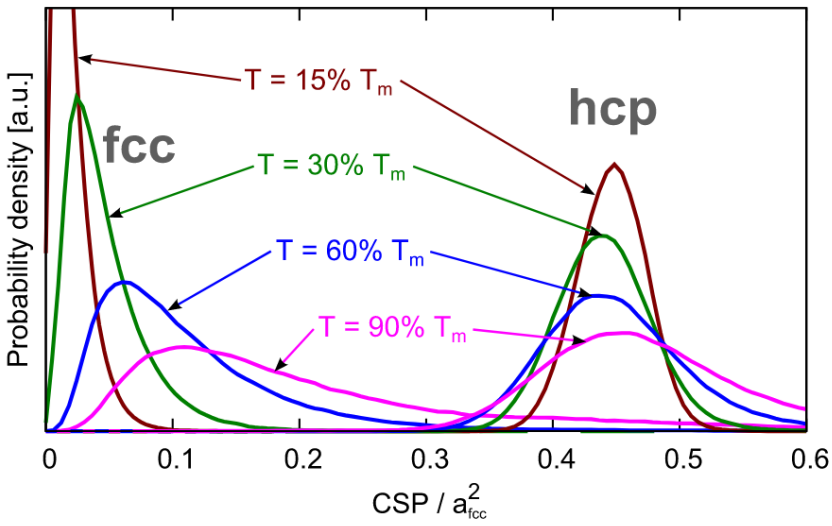


Figure 5.2: Distribution of CSP values of atoms in a copper crystal with a stacking fault corresponding to hcp-like atomic arrangement at different temperatures. This figure was obtained from reference [132].

the bcc lattice, which allow them to be within first two neighbor shells. If an atom has n number of neighboring atoms, above analysis will yield n number of indices, which can be compared against a set of reference structures to identify local structure around the central atom.

Tsuzuki *et al.* [134] developed an alternative method that calculate a single parameter called *common neighborhood parameter* (CNP) by combining both CNA and CSP methods. CNP for atom i can be calculated using equation 5.2.

$$CNP_i = \frac{1}{n_i} \sum_{j=1}^{n_i} \left(\sum_{k=1}^{n_{ij}} (\vec{r}_{ik} + \vec{r}_{jk}) \right)^2 \tag{5.2}$$

Where n_i is the number of neighbors of atom i, n_{ij} is the number of common neighbors between atom i and j. $\vec{r}_{ik}$ is the vector connecting atom i to atom k, and $\vec{r}_{jk}$ is the vector connecting atom j to atom k. Figure 5.3 shows a schematic diagram of neighborhood of

atom i. CSP take in to account the state of bonds between the central and common neighbor atoms as well as the bonds between paired atom (j) and the common neighbors.

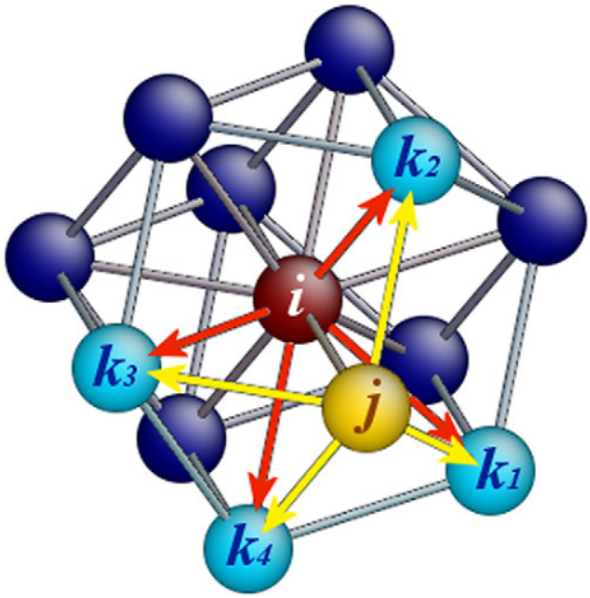


Figure 5.3: Illustration of the common neighborhood used in the calculation of the CNP_i parameter for the hcp structure. This figure was obtained from reference [134].

In a multi-phase crystal structure, the cutoff value of atomic distances to determine neighbors of given atom is ambiguous. To overcome this drawback, Stukowski *et al.* [132] propose a method called *adaptive common neighbor analysis* (a-CNA), that select the cutoff of the atomic distance individually for each atom depending on the reference structure that is being compared with.

5.3 Orientation Imaging Map (OIM)

Ravelo *et al.* [22] employed a three dimensional orientational order parameter (d_{100}, d_{110}, d_{111}) in order to readily visualize twins in a shocked sample. In this method, the order parameters are calculated for each centrosymmetric atom. d_{100}, d_{110}, and d_{111} are scalars

that represents misorientation of the local structure from ideal (100), (110), and (111) orientations along z-axis respectively. Thus for an atom with a local structure that is perfectly oriented or slightly misoriented from (100) along z-axis will have zero or smaller value for d_{100} and larger values for d_{110} and d_{111}. The normalized inverse triad of these misorientations (d_{100}^{-1}, d_{110}^{-1}, d_{111}^{-1}) is then mapped onto an RBG color triangle (shown in figure 5.4) using equation 5.3.

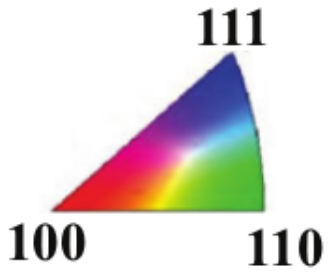


Figure 5.4: (a): OIM color map employed by Ravelo *et al.* [22].

$$q_{oim} = (R, G, B) = \left[\min\left(255, \frac{\sqrt{d_{100}^{-1}}}{d_{norm}}\right), \min\left(255, \frac{\sqrt{d_{110}^{-1}}}{d_{norm}}\right), \min\left(255, \frac{\sqrt{d_{111}^{-1}}}{d_{norm}}\right) \right] \quad (5.3)$$

Where $d_{norm} = \sqrt{d_{100}^{-1} + d_{110}^{-1} + d_{111}^{-1}}$.

Figure 5.5 shows twin structures in a (100) oriented single tantalum crystal. Atoms colored in red represents un-twinned atoms and atoms colored in green represents the atoms in twins.

Figure 5.5: Twin structures in a (100) oriented single tantalum crystal.

5.4 Dislocation Density Analysis

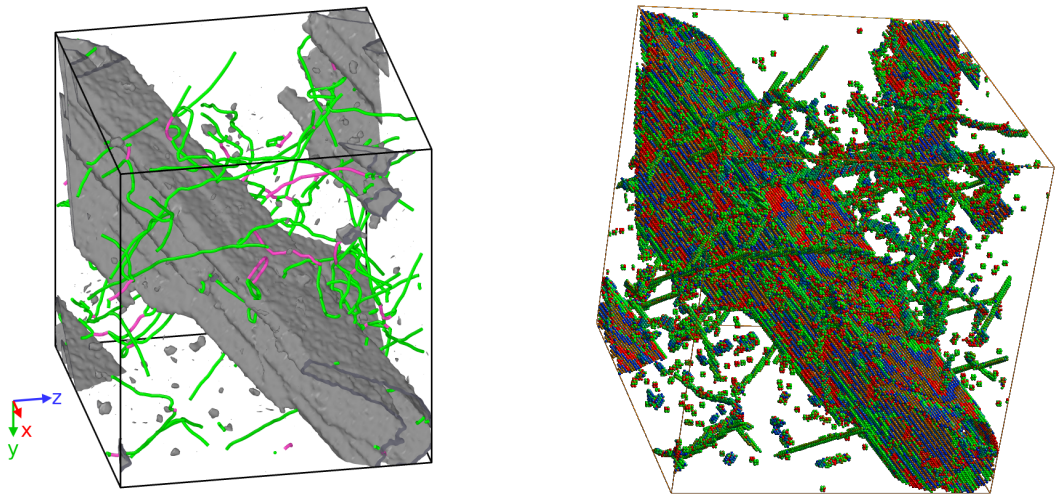
The Dislocation Extraction Algorithm (DXA), developed by Stukowski *et al.* [39, 38],is a robust method for identifying types of dislocations in a given crystal structure and calculating Burgers vector and length for each of the detected dislocation lines. Most of the steps in DXA are generic and does not depend on the underlying crystal structure. Figure 5.6 shows dislocations and defect structures obtained from implementation of DXA in OVITO[135] (figure 5.6 (a)) and by filtering perfect atoms using CSP (figure 5.6 (b)).

5.4.1 Dislocation Analysis Algorithm

DXA processes a given crystal structure in three principal steps:

1. The Common Neighbor Analysis (CNA) [84] is performed on the structure to distinguish crystalline atoms from defective atoms.

2. A closed, orientable, two-dimensional interface mesh) is constructed that separates



(a) (b)

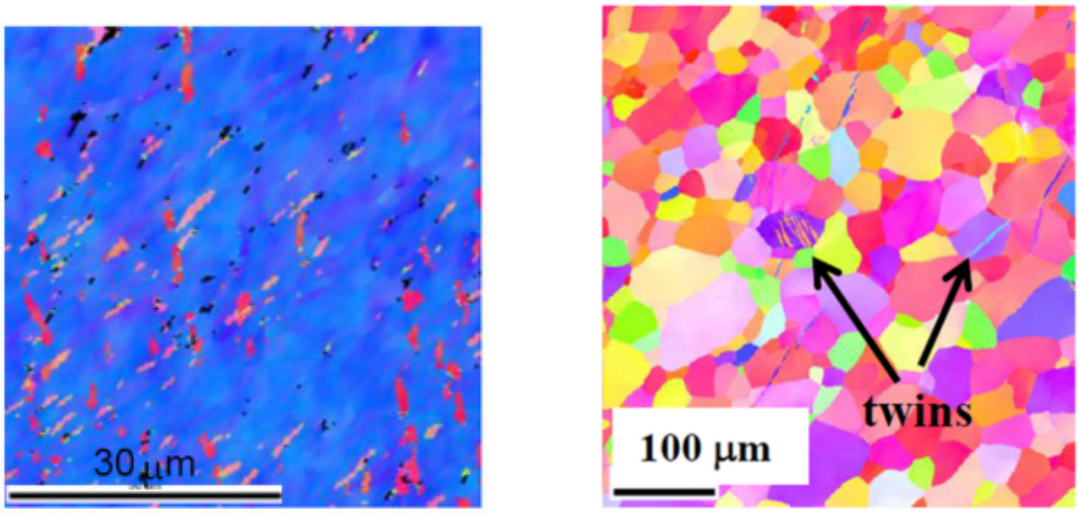
Figure 5.6: Dislocations and defect structures obtained from DXA algorithm (a) and using CSP with a cutoff value of 5.0 (b).

the crystalline atoms from the disordered.

3. For each dislocation segment, Burgers vector is calculated moving through the available Burgers circuits.

5.5 Twin Volume Fraction Analysis

In experiments, twin fraction is estimated by using EBSD orientation mapping[61]. Typically, EBSD scans have to be performed at higher magnification in order to identify smaller twins and large number of scans have to be generated in order to estimate the twin volume fraction with a satisfying statistical confidence. Figure 5.7 shows twins observed in shock recovered tantalum monocrystalline (a) and quasi-statically compressed nanocrystalline(b) samples using EBSD scans by Florando *et al.* [61] and [69] respectively.



(a) (b)

Figure 5.7: Twins observed in tantalum monocrystalline(a) and tantalum nanocrystalline (b) samples using EBSD analysis. Figure (a) and (b) are adapted from references [61] and [69] respectively.

5.5.1 Polyhedral Template Matching

Larsen *et. al* [85] proposed a new method called Polyhedral Template Matching (PTM) for reliably identifying local crystalline structure of materials in solid phase. This method utilize topology of neighborhood around an atom to determine likelihood of the atom belonging to a certain crystalline structure out of set of candidates. As a result, PTM predictions are less sensitive to strain and thermal vibrations of the atoms than other methods using interatomic distances for classification. Thus PTM classifications are more reliable than common neighbor analysis even at high temperatures. Further, this method is capable of estimating the local orientation or local elastic strain without information of original atomic configuration.

PTM calculates the degree of point-to-point correspondences between an atom's neighborhood and candidate template structures using root-mean-square deviation (RMSD) cal-

culation given in equation 5.4.

$$\mathbf{RMSD(v,\ w)} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \|\vec{v_i} - \vec{w_i}\|^2} \tag{5.4}$$

Where, **v** and **w** are local atomic configuration and the candidate template respectively.

If the optimal rotation and optimal scaling of **v** that minimize the RMSD value is given by s and **Q** respectively;

$$\mathbf{RMSD(v,\ w)} = min(s, \mathbf{Q})\sqrt{\frac{1}{N}\sum_{i=1}^{N} \|s[\vec{v_i} - \vec{\mathbf{v}}] - (\mathbf{Q}[\vec{w_i} - \vec{\mathbf{w}}]^T)^T\|^2} \tag{5.5}$$

Where, $\vec{\mathbf{v}} = \frac{1}{N}\sum_{i=1}^{N} \vec{v_i}$ and $\vec{\mathbf{w}} = \frac{1}{N}\sum_{i=1}^{N} \vec{w_i}$

Theobald *et al.* [136] provided a comprehensive description of finding the rotational matrix. and Horn *et al.* [137] explained a method for finding the optimal scale and showed **Q** is independent of the s.

An scale-invariant RMSD can be obtained by forcing mean distance of each point in the template to become 1. Thus:

$$\mathbf{RMSD(v,\ w)} = min(s, \mathbf{Q})\sqrt{\frac{1}{N}\sum_{i=1}^{N} \|s[\vec{v_i} - \vec{\mathbf{v}}] - \frac{1}{l(\mathbf{w})}(\mathbf{Q}[\vec{w_i} - \vec{\mathbf{w}}]^T)^T\|^2} \tag{5.6}$$

Where, $\frac{1}{l(\mathbf{w})} = \frac{1}{N}\sum_{i-1}^{N} \|\vec{w_i} - \vec{\mathbf{w}}\|$.

Number of neighboring atoms required to determine the RMSD value depends on the candidate structure. Atoms in first neighbor shell are sufficient to determine whether the local structure is simple cubic, fcc, hcp, or icosahedral. However, for bcc structure, atoms in first two neighbor shells are needed. Identifying atoms in correct neighbor shell can be

tricky at high temperature. To counter this, PTM orders neighbor atoms by the area of the bonding interface polygons of the Voronoi cell of the central atom and assign them in to correct neighbor shell. Thus, atoms with highers interface area are the atoms belong to the first neighbor shell. Faces of the convex hull could be triangular, perfect square, or rhombus. Perfect square facet and rhombus facet can have two different triangulations. Thus, to preserve the accuracy of predictions, all the possible triangulation of convex hull is considered and tri-connected planar graphs generated for each of them. Then, graph isomorphism is used on planar graphs of the convex hull and the same of the candidate structures to predict possible matches.

Twin volume fractions (TVF) of the strained crystals were computed using the Polyhedral Template Matching (PTM) method[85] which identifies the local crystalline structure of simple condensed phases, as well as local orientation of atoms. Twin-volume fractions were evaluated as a function of time using a root-mean-squared-displacement (RMSD) cut-off of 0.1.

Chapter 6

Strain Rate Dependence of Deformation-Twinning in Ta

Conditions for twinning and its interaction with dislocations in bcc materials subjected to high strain rates of deformation and pressures, such as those achieved in laser-induced shock loading, remain for the most part unexplored experimentally and computationally. While it is widely accepted that twins nucleate as a stress release mechanism, the role dislocation densities play in twin nucleation is not well understood. Recent uniaxial deformation experiments on tantalum samples cold-rolled to increase dislocation densities, have shown that twinning is suppressed with increase in dislocation density/activity [69]. We have performed a systematic computational study of twin nucleation in tantalum, under shock loading conditions, employing large scale non-equilibrium molecular dynamics (NEMD) simulations aim at quantifying the effect of strain, strain rate and initial dislocation density on twin nucleation in tantalum samples nominally oriented along (100) with preexisting defects (primarily screw dislocations with densities in the $10^{11} - 10^{12}$ cm^{-2} range). The quasi-isentropic compression (QIC) and expansion (QIE) simulations sample strain rates in the $10^{8} - 10^{10} s^{-1}$ range.

6.1 Methodology

Defective tantalum samples with three different initial dislocation densities were created for investigating the effect of existing dislocations on twinning. These samples approximately contained $a/2$ <111> dislocation densities of $2.5 \times 10^{12} cm^{-2}$, $2.5 \times 10^{11} cm^{-2}$,

$5.0 \times 10^{10} cm^{-2}$. Hereafter, these samples will be referred as HDD (higher dislocation density), MDD (medium dislocation density), LDD (lower dislocation density) attributing to their dislocation density. HDD and MDD samples contained approximately 4 million atoms while LDD sample contained approximately 34 million atoms. All the samples were roughly cubical in shape. Dimensions of HDD and MDD samples were 40nm×40nm×42nm, while the same of LDD was 82nm×82nm×88nm.

6.1.1 Defective Sample Preparation

Perfect tantalum sample were generated by replicating a bcc unit cell with a lattice parameter of 3.304nm. MDD and HDD samples contained 125 unit cells in x and y directions, and 135 unit cells in z direction. LDD sample contained 250 unit cells in x and y directions, and 270 unit cells in z direction. Dislocations were later introduced in to these samples by performing series of non-equilibrium and equilibrium MD simulations.

LDD and MDD samples were prepared following a similar procedure. First, the samples were introduced with a linear defect along *y-axis* by removing a single plane of atoms in the center as seen in figure 6.1 (a) and (d). Length of the linear defect was approximately 10% and 20% of the length of the sample in y-direction for LDD and MDD respectively. The samples were uniaxially compressed to a final strain of 10% by applying a strain rate in the order of $10^9 s^{-1}$. During the initial compression a $a/2$ <111> dislocation loop was emitted by the linear defect. Images of the dislocation loop are shown in figure 6.1 (b) and (e) for LDD and MDD respectively. The samples were uniaxial expanded back to the original volume at a strain rate in the order of $10^9 s{-}1$. The growth of the dislocation loop was slowed down by the uniaxial expansion of the samples and eventually became stationary. The uniaxial expansion also brought the samples back to zero pressure (figure 6.1 (c) and (f)). Finally the samples were thermalized at 300K.

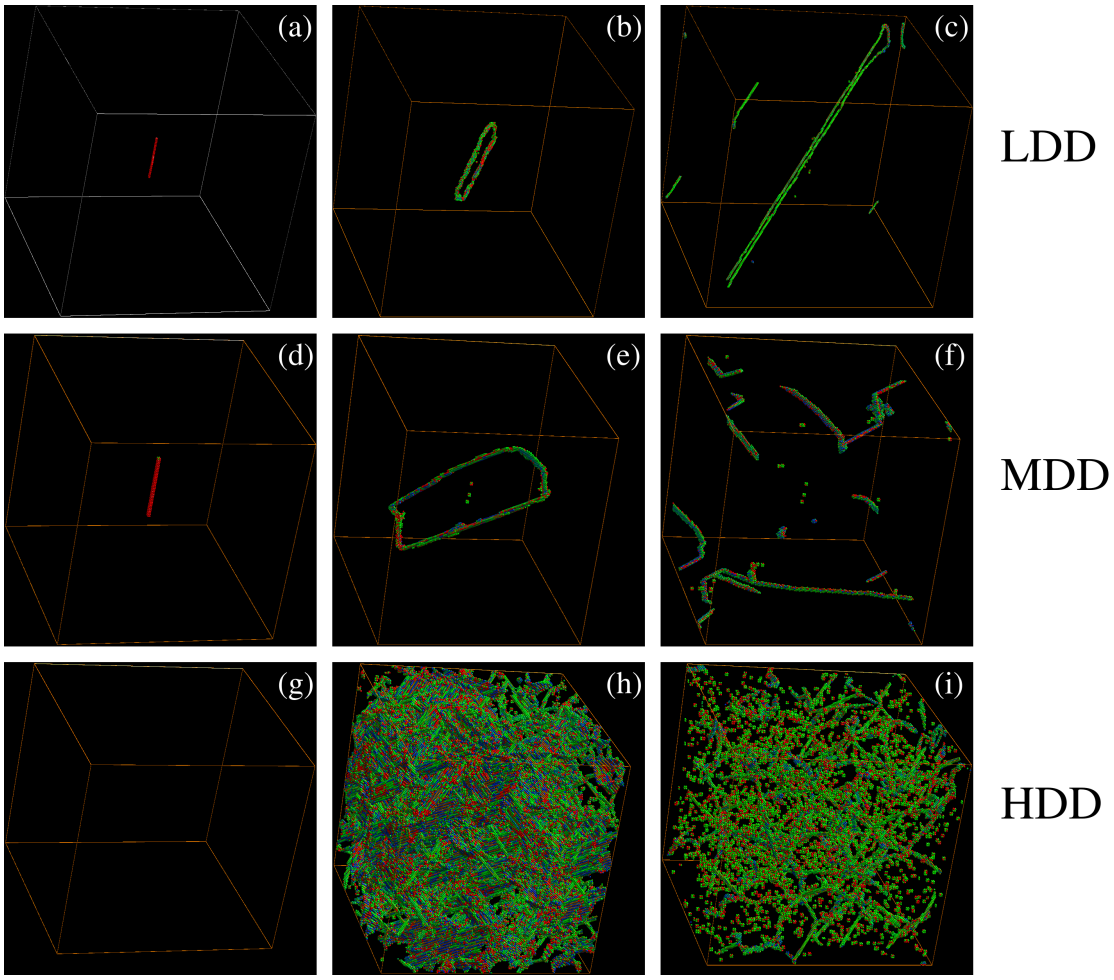


Figure 6.1: Snapshots taken from different stage of defective sample preparation. First, second, and third rows represent LDD, MDD, and HDD samples respectively while columns from left to right represent snapshots taken at initial stage, end of the uniaxial compression and after samples were brought back to zero pressure. These images show only the defective atoms filtered by central symmetry parameter

HDD sample was prepared by performing series of equilibrium and non-equilibrium simulations. First, a perfect tantalum sample (figure 6.1(g)) was uniaxially compressed to a final strain of 14% by applying a strain rate in the order of $10^{10}s^{-1}$. The sample was plastically deformed by the end of the compression as seen in figure 6.1(h). Growth of newly formed defects structures and dislocations glide caused a rapid increase in tempera-

ture. Then, the sample was anneal for 200ps to allow the sample to achieve it's equilibrium temperature . The sample was brought back to zero pressure by performing an isotropic expansion followed by an NVE simulation for 100ps. After this step a network of dislocations was present in the sample as seen in figure 6.1(i). High temperature NVE was performed for promoting dislocations glide which lowered residual shear stress of the sample. Finally, the sample was set back to 300K and annealed until temperature is equilibrated.

Figure 6.2 (a), (b) and (c) show the distribution of dislocations of LDD, MDD and HDD samples respectively. Blue colored line segments represents edge type dislocations while red colored segments represents screw type dislocations.

6.1.2 Simulation details

Atomistic simulations of quasi-isentropic compression (QIC) and expansion (QIE) simulations were carried out on the defective tantalum samples using the massively parallel MD code SPaSM. The atomic interactions were modeled using a recently developed EAM potential of tantalum (Ta2) [22].

Sampled final strains were ranging from 8% to 25%, as a result, the sampled pressures ranged from 19 GPa to 104 GPa. Applied strain rates were in the order of $10^8 s^{-1}$ to $10^{11} s^{-1}$. These strain rates are similar to the same achieved by laser shock experiments conducted in the Janus Laser Facility at Lawrence Livermore National Laboratory and the Omega Laser Facility at the University of Rochester in New York [71].

QIE simulations were performed on HDD sample that were pre-strained up to 20% and 25%. Thus, the initial samples were at 70GPa and 104GPa. The pre-compression was performed at lower strain rates ($\sim 1.0 \times 10^9 s^{-1}$) which guaranteed dislocation glide was solely capable of mitigating accumulated shear without nucleation of twins. Dislocation

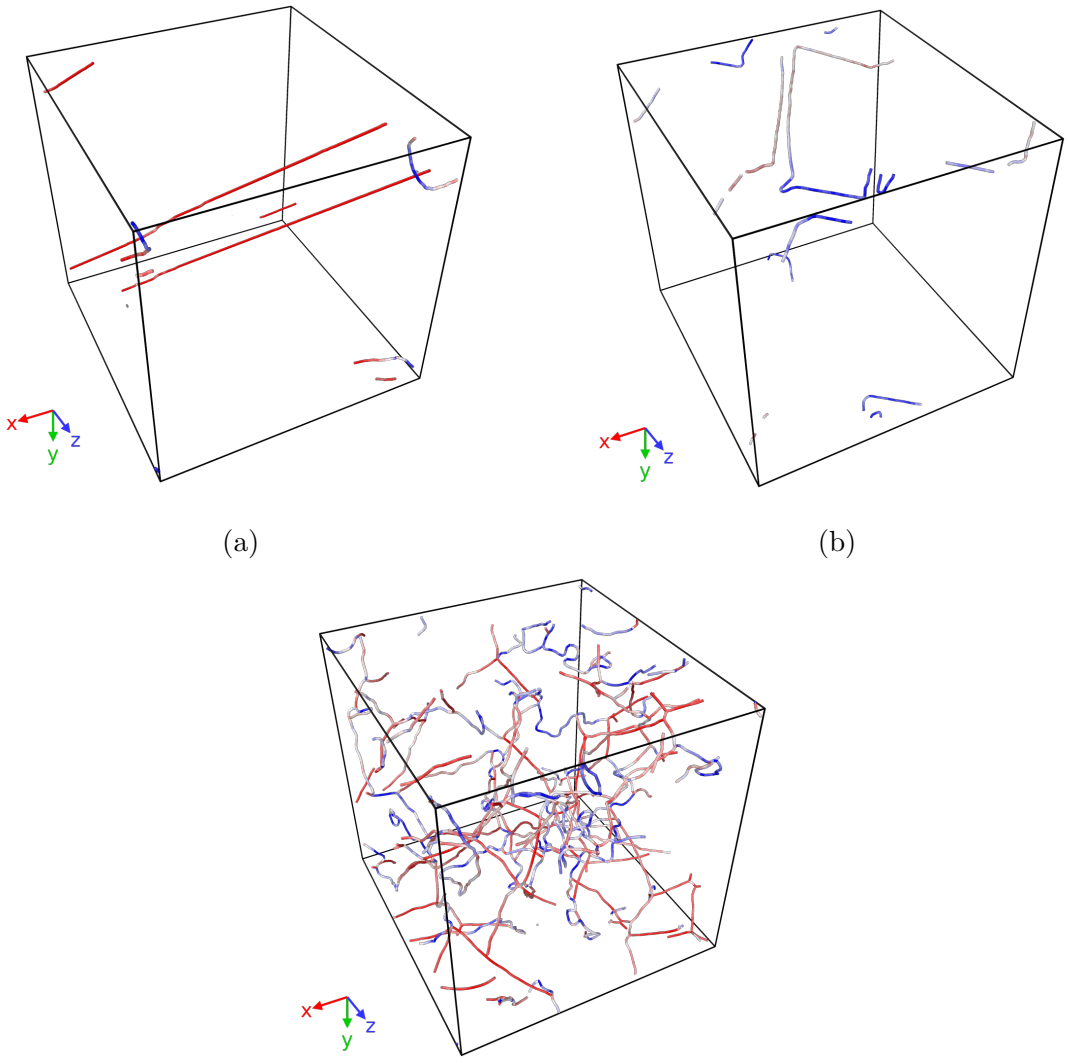
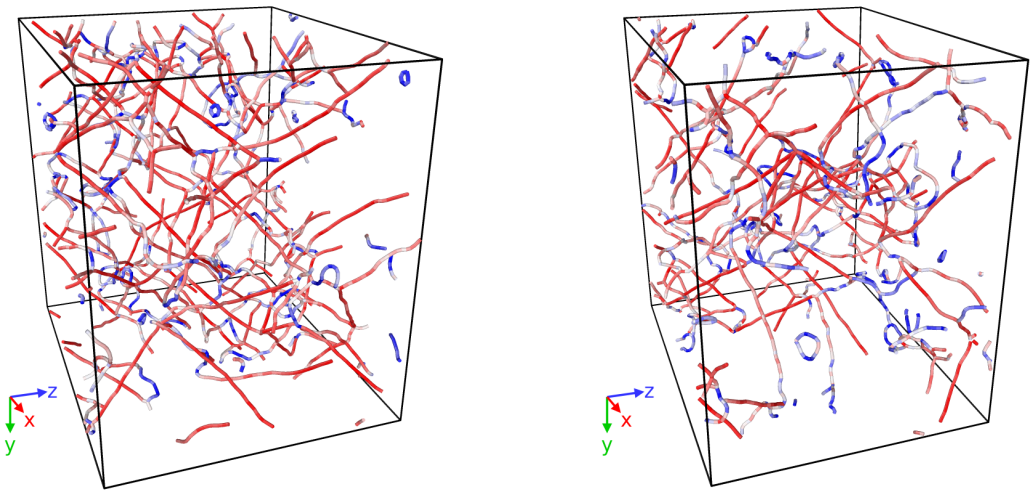


Figure 6.2: Initial dislocation network in LDD (a), MDD (b), and HDD (C).

density of pre-strained samples were $\sim 3.0 \times 10^{12} cm^{-2}$. Dislocation network of 20% and 25% pre-strained samples are shown in figure 6.3 (a) and (b) respectively.

Twinning under QIE was investigated at strain rated ranging from $10^{9} s^{-1}$ to $10^{11} s^{-1}$. The applied tension on each sample was calculated in a manner that post-expansion pressure is 0GPa. Following calculation demonstrates how tension(ϵ_{exp}) can be obtained for a



(a) (b)

Figure 6.3: Dislocation network in HDD samples pre-strained up to 20% (a) and 25% (b). QIE simulations were performed on these two samples under tensions of 25% and 33%.

given pre-strain (ϵ_{com}).

Pre-strain:

$$\epsilon_{com} = \frac{L_f - L_0}{L_0} \tag{6.1}$$

Where L_0 and L_f are initial and final length of the sample in the direction of compression.

$$\frac{L_f}{L_0} = 1 + \epsilon_{com} \tag{6.2}$$

Tension that required to bring the sample back to original volume:

$$\epsilon_{exp} = \frac{L_0 - L_f}{L_f} \tag{6.3}$$

By substituting for L_0/L_f from equation 6.2:

$$\epsilon_{exp} = \frac{-\epsilon_{com}}{1 + \epsilon_{com}} \tag{6.4}$$

Once QI compression or expansion is over, the sample is simulated by a microcanonical ensemble (NVE). Periodic boundary conditions were applied in all three directions. All

three defective samples were thermalized to a temperature of 300K and an initial pressure of $< 5kB$.

6.2 Results

Large-scale QIC and QIE simulations were performed on defective tantalum samples using the massively parallel MD code SPaSM. Simulated samples were analyzed for identifying dislocation structures, types and density using Crystal Analysis Tool (CAT) and DXA implementation on OVITO [135]. Analysis of deformation twin structures and it's relative volume were also performed by combining PTM and CNA methods for understanding strain rate dependence of twinning in tantalum at strain rates comparable to laser shock experiments.

Through out this chapter, the time at which non-equilibrium portion of the simulation ended will be referred as t_{rise} and the time at which the simulation was stopped will be referred as t_{final}. The compression and expansion simulations were performed under constant engineering strain rates. Thus, value of average strain rate is given when referring to the strain rate.

After the QI compression, the final pressure of defective samples were approximately independent of the applied strain rates, but solely a function of the final strain. Thus, final pressure of the compressed samples ranged from 18GPa to 104GPa.

Figure 6.4 shows twin structures in MDD sample compressed at a strain rate of $7.0 \times 10^8 s^{-1}$ along with a inset which is zoomed on to a twin structure with a (121) twin plane. The inset was generated by a slicing the defective crystal and showing atoms no deeper than 2.5 angstroms in x-direction. In this figure, un-twinned atoms which preserved original

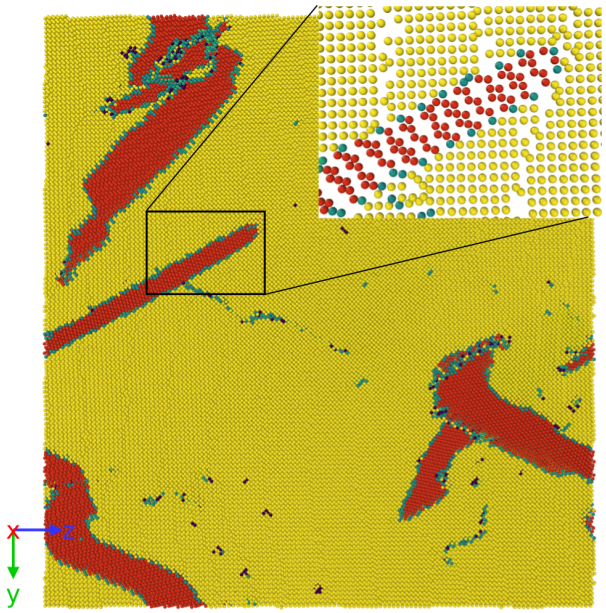


Figure 6.4: Twin structures in MDD sample compressed up to a strain of 0.14 at a strain rate of $7.0 \times 10^8 s^{-1}$. The inset is a closer look of a twin with {121} twin plane.

crystallographic orientation are colored in yellow, atoms in the twinned region are colored in red, and atoms in the twin boundary are colored in green.

During the compression, the density of $^a/_2$ <111> dislocations increased due to dislocation multiplication. However, at the nucleation of the twins, $^a/_2$ <111> dislocation density started to decrease. This could be due to DXA's inability to identify partial dislocations in the twin boundaries. Figure 6.5 shows $^a/_2$ <111> dislocation density profile for HDD sample compressed to a final strain of 25% at a strain rate of $3.3 \times 10^9 s^{-1}$.

QIC equations of motion assures that temperature would increase during the compression. figure 6.6 shows temperature (a) and shear stress (b) time profiles from QIC simulations performed on HDD sample at different strain rates up to a final strain of 20%. The temperature kept rising even after the compression was stopped (at t_{rise}) due to the plastic work done by the mobile dislocations and some of the twin structures created during

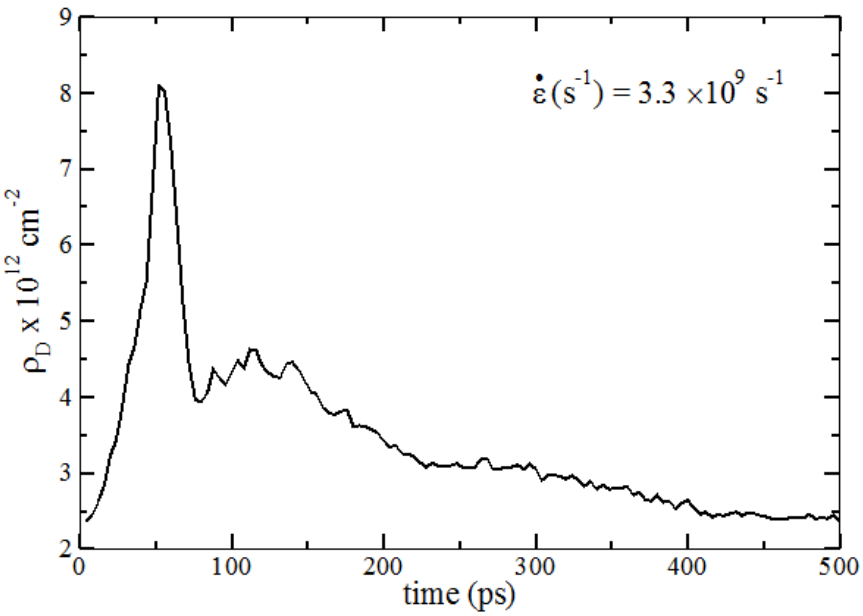
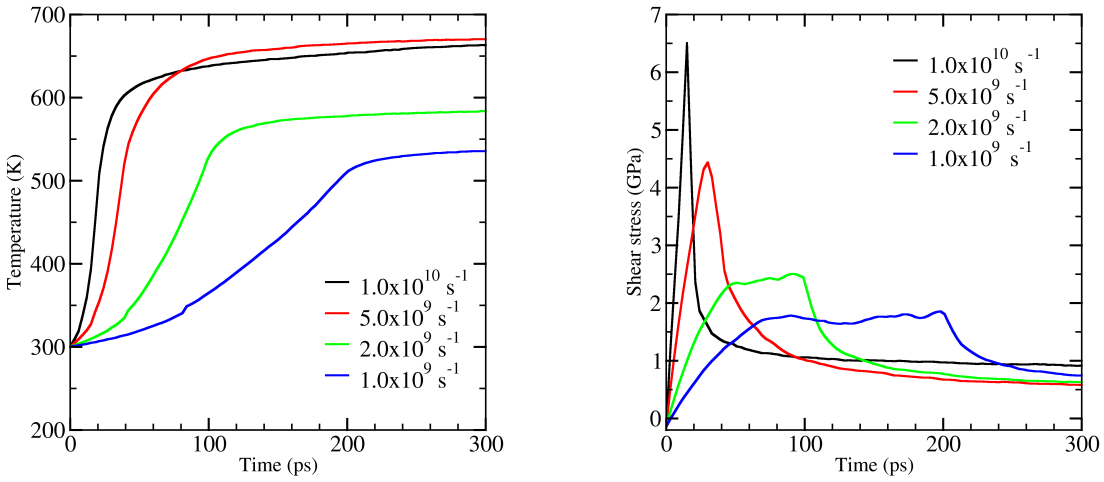


Figure 6.5: Time profile of $^{a}/_{2}$ <111> dislocations density obtained from a QIC simulation performed on HDD sample at strain-rate of $3.3 \times 10^{9} s^{-1}$.

the compression is being annealed out. Higher strain rates caused the final temperature to be higher. However, final temperatures for the simulations performed at a strain rate of $1.0 \times 10^{10} s^{-1}$ and $5.0 \times 10^{9} s^{-1}$ are approximately similar. This can be account to the different amount of twin structures are being annealed out during this time period. Twin volume fraction (TVF), and dislocation density at the end of the compression as well as at the end of the simulation is given in the table 6.1. At higher strain rates shear stress profiles exhibit a clear maxima as seen in figure 6.6 (b). Twin nucleation started at or very close these points.

LDD and MDD samples also exhibit similar behaviors in temperature and shear stress time profiles. Figure 6.7 shows a sequence of snapshots from a simulation of an MDD



(a) (b)

Figure 6.6: Temperature (a) and shear stress (b) time profiles for HDD samples compressed to a final strain of 20% at different strain rates.

Table 6.1: TVF, dislocation density (DD) and temperature at t_{rise} and t_{final} for 20% compression of HDD sample at three different strain rates.

	TVF (%)		DD ($10^{12} cm^{-2}$)		Temperature (K)		
Strain Rate	t_{rise}	t_{final}	t_{rise}	t_{final}	t_{rise}	t_{final}	$t_{final} - t_{rise}$
$1.0 \times 10^9 s^{-1}$	0.1	0.02	4.8	3.0	507.0	534.0	27.0
$2.0 \times 10^9 s^{-1}$	0.4	0.03	7.5	2.9	524.0	582.0	58.0
$5.0 \times 10^9 s^{-1}$	11.6	0.04	4.2	3.5	526.0	669.0	143.0
$1.0 \times 10^{10} s^{-1}$	24.2	25.5	3.9	6.0	476.0	663.0	187.0

sample compressed to a final pressure of 39 GPa at a strain rate of 7.0×10^8 s^{-1}. Only defective atoms (non centrosymmetric atoms) are shown in frames (a), (b) and (c). Figure 6.7 (d) shows the time evolution of the global shear stress in which the labels (a), (b) and (c) indicate the time at which the corresponding snapshots were taken. The maximum shear stress and yield of the sample can be identified as the time at which twin nucleation

occurs. The MDD samples yield at strains between 11%- 12.5% while perfect, defect-free Ta crystals compressed along the (100) direction, yield at strains between 13.5%-14% regardless of the applied strain rate (strain-rate independent).

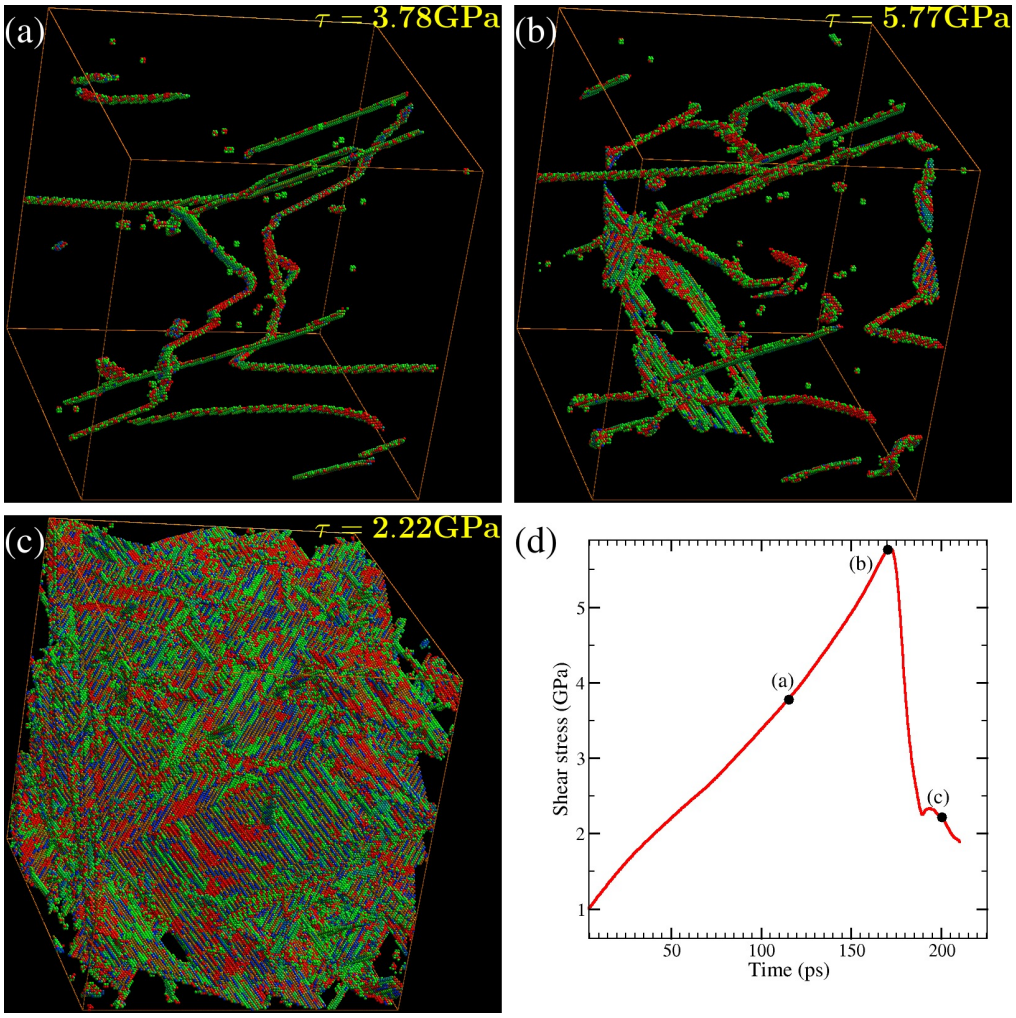
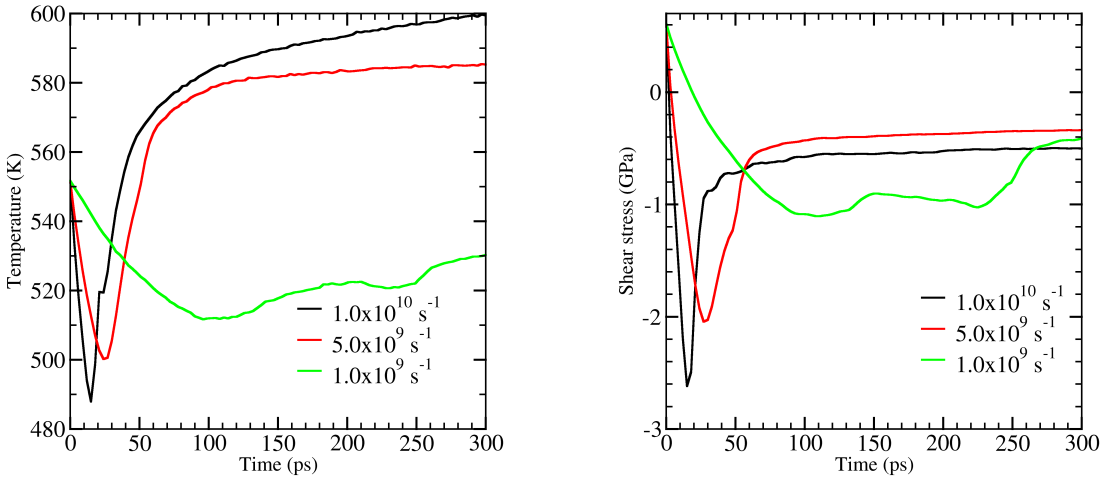


Figure 6.7: Time evolution of twin nucleation in an MDD sample (initial $\rho_D \approx 2.5 \times 10^{11}$ cm^{-2}) compressed 14% at a strain rate of 10^9 s^{-1}. showing defective atoms. Snapshot (a) was taken before twin nucleation event; (b) at about the start of twin-nucleation and (c) long after; (d): time evolution of shear stress in which the labels (a), (b) and (c) correspond to the snapshots of the defects shown.

Figure 6.8 shows the temperature (a) and shear stress (b) time profiles from QIE simulations performed on pre-compressed HDD sample at different rates of expansion. The sample was expanded 25% in z direction, thus volume of the sample at the end of the expansion is equal to the same of HDD sample.



(a) (b)

Figure 6.8: Temperature (a) and shear stress (b) time profiles for HDD samples expanded to a final strain of 25% at different strain rates.

During the expansion the temperature was decreased up to a certain value that depend on the applied strain rate before starting to increase. Rapid increment in temperature was observed under higher strain rates as soon as deformation twins start nucleating. This trend in temperature did not hold for lower strain rates where no twins were observed. Instead, the temperature started increasing slowly as seen in green line in figure 6.8 (a). A summary of twin volume fraction, and dislocation density, and temperature at the end of the expansion as well as at the end of the simulation is given in the table 6.2. Under tension, shear stress drops below zero meaning that P_{zz} of the stress tensor is smaller than the average of P_{xx} and P_{yy}.

Table 6.2: TVF, dislocation density (DD) and temperature at t_{rise} and t_{final} for 25% expansion of pre-compressed HDD sample at three different strain rates.

	TVF (%)		DD ($10^{12} cm^{-2}$)		Temperature (K)		
Strain Rate	t_{rise}	t_{final}	t_{rise}	t_{final}	t_{rise}	t_{final}	$t_{final} - t_{rise}$
$1.0 \times 10^{9} s^{-1}$	0.0	0.0	2.8	2.5	521.0	530.0	9.0
$5.0 \times 10^{9} s^{-1}$	0.2	0.0	6.5	4.0	547.0	585.0	38.0
$1.0 \times 10^{10} s^{-1}$	14.3	6.2	6.2	5.0	520.0	600.0	80.0

As seen in figure 6.8 (b) Shear stress started rapidly increasing resulting a sharp minima for higher strain rate simulations once the twins started nucleating, similar to what was observed during the compression simulations.

Figure 6.9 shows shear stress profiles as a function of instantaneous strain (the total strain the sample is experiencing at the moment shear stress was measured) for LDD (green line), MDD (red line), HDD (blue lines) and defect-free sample (dash-dot line) compressed up to a final strain of 14%. A defect-free crystal which obatined the highest initial yielding point of 9.13 GPa was compressed at a strain rate of $7.0 \times 10^{8} s^{-1}$.

6.2.1 Twin Volume Fraction Analysis

Twin volume fraction(TVF) analysis was performed using PTM method on LDD, MDD, and HDD Tantalum samples at different stages through out the simulation. OIM and implementation of PTM in OVITO were used for visualizing twin structures. Deformation twins nucleated under low strain simulation were appeared to be coherent, but under high compressive strains irregular twin boundaries were observed.

In this study twins were observed in the LDD sample that had a dislocation density of

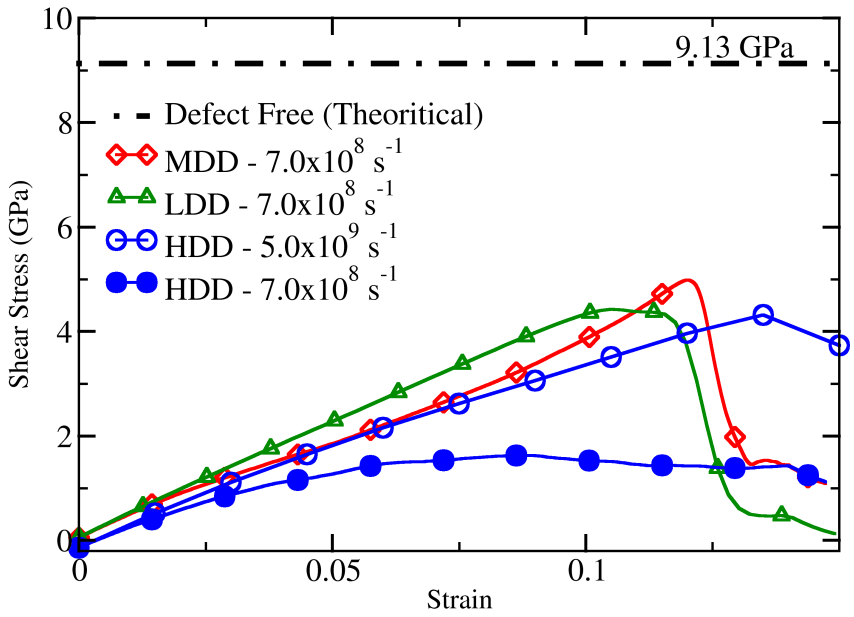


Figure 6.9: Shear stress as a function of instantaneous stain for LDD, MDD, HDD and defect free samples compressed up to 14% of strain.

$5.0 \times 10^{10} cm^{-2}$ for every strain rate sampled at final strains $\geq 10\%$. From the two simulations where applied final strain was 8% twins were not observed at strain rate of $7.0 \times 10^{8} s^{-1}$ but at strain rate of $5.0 \times 10^{9} s^{-1}$ a small volume fraction of twins were observed.

Simulations performed with MDD sample generated deformation twins for all the strain rates sampled where the applied final strain was $\geq 10\%$. In HDD sample, twins was observed in simulations where applied strain rate is higher, suggesting existence of a threshold in strain rate for twinning or, an existence of a threshold in shear stress for twinning.

Figure 6.10 shows TVF profiles for MDD samples compressed at different strain rates in the range of $10^{8} - 10^{9} s^{-1}$ up to a final strain of 14%. X-axis of this figure represents

the instantaneous strain. Once, the stain reach to 14% the sample is simulated with NVE equations of motion allowing the sample to naturally achieve correct pressure and temperature.

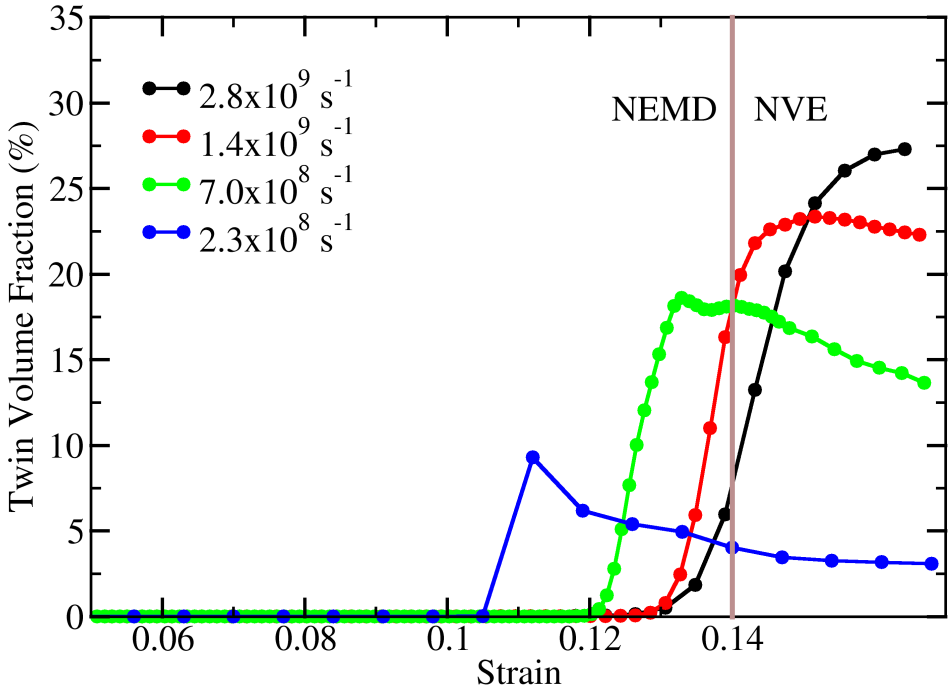


Figure 6.10: TVF profiles from MDD sample compressed up to a strain of 14% at different strain rates. The x-axis is sample strain at the moment twin volume fraction is calculated.

A series of snapshots taken in closed time intervals from a QIC simulation of MDD at a strain rate of $1.4 \times 10^9 s^{-1}$ are shown in figure 6.11. The images in the top row show twin structures in green and untwinned regions in yellow, while the bottom row of images show only dislocations (in green) and twin boundaries (in gray). Images in the same column are obtained from the same saved atom configuration file. In these snapshots, (a), (b) and (c) represents snapshots taken, before, at, and after the nuclations twins while (d) represents

a snapshot taken about 30 picoseconds after the nucleation of twins. A summary of TVF and dislocation densities are given in table 6.3.

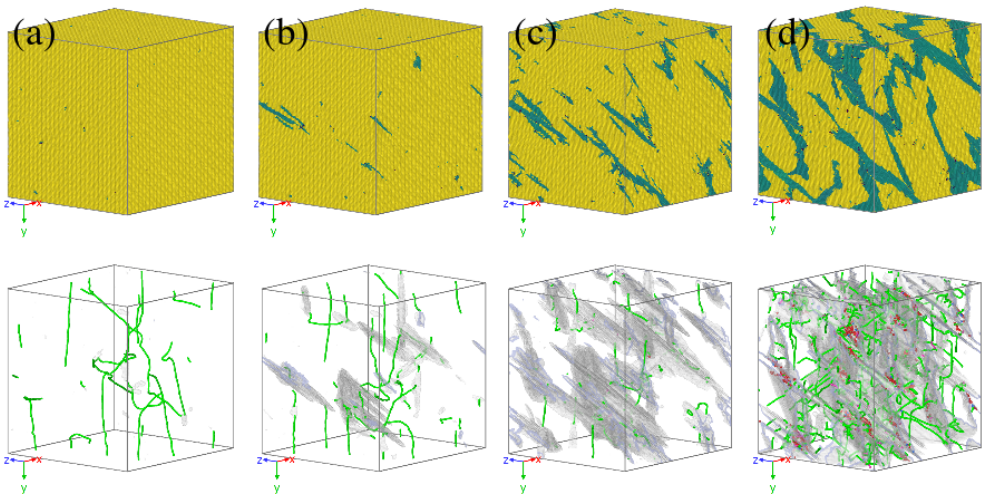


Figure 6.11: Snapshots taken in close intervals from QIC simulation where MDD sample was compressed at strain rate of $1.4 \times 10^9 s^{-1}$. A summary of TVF and dislocation density gathered from these snapshots are given in table 6.3.

Table 6.3: Summary of TVF and dislocation densities of the snapshot shown in figure 6.11. Dislocation density values are given in 1.0×10^{12} cm^{-2}.

Image Label	Time (ps)	TVF (%)	DD_{final}
(a)	81.3	0.0	0.7
(b)	91.3	0.6	0.6
(c)	96.3	4.9	0.5
(d)	117.3	21.5	3.5

Effect of Strain

The effect of strain on TVF and dislocation density was analysed by comparing QIC simulations performed on HDD sample at various strain rates by applying final strains of 20% and 25%. Figure 6.12 shows defective atoms (top row) and OIM (bottom row) of snapshots taken at the end of the simulations. A summary of TVF and Dislocation density data of these simulations along with the strain rate is given in table 6.4.

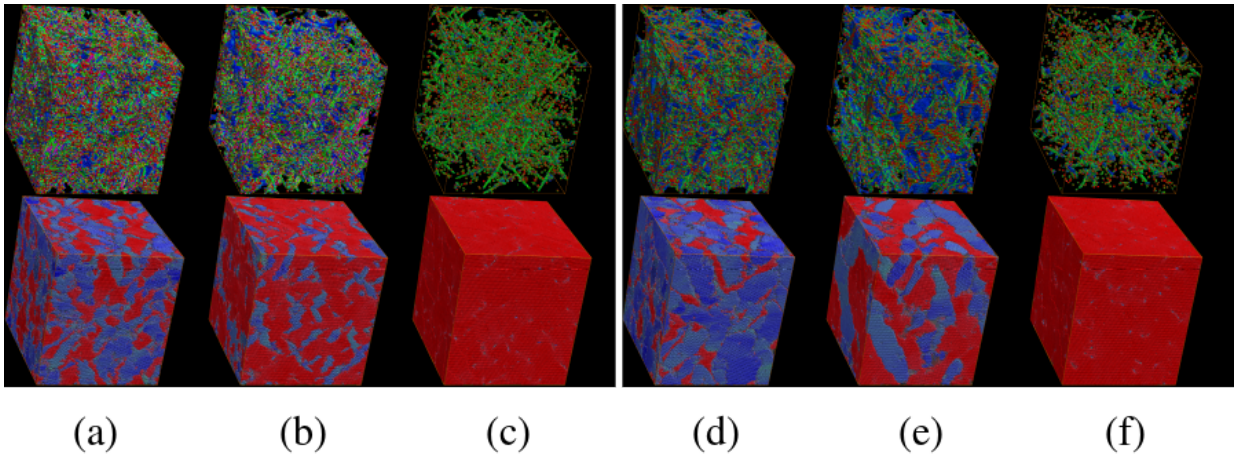


Figure 6.12: Snapshots taken at the end of the from QIC simulations performed on HDD sample by applying final strains of 20% ((a) -(c)) and 25% ((d) - (f)). Summary of TVF and dislocation density of these simulations are given in table 6.4

A similar analysis was also performed using QIE simulations by applying final expansions of 25% and 33% at various expansion rates. Figure 6.13 shows defective atoms (top row) and OIM (bottom row) of snapshots taken at the end of the simulations. A summary of TVF and Dislocation density data of these simulations along with the strain rate is given in table 6.5.

Table 6.4: Summary of TVF and dislocation analysis performed on snapshot shown in figure 6.12. Dislocation density values are given in 1.0×10^{12} cm^{-2}.

Image Label	Final Strain	Strain Rate (ns^{-1})	TVF_{final} (%)	DD_{final}
(a)	0.20	10.0	36.0	5.1
(b)	0.20	5.0	11.3	4.5
(c)	0.20	2.5	0.0	4.6
(d)	0.25	10.0	47.0	6.5
(e)	0.25	5.0	33.2	4.6
(f)	0.25	2.5	0.0	4.0

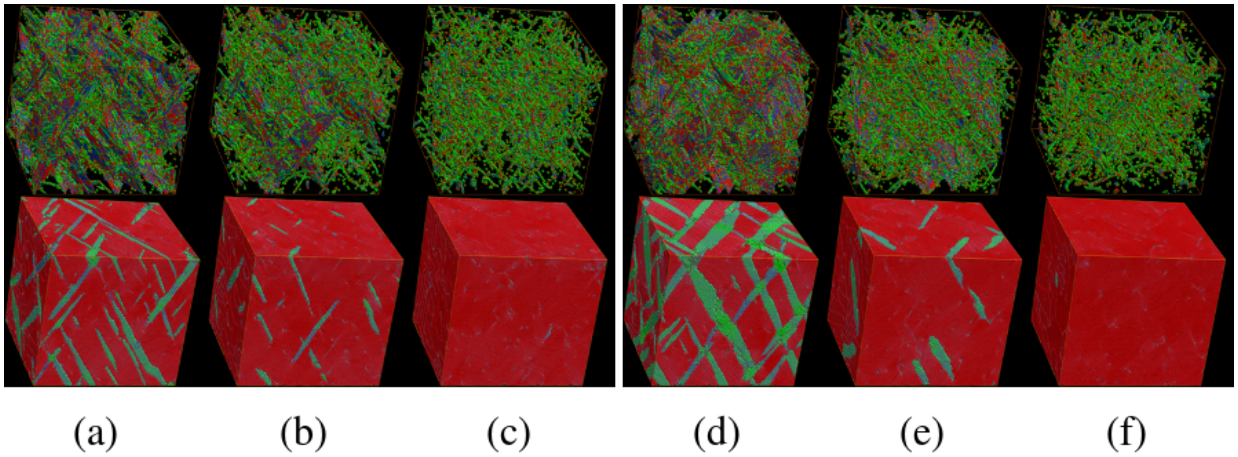


Figure 6.13: Snapshots taken at the end of the from QIE simulations performed on HDD sample by applying final strains of 25% ((a) -(c)) and 33% ((d) - (f)). Summary of TVF and dislocation density of these simulations are given in table 6.5

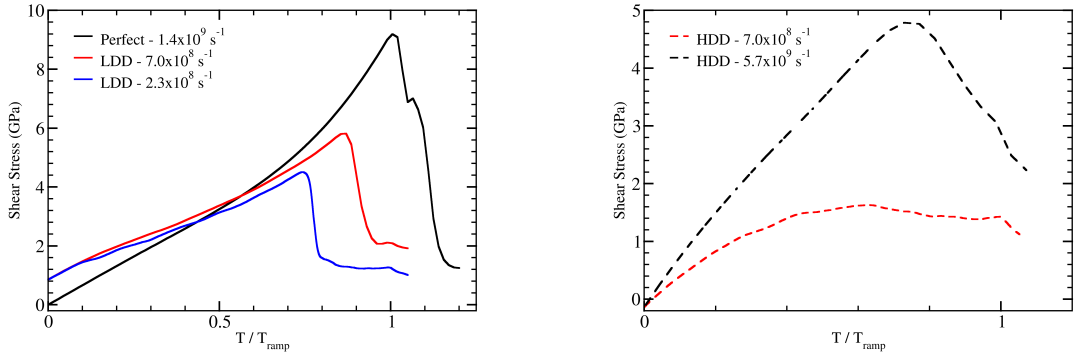
Effect of Initial Dislocation Density

Figure 6.14 (a) shows the time evolution of the global shear stress for defective MDD and HDD Ta samples as well as that of a compressed defect-free crystal. The time scale has been normalized by the loading time, t_{ramp}. Figure 6.14 (b) shows shear stress profiles for HDD Ta sample compressed to 104 GPa at two different strain rates. Twinning was not

Table 6.5: Summary of TVF and dislocation analysis performed on snapshot shown in figure 6.13. Dislocation density values are given in 1.0×10^{12} cm^{-2}.

Image Label	Final Expansion	Expansion Rate (ns^{-1})	TVF$_{\text{final}}$ (%)	DD$_{\text{final}}$
(a)	0.25	10.0	36.0	4.1
(b)	0.25	6.0	11.3	6.0
(c)	0.25	5.0	0.0	6.4
(d)	0.33	10.0	47.0	5.0
(e)	0.33	3.3	33.2	5.6
(f)	0.33	2.8	0.0	4.6

observed at strain rate of $7.0 \times 10^8 s^{-1}$ (red dashed line). Twin nucleation was observed in HDD samples (both under compression and expansion) only for strain rates $> 2.5-3.0 \times 10^9$ s^{-1}.



(a) (b)

Figure 6.14: Shear stress time profiles for perfect and MDD Ta samples compressed at 14% (a), and HDD Ta sample compressed 20% (black dashed line) and 14% (red dashed line) at different strain rates (b). Time scale has been normalized to compression time t_{ramp} for the sampled strain rates.

QIC simulation performed on LDD and MDD samples nucleated twins under lower final strains and lower strain rates compared to HDD. At low final strains, HDD sample required larger strain rates to produce twins. TVF in HDD under low strain, and high strain rates was small (usually $\leq$10%) and slowly annealed out and disappeared. Figure 6.15 shows OIM images of MDD ((a) - (c)) and HDD ((d) - (f)) samples compressed at strain rate of $5.0 \times 10^9 s^{-1}$ for final strains of 0.14, 0.20, and 0.25. Table 6.6 summarizes the TVF and dislocation analysis on the snapshots shown in Figure 6.15. Regions in red color represents atoms that has [100] orientation aligned with z-direction while atoms in blue color were identified as twins.

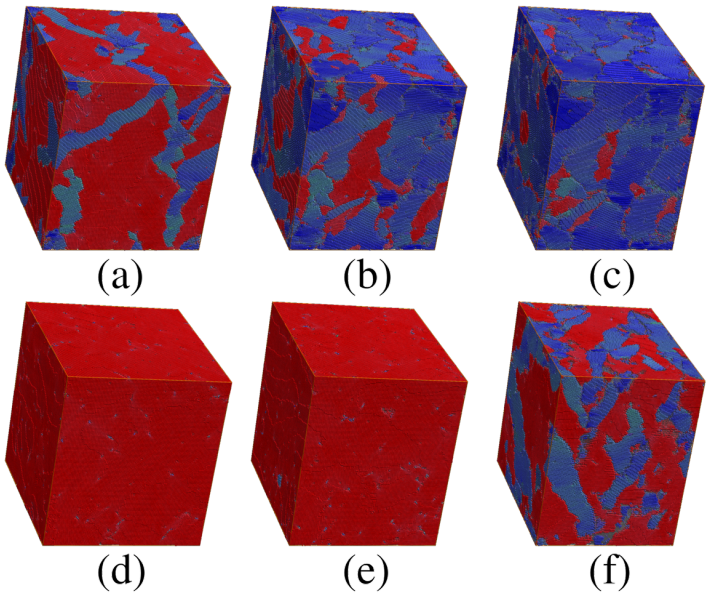


Figure 6.15: OIM images of MDD ((a) - (c)) and HDD ((d) - (f)) samples compressed at strain rate of $5.0 \times 10^9 s^{-1}$ for final strains of 0.14, 0.20, and 0.25. Table 6.6 summarizes the TVF and dislocation analysis on these snapshots.

Twinning in defective Ta crystals was observed under QI compression for strain rates above a shear stress threshold regardless of the initial dislocation density and final strain. Thus, if the rate of compression is higher enough to overcome relaxation caused by dislo-

Table 6.6: Summary of TVF and dislocation analysis performed on snapshot shown in Figure 6.15. Dislocation density values are given in 1.0×10^{12} cm^{-2}.

Image label	Sample	Final strain	$\text{TVF}_{\text{final}}$ (%)	DD_{final}
(a)	MDD	0.14	23.8	4.9
(b)	MDD	0.20	55.2	7.7
(c)	MDD	0.25	65.2	9.7
(d)	HDD	0.14	0.0	4.4
(e)	HDD	0.20	0.3	5.2
(f)	HDD	0.25	29.6	4.2

cation slip and build up shear stress above a certain threshold value the twins would be nucleated. However, since samples with higher initial dislocations are capable of relaxing shear stress better than the samples with lower initial dislocations, higher strain rates are required to achieve this shear stress threshold in samples with higher initial dislocation densities. We found this shear stress threshold to be in the range of 4.3GPa to 5.0GPa

Figure 6.16 shows the shear stress at which twin nucleation started as a function of sample dislocation density. The three data points were taken from the simulation twins were observed with the smallest possible strain rate.

Effect of Strain Rate

Figure 6.17 (a) shows the twin volume fraction (TVF) and maximum shear stress as a function of applied strain rate on LDD, MDD and HDD samples under QI compression. The final strain applied on HDD (black data points) was 25% while the same was 14% on LDD and MDD. At lower strain rates, samples with lower defect density samples nucleate more twins than the samples with higher dislocation densities as seen in Figure 6.17 (a).

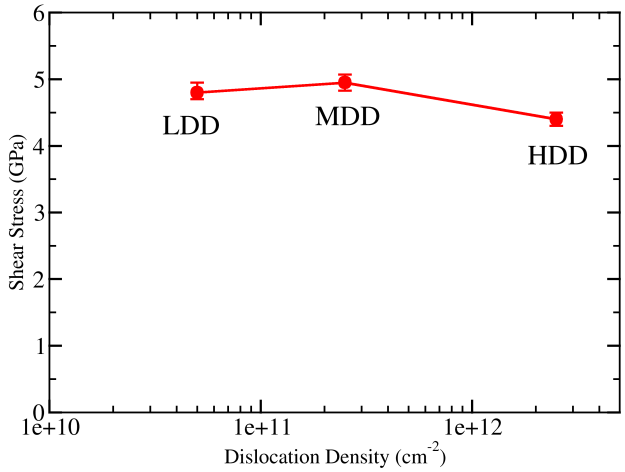


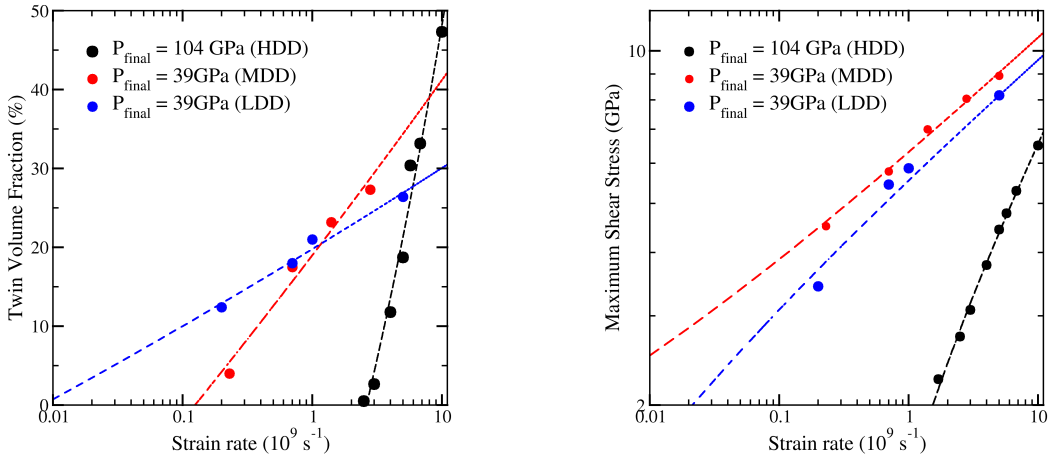
Figure 6.16: Shear stress at which twin nucleation started as a function of initial dislocation density of the samples.

The dashed straight lines are power law fits to the data.

Figure 6.17 (b) shows the maximum shear stress as a function of applied strain rate on LDD, MDD and HDD samples under QIC compression. Within the range of sampled strain rates, for a given strain rate, LDD and MDD exhibit higher yielding stress compared to HDD sample.

6.3 Summary

In this study, we have performed large scale MD simulations for studying twinning and it's interplay with dislocations in model bcc metal tantalum. Three multi-million atom samples with different $^{a}/_{2}$ <111> screw dislocation densities were created. Uniaxial quasi-isentropic compression and expansion simulations were performed on the defective samples by varying strain and strain rate. Table 6.7 contains a summary of few of the selected QIC



(a) (b)

Figure 6.17: (a): TVF as a function of strain rate for defective tantalum sample compressed uniaxially to a final strain of 25% (black line) and 14% (red and blue lines); (b): maximum shear stress as s function of strain rate for defective tantalum samples compressed uniaxially to a final strain of 25% (black) and 14% (red and blue lines) respectively.

simulations in order to present the domain of strain and strain rate sampled in this study. Twins volume fraction and dislocation densities were calculated by employing PTM [85] and DXA[38, 39] respectively.

We find that volume of deformation twin in tantalum is strongly dependent on the initial dislocation density as well as final strain (pressure) and strain rate, similar to the recent experimental work of Florando et al [69]. Deformation twins under expansion (where the final pressure was approximately 0GPa) and under lower strain compressions were coherent, however twins with incoherent boundaries were observed in all Ta samples compressed to pressures above 39 GPa.

The effect of applied strain, strain rate and initial dislocation density on strength and twinning of defective tantalum crystals are summarized below.

Table 6.7: Summary of few of the selected simulations for demonstrating the domain of the strains, and strain rates sampled on each of the defective Ta samples.

Sample	Final strain	Strain rate (ns^{-1})	Maximum shear stress (GPa)	TVF_{final} (%)	DD_{final} ($\times 10^{12} cm^{-2}$)
LDD	0.08	0.7	4.5	0.0	0.8
LDD	0.08	5.0	4.9	4.6	0.6
LDD	0.14	1.0	5.9	26.1	4.2
LDD	0.14	5.0	10.7	26.4	5.5
LDD	0.17	5.0	10.7	51.2	6.8
LDD	0.20	5.0	10.7	61.4	8.7
LDD	0.25	5.0	10.7	66.3	10.3
MDD	0.08	5.0	4.5	0.0	2.4
MDD	0.10	4.0	5.7	8.7	2.4
MDD	0.10	5.0	5.7	11.3	2.2
MDD	0.14	0.7	5.8	10.5	3.5
MDD	0.14	5.0	8.9	23.8	4.8
MDD	0.17	5.0	8.9	41.3	6.0
MDD	0.20	5.0	8.9	55.2	7.7
MDD	0.25	5.0	8.9	65.2	9.7
HDD	0.08	5.0	4.3	0.0	2.6
HDD	0.14	5.0	4.3	0.0	4.4
HDD	0.14	50.0	8.1	0.0	3.9
HDD	0.20	5.0	4.4	0.3	5.2
HDD	0.20	50.0	12.5	48.3	5.6
HDD	0.25	5.0	4.5	29.6	4.2
HDD	0.25	50.0	12.5	43.8	6.8

6.3.1 Effect of strain

- At lower final strains, the twinning threshold in strain rate was higher. In HDD sample twinning threshold in strain rate for a compression of 20% is extrapolated to be close to $3.5 \times 10^9 s^{-1}$ while the same is close to $2.5 \times 10^9 s^{-1}$ for a compression of 25%.

- At higher final strains, higher twin volume fractions were observed.

- Higher final strains caused dislocation density at the end of the compression to be higher.

6.3.2 Effect of initial dislocation density

- The twinning threshold in strain rate was lower for samples with lower initial dislocation densities. LDD and MDD nucleates twins at lower strain rates and lower strains than HDD. Twinning threshold for LDD and MDD was extrapolated to be $\sim 1.0 \times 10^7 s^{-1}$ and $1.0 \times 10^8 s^{-1}$ respectively at a final strain of 14% while the same was $2.5 \times 10^9 s^{-1}$ for HDD sample at a final strain of 25%.

- The sample with lower dislocation density yields at higher shear stress than the sample with higher initial dislocation density.

- For a given strain and strain rate, the defect free sample has the highest yielding point compared to all of the defective samples.

6.3.3 Effect of strain rate

- Higher strain rates tend to impose a higher initial yield point on defective samples.

- Lower strain rates do not show an upper yield point, instead shear stress profiles are plateaued. No twins were observed for such cases.

Chapter 7

Summary

We have used CoMD, a proxy application for classical MD, to investigate load balancing of shock wave simulations as well as improvements of force evaluation kernels in GPUs by implementing neighbolists.

In shock wave simulations, atomic density varies significantly within the sample. Thus, when using a Eulerian type domain decomposition would cause processors which are assigned highly dense regions of the sample to perform more operations compared to processors which are assigned regions with lower atomic density. Thus, execution time of a timestep will be slower due to inherent synchronization between time steps in MD codes.

CoMD-OpenCL was used for studying performance improvement of EAM force evaluation kernels by implementing neighborlist. A buffer was added to cutoff of the potential in order to mitigate frequent updates of the neighborlist. For a given buffer size, at higher temperatures, the frequency of nighborlist updates was higher which decreases performance of the code. This effect of high temperature can be mitigated by increasing the buffer size. Speedup of two times was achieved by using buffer size of 20% of cut off distance at 600K.

Under substantial amount of uniaxial deformation depending on the orientation, the perfect fcc and bcc lattice becomes face centered tetragonal (fct) and body centered tetragonal (bct). However, the CNA or PTM methods that we employed in this study identify fct structures as fcc and bct structures as bcc.

In adaptive-CNA, the cutoff of neighbors is individually for each atom depending on the reference structure that is being compared with. In contrast, when using CNA, the user has to provide this cutoff distance. Depending on the provided cutoff distance some atoms can be identified differently. Since our method of identification of twin structures involve both CNA and PTM methods depending on the cutoff the value of TVF changed slightly. Thus we have used adaptive-CNA method when calculating TVF.

We also have implemented QI compression and expansion model in Los Alamos production MD code SPaSM by incorporating a deformation (either strain or tension) rate function in both positions and velocities equation of motions. This new formalism, change in internal energy is exactly equal to work done during the compression or expansion.

Large-scale NEMD and quasi-isentropic compression simulations of Ta crystals with different levels of pre-existing defects has been carried out examining the strain-rate dependence of plastic deformation, and strain rate dependence of twinning in the $10^8 - 10^{11} s^{-1}$ regime. The defective samples contained initial densities of $a/2$ <111> dislocation $\sim 5.0 \times 10^{10}$ cm^{-2} (LDD), $\sim 2.5 \times 10^{11}$ cm^{-2} (MDD), and $\sim 2.5 \times 10^{12}$ cm^{-2} (HDD).

The shear stress profiles contained a clear yielding point for all of the QI compression and expansion simulation where deformation twins were nucleated. Thus, twinning act as a mechanism for releasing shear stress rapidly. Contrarily, shear stress profiles did not show a clear yielding point in simulations where deformation twins were not observed which was common for HDD sample at lower strain rate.

Twins have been observed for QIC simulation where strain and the strain rate were higher enough to drive shear stress up to the range of 4.3GPa to 5.0GPa. From the QIC simulations where deformation twins were nucleated under compression, the minimum yielding point of LDD, MDD and HDD samples are 4.8GPa, 4.9GPa and 4.4GPa

respectively.

A threshold in strain rate for twin nucleation under compression was observed to be within the range of 2.5×10^9 s^{-1} - 3.0×10^9 s^{-1} for HDD sample for a final strain of 25%. The estimated twinning threshold of strain rate for LDD and MDD sample were $1.0 \times 10^7 s^{-1}$ and $1.0 \times 10^8 s^{-1}$ for a final strain of 14%. This threshold in strain rate is the bare minimum for driving the shear stress up to the range of minimum yielding points.

At high strain rates, the flow stress of all three defective tantalum samples follows a power law behavior in strain rate with exponents: ~ 0.45 for HDD and ~ 0.20 for LDD and MDD. Our results suggest the existence of a linear regime in the flow stress - strain rate domain with a crossover to a near power law regime at higher strain rates.

References

[1] J. Wadsworth, G. Crabtree, R. Hemley, R. Falcone, I. Robertson, J. Stringer, P. Tortorelli, G. Gray, M. Nicol, J. Lehr, *et al.*, "Basic research needs for materials under extreme environments. report of the basic energy sciences workshop on materials under extreme environments, june 11-13, 2007," Tech. Rep. (DOESC (USDOE Office of Science (SC)), 2008).

[2] M. Koenig, E. Henry, G. Huser, A. Benuzzi-Mounaix, B. Faral, E. Martinolli, S. Lepape, T. Vinci, D. Batani, M. Tomasini, *et al.*, Nuclear fusion **44**, p. S208 (2004).

[3] P. Celliers, G. Collins, D. Hicks, M. Koenig, E. Henry, A. Benuzzi-Mounaix, D. Batani, D. Bradley, L. Da Silva, R. Wallace, *et al.*, Physics of Plasmas **11**, L41–L44 (2004).

[4] H. Robey, Y. Zhou, A. Buckingham, P. Keiter, B. Remington, and R. Drake, Physics of Plasmas **10**, 614–622 (2003).

[5] A. Dewaele, P. Loubeyre, and M. Mezouar, Physical Review B **70**, p. 094112 (2004).

[6] A. D. Chijioke, W. Nellis, and I. F. Silvera, Journal of applied physics **98**, p. 073526 (2005).

[7] P. Celliers, G. Collins, D. Hicks, and J. Eggert, Journal of applied physics **98**, p. 113529 (2005).

[8] R. F. Smith, J. H. Eggert, A. Jankowski, P. M. Celliers, M. J. Edwards, Y. M. Gupta, J. R. Asay, and G. W. Collins, Physical review letters **98**, p. 065701 (2007).

[9] L. Burakovsky, S. Chen, D. Preston, A. Belonoshko, A. Rosengren, A. Mikhaylushkin, S. Simak, and J. Moriarty, Physical review letters **104**, p. 255702 (2010).

[10] C.-H. Lu, B. Remington, B. Maddox, B. Kad, H.-S. Park, S. Prisbrey, and M. Meyers, Acta Materialia **60**, 6601–6620 (2012).

[11] C.-H. Lu, B. Remington, B. Maddox, B. Kad, H.-S. Park, M. Kawasaki, T. Langdon, and M. Meyers, Acta Materialia **61**, 7767–7780 (2013).

[12] A. Comley, B. Maddox, R. Rudd, S. Prisbrey, J. Hawreliak, D. Orlikowski, S. Peterson, J. Satcher, A. Elsholz, H.-S. Park, *et al.*, Physical review letters **110**, p. 115501 (2013).

[13] F. J. Zerilli and R. W. Armstrong, Acta Metallurgica et Materialia **40**, 1803–1808 (1992).

[14] D. L. Preston, D. L. Tonks, and D. C. Wallace, Journal of Applied Physics **93**, 211–220 (2003).

[15] P. Follansbee and U. Kocks, Acta Metallurgica **36**, 81–93 (1988).

[16] P. Follansbee and G. Gray III, Materials Science and Engineering: A **138**, 23–31 (1991).

[17] W. Tong, R. J. Clifton, and S. Huang, Journal of the Mechanics and Physics of Solids **40**, 1251–1294 (1992).

[18] A. B. Tanner and D. L. McDowell, International Journal of Plasticity **15**, 375–399 (1999).

[19] M. A. Meyers, F. Gregori, B. Kad, M. Schneider, D. Kalantar, B. Remington, G. Ravichandran, T. Boehly, and J. Wark, Acta Materialia **51**, 1211–1228 (2003).

[20] M. S. Schneider, B. Kad, D. H. Kalantar, B. A. Remington, E. Kenik, H. Jarmakani, and M. A. Meyers, International journal of impact engineering **32**, 473–507 (2005).

[21] W. Murphy, A. Higginbotham, G. Kimminau, B. Barbrel, E. Bringa, J. Hawreliak, R. Kodama, M. Koenig, W. McBarron, M. Meyers, *et al.*, Journal of Physics: Condensed Matter **22**, p. 065404 (2010).

[22] R. Ravelo, T. Germann, O. Guerrero, Q. An, and B. Holian, Physical Review B **88**, p. 134101 (2013).

[23] B. L. Holian and P. S. Lomdahl, Science **280**, 2085–2088 (1998).

[24] R. Zhang, J. Wang, I. Beyerlein, and T. Germann, Philosophical Magazine Letters **91**, 731–740 (2011).

[25] E. M. Bringa, A. Caro, Y. Wang, M. Victoria, J. M. McNaney, B. A. Remington, R. F. Smith, B. R. Torralva, and H. Van Swygenhoven, Science **309**, 1838–1841 (2005).

[26] T. C. Germann, B. L. Holian, P. S. Lomdahl, and R. Ravelo, Physical Review Letters **84**, p. 5351 (2000).

[27] R. Zhang, T. Germann, J. Wang, X.-Y. Liu, and I. Beyerlein, Scripta Materialia **68**, 114–117 (2013).

[28] B. L. Holian, Physical Review A **37**, p. 2562 (1988).

[29] J.-B. Maillet, M. Mareschal, L. Soulard, R. Ravelo, P. S. Lomdahl, T. C. Germann, and B. L. Holian, Physical Review E **63**, p. 016121 (2000).

[30] R. Ravelo, B. Holian, T. Germann, and L. PS, Physical Review B **70**, p. 014103 (2004).

[31] ExMatEx, "Proxy apps overview," (2015).

[32] P. Lomdahl, P. Tamayo, N. Gronbech-Jensen, and D. Beazley, "50 gflops molecular dynamics on the connection machine-5," in *Supercomputing '93. Proceedings* (1993), pp. 520–527.

[33] D. M. Beazley and P. S. Lomdahl, arXiv preprint comp-gas/9303002 (1993).

[34] D. M. Beazley and P. S. Lomdahl, Computers in Physics **11**, 230–238 (1997).

[35] D. Hull and D. J. Bacon, *Introduction to dislocations* (Elsevier, 2011).

[36] L. Hsiung and D. Lassila, Acta materialia **48**, 4851–4865 (2000).

[37] C. L. Kelchner, S. Plimpton, and J. Hamilton, Physical review B **58**, p. 11085 (1998).

[38] A. Stukowski and K. Albe, Modelling and Simulation in Materials Science and Engineering **18**, p. 085001 (2010).

[39] A. Stukowski, V. Bulatov, and A. Arsenlis, Modelling Simul. Mater. Sci. Eng **20**, p. 085007 (2012).

[40] F. Nabarro, Proceedings of the Physical Society **59**, p. 256 (1947).

[41] A. Sleeswyk, Philosophical Magazine **8**, 1467–1486 (1963).

[42] M. Meyers, H. Jarmakani, E. Bringa, and B. Remington, Dislocations in Solids **15**, 91–197 (2009).

[43] M. Arzaghi, B. Beausir, and L. Tóth, Acta Materialia **57**, 2440–2453 (2009).

[44] J. Christian, Metallurgical Transactions A **14**, 1237–1256 (1983).

[45] C. Woodward and S. Rao, Philosophical Magazine A **81**, 1305–1316 (2001).

[46] M. Duesbery and V. Vitek, Acta Materialia **46**, 1481–1492 (1998).

[47] K. Ito and V. Vitek, Philosophical Magazine A **81**, 1387–1407 (2001).

[48] R. Gröger, A. Bailey, and V. Vitek, Acta Materialia **56**, 5401–5411 (2008).

[49] L. H. Yang, M. Tang, and J. A. Moriarty, Dislocations in Solids **16**, 1–46 (2010).

[50] M. Yamaguchi and V. Vitek, Journal of Physics F: Metal Physics **3**, p. 523 (1973).

[51] V. Vitek and M. Yamaguchi, Journal of Physics F: Metal Physics **3**, p. 537 (1973).

[52] W. Nellis, J. Moriarty, A. Mitchell, M. Ross, R. Dandrea, N. Ashcroft, N. Holmes, and G. Gathers, Physical review letters **60**, p. 1414 (1988).

[53] W. Nellis, A. Mitchell, and D. Young, Journal of Applied Physics **93**, 304–310 (2003).

[54] R. F. Trunin, *Shock compression of condensed materials* (Cambridge university press, 2005).

[55] Q. Johnson, A. Mitchell, R. N. Keeler, and L. Evans, Physical Review Letters **25**, p. 1099 (1970).

[56] M. Rice, R. G. McQueen, and J. Walsh, in *Solid state physics*, Vol. 6 (Elsevier, 1958), pp. 1–63.

[57] L. Al'Tshuler, K. Krupnikov, and M. Brazhnik, Sov. Phys. JETP **34**, 614–619 (1958).

[58] L. Murr, M. Meyers, C.-S. Niou, Y. Chen, S. Pappu, and C. Kennedy, Acta materialia **45**, 157–175 (1997).

[59] L. Hsiung and D. Lassila, Scripta materialia **39**, 603–609 (1998).

[60] W. R. Thissell, A. K. Zurek, D. L. Tonks, and R. S. Hixson, "Experimental quantitative damage measurements and void growth model predictions in the spallation of tantalum," in *AIP Conference Proceedings*, Vol. 505 (AIP, 2000), pp. 451–454.

[61] J. N. Florando, N. R. Barton, B. S. El-Dasher, J. M. McNaney, and M. Kumar, Journal of Applied Physics **113**, p. 083522 (2013).

[62] M. D. Furnish, L. C. Chhabildas, and D. J. Steinberg, "Dynamical behavior of tantalum," in *AIP Conference Proceedings*, Vol. 309 (AIP, 1994), pp. 1099–1102.

[63] G. Whiteman, S. Case, and J. Millett, "Planar shock compression of single crystal tantalum from 6–23 gpa," in *Journal of Physics: Conference Series*, Vol. 500 (IOP Publishing, 2014) p. 112067.

[64] R. White, IRE Transactions on Instrumentation **1**, 294–298 (1962).

[65] C. Skeen and C. York, Applied Physics Letters **12**, 369–371 (1968).

[66] C. Bell and J. Landt, Applied Physics Letters **10**, 46–48 (1967).

[67] N. Anderholm, Applied Physics Letters **16**, 113–115 (1970).

[68] M. Meyers, M. Schneider, H. Jarmakani, B. Kad, B. Remington, D. Kalantar, J. McNaney, B. Cao, and J. Wark, Metallurgical and Materials Transactions A **39**, 304–321 (2008).

[69] J. N. Florando, B. S. El-Dasher, C. Chen, D. C. Swift, N. R. Barton, J. M. McNaney, K. T. Ramesh, K. J. Hemker, and M. Kumar, AIP Advances **6**, p. 045120 (2016).

[70] E. I. Moses, in *Edward Teller Lectures: Lasers and Inertial Fusion Energy* (World Scientific, 2016), pp. 421–426.

[71] J. Soures, R. McCrory, C. Verdon, A. Babushkin, R. Bahr, T. Boehly, R. Boni, D. Bradley, D. Brown, R. Craxton, *et al.*, Physics of Plasmas **3**, 2108–2112 (1996).

[72] J.-P. Davis, Journal of applied physics **99**, p. 103512 (2006).

[73] J. Edwards, K. Lorenz, B. Remington, S. Pollaine, J. Colvin, D. Braun, B. Lasinski, D. Reisman, J. McNaney, J. Greenough, *et al.*, Physical review letters **92**, p. 075002 (2004).

[74] K. T. Lorenz, M. Edwards, S. Glendinning, A. Jankowski, J. McNaney, S. Pollaine, and B. Remington, Physics of plasmas **12**, p. 056309 (2005).

[75] Y. Wang, E. Bringa, J. McNaney, M. Victoria, A. Caro, A. Hodge, R. Smith, B. Torralva, B. Remington, C. Schuh, *et al.*, Applied physics letters **88**, p. 061917 (2006).

[76] C. A. Hall, Physics of Plasmas **7**, 2069–2075 (2000).

[77] C. Hall, J. Asay, M. Knudson, W. Stygar, R. Spielman, T. Pointon, D. Reisman, A. Toor, and R. Cauble, Review of Scientific Instruments **72**, 3587–3595 (2001).

[78] M. E. Savage, K. LeChien, M. Lopez, B. Stoltzfus, W. Stygar, J. Lott, and P. Corcoran, "Status of the z pulsed power driver," in *Pulsed Power Conference (PPC), 2011 IEEE* (IEEE, 2011), pp. 983–990.

[79] J. Brown, C. Alexander, J. Asay, T. Vogler, D. Dolan, and J. Belof, Journal of Applied Physics **115**, p. 043530 (2014).

[80] J. M. D. Lane, S. M. Foiles, H. Lim, and J. L. Brown, Physical Review B **94**, p. 064301 (2016).

[81] J.-P. Davis, J. L. Brown, M. D. Knudson, and R. W. Lemke, Journal of Applied Physics **116**, p. 204903 (2014).

[82] E. Bringa, K. Rosolankova, R. Rudd, B. Remington, J. Wark, M. Duchaineau, D. Kalantar, J. Hawreliak, and J. Belak, Nature materials **5**, p. 805 (2006).

[83] R. E. Rudd, "High-rate plastic deformation of nanocrystalline tantalum to large strains: Molecular dynamics simulation," in *Materials Science Forum*, Vol. 633 (Trans Tech Publ, 2010), pp. 3–19.

[84] J. D. Honeycutt and H. C. Andersen, Journal of Physical Chemistry **91**, 4950–4963 (1987).

[85] P. M. Larsen, S. Schmidt, and J. Schiøtz, Modelling Simul. Mater. Sci. Eng **24**, p. 055007 (2016).

[86] M. P. Allen and D. J. Tildesley, *Computer simulation of liquids* (Oxford university press, 1989).

[87] A. Rahman, Physical Review **136**, p. A405 (1964).

[88] J. Barojas, D. Levesque, and B. Quentrec, Physical Review A **7**, p. 1092 (1973).

[89] L. Verlet, Physical review **159**, p. 98 (1967).

[90] A. Nordsieck, Mathematics of Computation **16**, 22–49 (1962).

[91] R. W. Hockney and J. W. Eastwood, *Computer simulation using particles* (crc Press, 1988).

[92] S. Nosé, Molecular physics **52**, 255–268 (1984).

[93] R. Ravelo, B. L. Holian, and T. C. Germann, "High strain rates effects in quasi-isentropic compression of solids," in *AIP Conference Proceedings*, Vol. 1195 (AIP, 2009), pp. 825–828.

[94] B. Holian, A. Voter, N. Wagner, R. Ravelo, S. Chen, W. G. Hoover, C. Hoover, J. Hammerberg, and T. Dontje, Physical Review A **43**, p. 2655 (1991).

[95] M. S. Daw and M. I. Baskes, Physical review letters **50**, p. 1285 (1983).

[96] M. S. Daw and M. I. Baskes, Physical Review B **29**, p. 6443 (1984).

[97] M. S. Daw, S. M. Foiles, and M. I. Baskes, Materials Science Reports **9**, 251–310 (1993).

[98] M. Finnis and J. Sinclair, Philosophical Magazine A **50**, 45–55 (1984).

[99] F. Ercolessi, E. Tosatti, and M. Parrinello, Physical review letters **57**, p. 719 (1986).

[100] D. Pettifor, Physical review letters **63**, p. 2480 (1989).

[101] S. Foiles, Physical Review B **48**, p. 4287 (1993).

[102] J. A. Moriarty, Physical Review B **38**, p. 3199 (1988).

[103] C. A. Becker, F. Tavazza, Z. T. Trautt, and R. A. B. de Macedo, Current Opinion in Solid State and Materials Science **17**, 277–283 (2013).

[104] C. Liu, C. Xu, Y. Cheng, X. Chen, and L. Cai, Journal of Applied Physics **118**, p. 235901 (2015).

[105] A. P. Moore, J. L. Brown, H. Lim, and J. M. D. Lane, Physical Review Materials **2**, p. 053601 (2018).

[106] D. Tramontina, P. Erhart, T. Germann, J. Hawreliak, A. Higginbotham, N. Park, R. Ravelo, A. Stukowski, M. Suggit, Y. Tang, *et al.*, High Energy Density Physics **10**, 9–15 (2014).

[107] T. Remington, C. Ruestes, E. Bringa, B. Remington, C. Lu, B. Kad, and M. Meyers, Acta Materialia **78**, 378–393 (2014).

[108] M. J. Louwerse and E. J. Baerends, Chemical physics letters **421**, 138–141 (2006).

[109] A. P. Thompson, S. J. Plimpton, and W. Mattson, The Journal of chemical physics **131**, p. 154107 (2009).

[110] K. Kadau, T. C. Germann, and P. S. Lomdahl, International Journal of Modern Physics C **17**, 1755–1761 (2006).

[111] J. N. Glosli, D. F. Richards, K. Caspersen, R. E. Rudd, J. A. Gunnels, and F. H. Streitz, "Extending stability beyond cpu millennium: a micron-scale atomistic simulation of kelvin-helmholtz instability," in *Proceedings of the 2007 ACM/IEEE conference on Supercomputing* (ACM, 2007) p. 58.

[112] T. C. Germann and K. Kadau, International Journal of Modern Physics C **19**, 1315–1319 (2008).

[113] Top 500 list, november 2017, Accessed: April-2018.

[114] F. R. Graziani, V. S. Batista, L. X. Benedict, J. I. Castor, H. Chen, S. N. Chen, C. A. Fichtl, J. N. Glosli, P. E. Grabowski, A. T. Graf, *et al.*, High Energy Density Physics **8**, 105–131 (2012).

[115] S. Plimpton, Journal of computational physics **117**, 1–19 (1995).

[116] J. Mohd-Yusof, Codesign molecular dynamics (comd)proxy app deep dive, 2012.

[117] S. Swaminarayan, K. Kadau, T. C. Germann, and G. C. Fossum, "369 tflop/s molecular dynamics simulations on the roadrunner general-purpose heterogeneous supercomputer," in *Proceedings of the 2008 ACM/IEEE conference on Supercomputing* (IEEE Press, 2008) p. 64.

[118] R. Hockney, S. Goel, and J. Eastwood, Journal of Computational Physics **14**, 148–158 (1974).

[119] T. C. Germann, K. Kadau, and S. Swaminarayan, Concurrency and Computation: Practice and Experience **21**, 2143–2159 (2009).

[120] J. Adams, S. Foiles, and W. Wolfer, Journal of Materials Research **4**, 102–112 (1989).

[121] K. Kadau, T. C. Germann, P. S. Lomdahl, and B. L. Holian, Science **296**, 1681–1684 (2002).

[122] K. Kadau, T. C. Germann, P. S. Lomdahl, and B. L. Holian, Physical Review B **72**, p. 064120 (2005).

[123] S. Zhou, D. Preston, P. Lomdahl, and D. Beazley, Science **279**, 1525–1527 (1998).

[124] S. Zhou, D. Beazley, P. Lomdahl, and B. Holian, Physical Review Letters **78**, p. 479 (1997).

[125] T. C. Germann, K. Kadau, I. M. Longini, and C. A. Macken, Proceedings of the National Academy of Sciences **103**, 5935–5940 (2006).

[126] K. Bergman, S. Borkar, D. Campbell, W. Carlson, W. Dally, M. Denneau, P. Franzon, W. Harrod, K. Hill, J. Hiller, *et al.*, Defense Advanced Research Projects Agency Information Processing Techniques Office (DARPA IPTO), Tech. Rep **15** (2008).

[127] M. A. Heroux, D. W. Doerfler, P. S. Crozier, J. M. Willenbring, H. C. Edwards, A. Williams, M. Rajan, E. R. Keiter, H. K. Thornquist, and R. W. Numrich, Sandia National Laboratories, Tech. Rep. SAND2009-5574 **3** (2009).

[128] P. Cicotti, S. M. Mniszewski, and L. Carrington, "An evaluation of threaded models for a classical md proxy application," in *Hardware-Software Co-Design for High Performance Computing (Co-HPC), 2014* (IEEE, 2014), pp. 41–48.

[129] P. Cicotti, M. Shantharam, and L. Carrington, "Tuning tasks, granularity, and scratchpad size for energy efficiency," in *Proceedings of the 2nd International Workshop on Hardware-Software Co-Design for High Performance Computing* (ACM, 2015) p. 9.

[130] "Nvidia/comd-cuda," (2003).

[131] H. H. Wu and D. R. Trinkle, Computational Materials Science **47**, 577–583 (2009).

[132] A. Stukowski, Modelling and Simulation in Materials Science and Engineering **20**, p. 045021 (2012).

[133] D. Faken and H. Jónsson, Computational Materials Science **2**, 279–286 (1994).

[134] H. Tsuzuki, P. S. Branicio, and J. P. Rino, Computer physics communications **177**, 518–523 (2007).

[135] A. Stukowski, Modelling Simul. Mater. Sci. Eng **18**, p. 015012 (2010).

[136] D. L. Theobald, Acta Crystallographica Section A: Foundations of Crystallography **61**, 478–480 (2005).

[137] B. K. Horn, JOSA A **4**, 629–642 (1987).

Appendix A

QIC Implementation

A.1 Initialize Deformation

```
void init_Deformation(double strain, int ifun, int esign){

    Deformation=1;
    Tramp = t_ramp;
    TStopRamp = TotalTime + t_ramp;
    Isenfun = ifun;

    // Compress if esign = 1
    if (esign==1)
      strain=-strain;

    // Constant engineering strain rate
    if (ifun==0)
      edotavg=strain/t_ramp;

    else
      edotavg=log(1+strain)/(t_ramp);

    // bring center of simulation box to the origin.
    restore_Center();
```

```
}
```

A.2 Strain Rate Function

```
// edotfun(double dt}: evaluates the strain rate function according to
// preset functional form in Set_deformation function
double edotfun(double dt)
{
  double pi = acos(-1);
  double t;                 // time
  double erate;             // strain rate at time t

  t=(TotalTime+dt)/Tramp;

  // calculate strain rate if Deformation = 1
  if (Deformation){

    // const. true strain rate
    if (Isenfun==1)
      erate=edotavg;

    // parabolic true strain rate
    else if (Isenfun==2)
      erate=6.0*edotavg*t*(1-t);

    // Sin^2 true strain rate
    else if (Isenfun==3)
```

```
      erate=2*edotavg*(sin(pi*t)*sin(pi*t));

    // const. engineering strain rate
    else
      erate=edotavg/(1.0 + edotavg*(TotalTime+dt));
  }
  else erate=0.0;

  return erate;
}
```

A.3 Position Update

```
void integrate_adv_coord(void)
{
  int        i;
  Particle   *pt;
  double     efac1,efac2;        // multiplication factors
  double     edot;               // strain rate
  extern double edotfun(double dt);

  //Isentropic deformation (uniaxial:Z-direction) if Deformation = 1
  if ((Deformation==1) && (Anneal==0)) {
    if ((TotalTime+Dt/2) > Tramp) {
      Deformation=0;
      edot=0.0;
      efac1=1.0;
```

```
    efac2=Dt;
  }
  else {
    edot=edotfun(0.5*Dt);
    efac1=(1.0 + edot*Dt/2)/(1.0 - edot*Dt/2);
    efac2=Dt/(1.0 - edot*Dt/2);
  }
}
else {
  edot=0.0;
  efac1=1.0;
  efac2=Dt;
}

pt = (Particle *) Cells[0][0][0].ptr;

// Update positions
for (i = 0; i < nAtoms; i++, pt++) {
  pt->r.x = (pt->r.x + Dt * pt->s.x);
  pt->r.y = (pt->r.y + Dt * pt->s.y);
  pt->r.z = (efac1 * pt->r.z + efac2 * pt->s.z);
}

// Update simulation box length if Deformation =1
if ((Deformation==1) && (Anneal==0)) {
  Pmin_z *= efac1;
  Pmax_z *= efac1;
  SPaSM_subdivide();
```

```
  }
}
```

A.4 Velocity Update

```
void integrate_adv_velocity(void)
{
  int        i;
  Particle   *pt;
  double     invmass;            // 1/mass
  double     vfac1,vfac2;        // multiplication factors
  double     edot;               // strain rate

  extern double edotfun(double dt);

  // Isentropic deformation (uniaxial:Z-direction) if Deformation=1
  if ((Deformation==1) && (Anneal==0)) {
    if ((TotalTime+Dt) > Tramp) {
      Deformation=0;
      edot=0.0;
      vfac1=1.0;
      vfac2=Dt;
    }
    else {
      edot=edotfun(Dt);
      vfac1=(1.0 - edot*Dt/2)/(1.0 + edot*Dt/2);
      vfac2=Dt/(1.0 + edot*Dt/2);
    }
```

```
  }
  else {
    edot=0.0;
    vfac1=1.0;
    vfac2=Dt;
  }

  pt = (Particle *) Cells[0][0][0].ptr;

  // Update velocities
  for (i = 0; i < nAtoms; i++, pt++) {
    invmass = 1.0 / Masses[pt->type];
    pt->s.x = (pt->s.x+Dt*invmass*pt->f.x);
    pt->s.y = (pt->s.y+Dt*invmass*pt->f.y);
    pt->s.z = (vfac1 * pt->s.z + vfac2 * invmass*pt->f.z);
  }
}
```

Appendix B

Polyhedral Template Matching: Algorithm

Algorithm 1 PTM algorithm

procedure PTM_METHOD($\mathbf{A}$)

▷ A is the set of positions of central atom and its nearest neighbors ▷ A is ordered using Euclidean or topological ordering

$r^* \leftarrow$ FLOAT_MAX ▷ Set initial RMSD value
$\mathbf{Q}^* \leftarrow \mathbf{1}$ ▷ Unit rotation
$\mathbf{S}^* \leftarrow$ disordered ▷ No structure identified (yet)
for S $\in$ {SC, BCC, HCP, ICO, FCC} **do**
  $\mathbf{W} \leftarrow$ ReferenceTemplate($\mathbf{S}$)
  $\mathbf{U} \leftarrow \{\overrightarrow{a_j} \in \mathbf{A} \mid \mathrm{j} \leq \mid \mathbf{W} \mid\}$ ▷ Select central atom
  $\mathbf{W} \leftarrow$ Conv($\mathbf{U}$) ▷ Calculate convex hull for $\mathbf{U}$
  if $\overrightarrow{a_j} \notin \mathbf{C}$ **then** ▷ Concex hull must not contain central atom
    $\mathbf{G} \leftarrow$ CanonicalForm(Graph($\mathbf{C}$))
    for each triangulation $\mathbf{T}_i$ of Conv($\mathbf{W}$) **do**
      $\mathbf{G}_{ref} \leftarrow$ CanonicalForm(Graph($\mathbf{T}_i$))
      if $\mathbf{G} = \mathbf{G}_{ref}$ **then** ▷ Test graph isomorphism
        for each automorphism $\mathbf{A}_j$ of $\mathbf{G}$ **do**
          $\mathbf{U}' \leftarrow \mathbf{U} \times \mathbf{A}_j$
          {r, $\mathbf{Q}$} $\leftarrow$ RMSD($\mathbf{U}'$, $\mathbf{W}$) ▷ Optimal superposition
          if $r < r^*$ **then**
            $r^* \leftarrow r$
            $\mathbf{Q}^* \leftarrow \mathbf{Q}$
            $\mathbf{S}^* \leftarrow \mathbf{S}$
return $\{r^*, \mathbf{Q}^*, \mathbf{S}^*\}$:

Appendix C

Python Script for Calculating TVF Analysis Using OVITO Python Interface

```
from ovito.io import *
from ovito.modifiers import *
from ovito.vis import *

import numpy
import math
from PyQt5 import QtCore

# import cfg file
savF="CFG_FILE.cfg"
savPath="/PATH/TO/CFGS"
print("Configuration file: %s" %(savPath+"/"+savF))
node = import_file(savPath+"/"+savF)
print ("Configuration file's imported.")



# Step1:
# PHTM
```

```
modifier = PolyhedralTemplateMatchingModifier()
modifier.output_orientation = True
modifier.rmsd_cutoff = 0.1

#PolyhedralTemplateMatchingModifier.Type.BCC
node.modifiers.append(modifier)

# Step2:
# CNA
modifier = CommonNeighborAnalysisModifier()
modifier.cutoff = 3.659
node.modifiers.append(modifier)

# Step3:
# color coding with respect to orientation.w
modifier = ColorCodingModifier(
    particle_property = "Orientation.W",
    gradient = ColorCodingModifier.Viridis())
modifier.start_value = 0.75
modifier.end_value = 1.0
node.modifiers.append(modifier)

# Step4:
# Select all atoms with orientation.w corresponding to twins:
node.modifiers.append(
    SelectExpressionModifier(
        expression = 'Orientation.W<0.97 && StructureType==3'))
node.compute()
```

```
natoms=node.source.number_of_particles
ntwinatoms=node.output.attributes['SelectExpression.num_selected']
# Demonstrating two ways to get the number of selected atoms:
print("Twin atoms: "+ str(ntwinatoms))
print("Twin fraction (in %%): %.5f" % (ntwinatoms/natoms*100.0))


node.compute()

##############################################
# rest of the script is for creating images #
##############################################

# Create an overlay.
tripod = CoordinateTripodOverlay()
tripod.size = 0.07
tripod.alignment = QtCore.Qt.AlignRight ^ QtCore.Qt.AlignBottom

#setup display
pos_prop = node.source.particle_properties.position
pos_prop.display.radius = 1.4

# remove particles
#pos_prop.display.enabled = False

# setup Viewport
vp = Viewport()
```

```
vp.type = Viewport.Type.PERSPECTIVE



# Attach overlay to the active viewport.
vp = ovito.dataset.viewports.active_vp
vp.overlays.append(tripod)

# camera direction
cdir = [-2, 1.2, 1.6]
cdir = [0.864478, -0.395888, 0.30976]
cdir /= numpy.linalg.norm(cdir)

# camera position
cpos = [-886.199, 354.893, -296.144]
cpos /= numpy.linalg.norm(cpos)
dist = 1250

dx = cdir[0]
dy = cdir[1]
dz = cdir[2]
px = dist * cpos[0]
py = dist * cpos[1]
pz = dist * cpos[2]

vp.camera_pos = (px, py, pz)
vp.camera_dir = (dx, dy, dz)
vp.fov = math.radians(35.0)
```

```
# render settings
settings = RenderSettings()
settings.renderer = TachyonRenderer()
settings.filename = 'image_' + savF + '.png'
settings.size = (800, 800)

# add node to the scene
node.add_to_scene()

# render scene
vp.render(settings)
```

Appendix D

Python Script for Dislocation Density Analysis Using OVITO Python Interface

```
from ovito.io import *
from ovito.modifiers import *
from ovito.vis import *

import numpy
import math
from PyQt5 import QtCore

# import cfg file
savF="CFG_FILE.cfg"
savPath="/PATH/TO/CFGS"
print("Configuration file: %s" %(savPath+savF))
node = import_file(savPath+"/"+savF)
print ("Configuration file imported.")

# Extract dislocation lines from a crystal with BCC structure:
modifier = DislocationAnalysisModifier()
modifier.input_crystal_structure = DislocationAnalysisModifier.Lattice.BCC
```

```
node.modifiers.append(modifier)
node.compute()

total_line_length = node.output.attributes[
    'DislocationAnalysis.total_line_length']
print("Volume (in A^3): %f" % (node.output.cell.volume))
print("Dislocation density (in 1/cm^2): %e" % (
    total_line_length / node.output.cell.volume * 1.0e+16))

# print dislocation densities
h_111 = node.output.attributes['DislocationAnalysis.length.1/2<111>']
print("1/2<111> density (in 1/cm^2): %e" % (
    h_111 / node.output.cell.volume * 1.0e+16))
h_110 = node.output.attributes['DislocationAnalysis.length.<110>']
print("<110> density (in 1/cm^2): %e" % (
    h_110 / node.output.cell.volume * 1.0e+16))
h_100 = node.output.attributes['DislocationAnalysis.length.<100>']
print("<100> density (in 1/cm^2): %e" % (
    h_100 / node.output.cell.volume * 1.0e+16))

##############################################
# rest of the script is for creating images #
##############################################

# remove defect mesh
sm = node.output.surface
#sm.display.enabled = False
sm.display.cap_transparency = .90
```

```
sm.display.surface_transparency = .70

# Create an overlay.
tripod = CoordinateTripodOverlay()
tripod.size = 0.07
tripod.alignment = QtCore.Qt.AlignRight ^ QtCore.Qt.AlignBottom

#setup display
pos_prop = node.source.particle_properties.position
pos_prop.display.radius = 1.4

# remove particles
pos_prop.display.enabled = False

# setup Viewport
vp = Viewport()
vp.type = Viewport.Type.PERSPECTIVE



# Attach overlay to the active viewport.
vp = ovito.dataset.viewports.active_vp
vp.overlays.append(tripod)

# camera direction
cdir = [-2, 1.2, 1.6]
cdir = [0.864478, -0.395888, 0.30976]
cdir /= numpy.linalg.norm(cdir)
```

```
# camera position
cpos = [-886.199, 354.893, -296.144]
cpos /= numpy.linalg.norm(cpos)
dist = 1250

dx = cdir[0]
dy = cdir[1]
dz = cdir[2]
px = dist * cpos[0]
py = dist * cpos[1]
pz = dist * cpos[2]

vp.camera_pos = (px, py, pz)
vp.camera_dir = (dx, dy, dz)
vp.fov = math.radians(35.0)

# render settings
settings = RenderSettings()
settings.renderer = TachyonRenderer()
settings.filename = 'image_' + savF + '.png'
settings.size = (800, 800)

# add node to the scene
node.add_to_scene()

# render scene
vp.render(settings)
```

Vitae

Jayalath A M M Abeywardhana was born and raised in Sri Lanka. He graduated from high school in 2005 at Nalanda Central College in Muniangoda, Sri Lanka. In 2006, he was selected to pursue his Bachelor of Science at University of Colombo, Sri Lanka (UoC) by a competitive island wide selection criteria. He received his Bachelor of Science in 2010 from UoC with a major in Computational Physics. He worked as a teaching assistant (2011) and as a financial advisor (2012) before he was accepted to the Computational Science doctoral program at The University of Texas at El Paso (UTEP). In 2016 he received his Masters of Science in Computational Science as a completion of a milestone toward the Ph.D. While Studying at UTEP, In 2018 he completed his Doctor in Ph.D. in Computational Science at UTEP. He worked as a research assistant under supervision of Dr. Ramon Ravelo as well as a teaching assistant for the Mathematics Department.

Email: mabeywardhana@miners.utep.edu