

2016-01-01

Molecular Dynamics Proxy Applications for Optimizing Performance of Simulations under Extreme Conditions

Jayalath Abeywardhana Mudiyansele Madawa Abeywardhana
University of Texas at El Paso, mabeywardhana@miners.utep.edu

Follow this and additional works at: https://digitalcommons.utep.edu/open_etd



Part of the [Materials Science and Engineering Commons](#), and the [Mechanics of Materials Commons](#)

Recommended Citation

Abeywardhana, Jayalath Abeywardhana Mudiyansele Madawa, "Molecular Dynamics Proxy Applications for Optimizing Performance of Simulations under Extreme Conditions" (2016). *Open Access Theses & Dissertations*. 788.
https://digitalcommons.utep.edu/open_etd/788

This is brought to you for free and open access by DigitalCommons@UTEP. It has been accepted for inclusion in Open Access Theses & Dissertations by an authorized administrator of DigitalCommons@UTEP. For more information, please contact lweber@utep.edu.

MOLECULAR DYNAMICS PROXY APPLICATIONS FOR OPTIMIZING PERFORMANCE OF SIMULATIONS UNDER EXTREME CONDITIONS

JAYALATH ABEYWARDHANA MUDIYANSELAGE MADAWA ABEYWARDHANA

Master's Program in Computational Science

APPROVED:

Ramon Ravelo, Ph.D., Chair

Shirley Moore, Ph.D.

Rajendra Zope, Ph.D.

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

Copyright ©

by

Jayalath A M M Abeywardhana

2016

MOLECULAR DYNAMICS PROXY APPLICATIONS FOR OPTIMIZING
PERFORMANCE OF SIMULATIONS UNDER EXTREME CONDITIONS

by

JAYALATH ABEYWARDHANA MUDIYANSELAGE MADAWA ABEYWARDHANA,

B.Sc.

THESIS

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

MASTER OF SCIENCE

Computational Science

THE UNIVERSITY OF TEXAS AT EL PASO

May 2016

Acknowledgements

I would like to thank my research advisor Dr. Ramon Ravelo for his valuable time, and guidance which made this study a reality. Gratitude is also paid for my parent, wife, and son Kaveesh for their sacrifices.

Part of this work supported by the Air Force Office of Scientific Research under AFOSR Award No. FA9550-12-1-0476 and by the Department of Energy under contract DE-AC52-06NA25396.

Abstract

Exascale performance of scientific application is a prominent goal for the scientific community and computer hardware vendors. As a result, several proxy applications for the widely used scientific codes have been developed with the purpose of testing performance, memory and power consumption issues associated with simulations at the exascale. Co-MD is a proxy application for classical Molecular Dynamics (MD) codes based on the (SPaSM) Los Alamos code.

This work comprises two different but related computational aspects: performance evaluation and optimization of Co-MD (Co-Design Molecular Dynamics) for different computer architectures/execution models, and direct application of SPaSM (Scalable Parallel Short-range Molecular Dynamics) to the study of material strength employing large-scale MD simulations.

Co-MD had been implemented in different computer architectures. An OpenCL implementation of Co-MD was developed starting from the MPI version, which included optimized interatomic force evaluation kernels and atom neighbor lists to eliminate exhaustive searches within 27 neighbor cells. Co-MD supports Lennard-Jones (LJ) pair-potential and Embedded Atom Method (EAM) many-body potentials for evaluating the atomic interactions. The embedded atom method is widely used in high performance computing MD simulations since it can deliver accurate results at a low computational cost. Generally, EAM potentials are in tabular format and the access pattern of these tables is random. Performance of EAM force evaluation kernels was studied by developing routines that simulate a multi-component system within Co-MD.

Large-Scale molecular dynamics (MD) simulations utilizing the Los Alamos SPaSM code were carried out to investigate the strength of materials subjected to high strain rate of deformation under quasi-isentropic compression (QIC). Defective copper and tantalum crystals were chosen as prototypes due to the large body of experimental data on these systems. The atomic interactions were modeled employing (EAM) potentials. Quasi-isentropic uniaxial compression is achieved by

incorporating a strain rate function in the positions and velocities equations of motion, which makes possible to quantify plastic work with temperature changes. In this new formalism the change in internal energy is exactly equal to the work done in compression. We examined the deformation mechanisms and material strength for strain rates in the 10^9 - 10^{12} s ranges for both Cu and Ta defective crystals. We find that the strain rate dependence of the flow stress in this strain rate regime follows a power law with an exponent close to 0.45 for Tantalum and 0.49 for Copper. Dislocations analysis was also performed on the compressed samples using DXA-CAT, which identifies different types of the dislocations and length of them. The relation between temperature change and the final dislocation density was analyzed as a way of studying the mobility of the dislocations.

Table of Contents

Acknowledgements	iv
Abstract	v
Table of Contents	vii
List of Figures	ix
Introduction	1
1.2 Molecular Dynamics Codes	2
1.3 MD Proxy Applications	4
1.4 Material Strength	5
1.4.1 Shock Waves	5
1.4.2 Stress Tensor	6
1.4.3 Strain Rate Dependence of Material Strength	8
1.4.4 Dislocations and Their Effect on Material Strength	10
1.5 Outline of the Thesis	11
Chapter 2: Quasi-Isentropic Implementation in SPaSM	13
2.1 MD Solvers	13
2.1.1 Verlet Integration	15
2.1.2 Leap-frog Method	16
2.2 Inter-atomic Interactions	16
EAM Model Potential	17
2.3 Ensembles	18
2.3.1 Microcanonical Ensemble (NVE)	18
2.3.2 Canonical Ensemble (NVT)	18
2.4 Quasi-Isentropic Compression (QIC)	19
2.4.1 Implementation of QIC Equations of Motion in SPaSM	22
2.5 GPU Execution Model and MD Codes	23
Heterogeneous Computing	24
Chapter 3: Proxy Applications for Molecular Dynamics	26
3.1 Co-design Molecular Dynamics (CoMD)	26
3.1.1 CoMD Data Structures	27

Domain Decomposition	27
Data Layout.....	27
Interatomic Interactions	28
Neighborlists.....	29
3.1.2 GPU Versions of CoMD.....	29
CUDA Implementation of CoMD	29
OpenCL Implementation of CoMD	30
3.1.3 CoMD Performance Analysis	31
Binary Lattice vs Single Lattice.....	31
Performance Improvements by Implementing Neighborlists for CoMD-OpenCL	33
Chapter 4: Large-scale MD Study of Material Strength.....	34
4.1 Simulation Details and Analysis Tools.....	35
Crystal Structure Analysis and Visualization	36
DXA Crystal Analysis Tool.....	36
Visualization	37
4.2 Analysis and Results	38
Chapter 5: Future Work	44
Strain Rate Dependence of Material Strength Using Polycrystalline Samples	44
References	46
Appendix.....	49
Source Code for QIC Implementation	49
Position Update	49
Velocity Update	50
Strain Rate Calculation	51
Curriculum Vitae	52

List of Figures

Figure 1 Different types for MD codes, corresponding proxy applications, and the level of scale they are focused on. This figure was obtained from [4] .	3
Figure 2 The Cauchy stress tensor components.	7
Figure 3 Shear stress response behavior over a wide range of average dislocation velocity. Different color schemes are used for temperatures. For MD data, red, green, and blue line are drawn for temperature of 300 K, 700 K, and 1000 K respectively. And for experimental data, yellow, magenta, cyan, red, green blue lines are correspond to temperatures at 298 K, 473 K, 673 K, 873 K, 1072 K, and 1273 K respectively. This figure was obtained from [11]	11
Figure 4 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 $\sin^2 \theta$	20
Figure 5 Time profiles of temperature (a) and shear stress (b) obtained from Tantalum QIC simulations at 0.20 strain for different strain rates.	22
Figure 6 An illustration of a Fermi stream multiprocessor with 32 CUDA cores, 16 load/store units, four special function units, 64kB shared L1 cache, and 128kB register file.	24
Figure 7 Data layout of Array of Structures (shown in top) vs Structure of Arrays (shown in the bottom) [22].	28
Figure 8 Unit cell of an $L1_2$ structure with Cu atoms in the faces and the Au atoms in the corners of the fcc structure	32
Figure 9 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.	32
Figure 10 Execution time of the main loop for binary and single lattice for different number of atoms.	33
Figure 11 Execution time of EAM force evaluation kernels with and without using neighborlist per iteration for different number of atoms is shown in this graph.	34
Figure 12 Execution time of the main loop with and without using neighborlist per iteration for different number of atoms is shown in this graph.	34
Figure 13 Orientation Imaging Map (OIM) used for both FCC and BCC materials.	38
Figure 14 Time profile of dislocation line density for a Ta sample compressed at constant strain rate of $3.3 \times 10^9 \text{ s}^{-1}$.	39

Figure 15 Dislocation line density at 300ps as a function of applied strain-rate for Tantalum QIC simulation at a fixed strain of 0.20.....	40
Figure 16 Snapshots at 250ps of Tantalum sample compressed quasi-isentropically to 20% (left), and 25% (right), at different strain-rates: 10^9 - 10^{10} s ⁻¹ . Top row of frames shows only defective atoms; bottom row of frames shows only bcc atoms colored according to their local orientation with respect to the compression direction (horizontal axis): red = (100), blue = (111).	41
Figure 17 Strain-rate dependence of yield strength of Tantalum from QIC simulations compared with predictions of the PTW continuum model [6] (red line). At high strain rates, the shear stress follows a power law in strain rate with an exponent of 0.45. The solid line is a fit to QIC data using a generalized Lorentzian	42
Figure 18 Shear stress response as a function of applied strain rate for Copper. MD data (red circles) are plotted along with experimental data and model predictions. The dashed line is a fit to the MD data.	43
Figure 19 A polycrystalline sample. This image was obtained from [27].....	45

Introduction

There is significant interest in understanding the behavior of materials exposed to extreme environments. Here, extreme environments or extreme “conditions” refers to states of matter generated far from equilibrium, like at very high temperatures, pressures or strain-rates. Understanding and predicting the behavior of materials to say high pressure or high rates of deformation, is crucial in the developing of new materials and many empirical models have been formulated for predicting the behavior of materials based for the most part on available experimental data.

Shock loading of condensed matter exerts very high strain-rates and as a result, subjects the sample to high internal pressures and temperatures. In a large variety of situations, --such as high-speed collisions, explosive welding, armor penetration, meteor impacts, interstellar dust dynamics, and inertial confinement fusion to name just a few, condensed matter could experience high strain-rates in the range applicable to shock loading conditions. Shock wave experiments in solids are most commonly conducted by using explosives, or by colliding a projectile accelerated by a compressed gas delivery system (known as a gas-gun) or by using strong magnetic fields to propel an impactor into the target sample. These experiments can generate shock waves with pressures up to a few hundreds of gigapascals (GPa).

Because of the extreme conditions and short time scales, experimental data collection is challenging and modeling and computer simulations play a major role in improve our understanding of material behavior in such environments.

Typically, the strain-rates in shock compression experiments are less than 10^7 s^{-1} . The dynamic response of a solid to strain rates $> 10^7 \text{ s}^{-1}$, while for the most part unexplored experimentally, has been modeled using a wide variety of constitutive models of material deformation and strength. However, many of the predictions and assumptions of these models remain to be tested experimentally. Large-scale MD simulations is a well-suited computational

technique for studying high-strain rate phenomena and have been used with great success to explore the dynamic response of material deformation under shock-wave loading. Fundamentals of MD codes are presented in the next subsection.

1.2 Molecular Dynamics Codes

Solving Classical equations of motion (Newtonian equations) for a set of molecules of atoms to predict future states of an atomic system is coined as Molecular Dynamics. With the development of the digital computers, the evaluation of Newtonian equations was handed to the digital computers. Thus, the era of MD Simulations was initiated. The first ever MD simulation for a system of hard spheres was obtained by Alder and Wainwright in 1957 for several hundreds of particles [1]. After several years, A. Rahman was successfully simulated a system of 864 liquid Argon atoms interacting with a Lennard-Jones potential on a CDC 3600 digital computer [2].

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, N₂, and H₂O) to large molecules such as proteins [3]. The results obtained from these simulations were compared to results obtained experimentally. The comparison 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 move forward in time the state of the system.

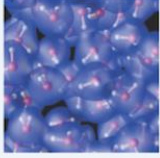
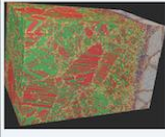
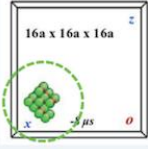
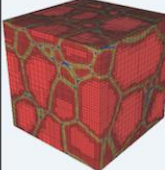
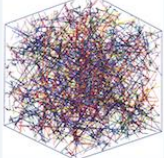

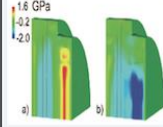
Ab-initio	MD	Long-time	Phase Field	Dislocation	Crystal	Continuum
Inter-atomic forces, EOS	Defects and interfaces, nucleation	Defects and defect structures	Meso-scale multi-phase evolution	Meso-scale strength	Meso-scale material response	Macro-scale material response
						
Code: Qbox/LATTE Motif: Particles 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 <i>in situ</i> visualization Prog. Model: MPI + Threads	Code: SEAKMC Motif: Particles and defects, explicit time integration, neighbor and linked lists, and <i>in situ</i> 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 arithmetic, 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 1 Different types for MD codes, corresponding proxy applications, and the level of scale they are focused on. This figure was obtained from [4] .

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 1 shows some of the many codes the scientific community is currently using and which structural scales these codes are focused on. Since the level of the structure focused on varies among these codes, the underlying algorithms vary significantly among these codes.

Execution time of these codes expands well beyond millions of CPU hours. Optimal execution of these production level codes is important, as they can be computationally expensive and use much of the supercomputing resources available to the scientific community. 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.

1.3 MD Proxy Applications

Currently, 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 updating with the advances of underlying technologies.

Optimum execution of scientific applications on these novel architectures and all new features added to CPU is essential as the current generation's supercomputing facilities are equipped with novel hardware devices. Introducing mini applications that represent the type of algorithmic complexity of the production level codes and optimizing them for architectural differences of the hardware avoid optimizing each production level MD application for new architectures as they are arrived or upgraded. 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 are available to external collaborators. This enables both the hardware vendors and software developers to co-optimize their products to ensure that future machines are able to provide the performance needed to solve these challenging problems.

1.4 Material Strength

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.

1.4.1 Shock Waves

A shock wave is a type of propagating disturbance. 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.

Material properties under extreme strain rates have been studied using non-equilibrium molecular dynamics (NEMD) simulations. 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 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 system length scale in the shock direction depends on the shock strength and can range from hundreds of nanometers to few microns.

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 [5]:

Mass:

$$\rho_0 u_s = \rho (u_s - u_p) \quad 1.1$$

where u_p is the particle velocity and u_s is the shock velocity.

Momentum:

$$P_{zz} = P_0 + \rho_0 u_s u_p \quad 1.2$$

Energy:

$$E_H = E_0 + \frac{1}{2} (P_{zz} + P_0) (V_0 - V) \quad 1.3$$

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 1.4 and the strain rate at the shock front by equation 1.5.

$$\varepsilon = \frac{u_p}{u_s} \quad 1.4$$

$$\dot{\varepsilon} = \frac{\varepsilon u_s}{\lambda} = \frac{u_p}{\lambda} \quad 1.5$$

Where λ is the width of the shock.

1.4.2 Stress Tensor

Deformation of material can be described using a stress tensor S_{ik} where i and k , are the directions independently selected from x, y, and z. The force in the direction of k , acting on an area which has its' normal is oriented in the direction i is denoted by S_{ik} . Figure 2 shows these tensor components drawn on a cubic cell. When $i = k$ i.e. the components σ_{xx} , σ_{yy} , and σ_{zz} are known

as normal stresses. According to Cauchy theorem, only nine stress components from three planes are needed to describe the state of stress at a point P . And these nine components can form a matrix and represented as:

$$\sigma_{ij} = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{pmatrix} \quad 1.6$$

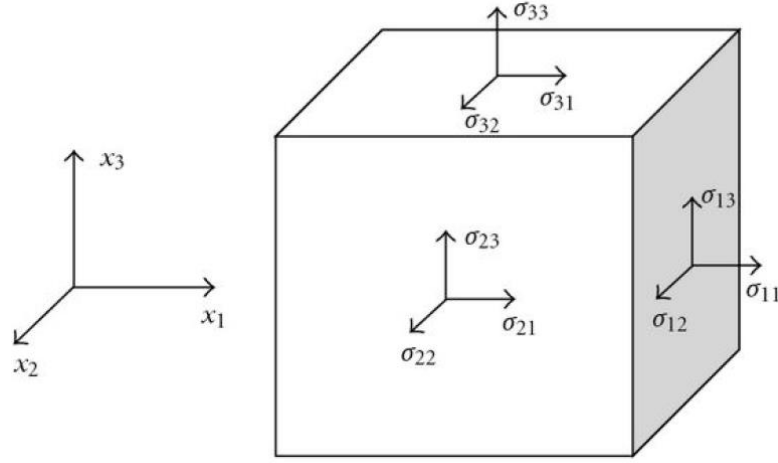


Figure 2 The Cauchy stress tensor components.

By convention, the direction of shock wave propagation or compression considered to be along z axis. The shear stress of the material along the direction of the load is a measurement of the strength, and can be calculated using equation 1.7 for z direction.

$$t_{zz} = \frac{1}{2} \left[P_{zz} - \frac{1}{2} (P_{xx} + P_{yy}) \right] \quad 1.7$$

where, P_{xx} , P_{yy} , and P_{zz} are the trace of the pressure tensor.

1.4.3 Strain Rate Dependence of Material Strength

The shear stress response of solid deformations at strain rates ($> 10^8 \text{ s}^{-1}$), remains for the most part unexplored experimentally due to accurate estimation of strain-rate and kinetics are difficult in these time scales. Figure shows shear stress vs. strain rate experimental data for Copper samples from several shock-wave experiments [6 - 12]. At very high strain rates ($> 10^8 \text{ s}^{-1}$) thermal activation mechanisms of plastic deformation are thought to be less important than dislocation motion opposed by phonon drag. Most models agree in the thermally activated regime, where model parameters are calibrated against experimental data. At high strain rates, where no experimental data is available, models differ significantly from one another and their predictions need to be tested via accurate computer simulations.

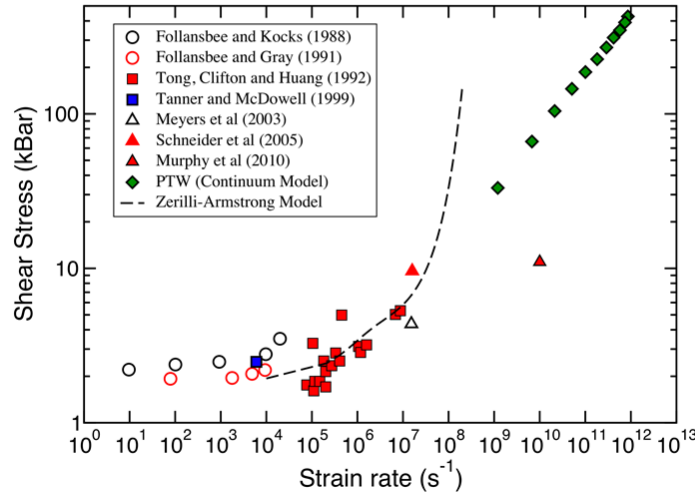


Figure 3 The strain-rate dependence of shear stress in Copper. The data points in the graph are experimental results. The dashed line represents the Zerilli-Armstrong (ZA) model [6] and the green diamonds represent the predictions of the Preston-Tonks-Wallace (PTW) model [7]. The ZA and PTW models deviate considerably at high strain rates.

NEMD simulations of shock wave propagation have been performed for studying material strength and validate strength models such as the Preston-Tonks-Wallace (PTW) model [7] at very high strain rates.

In 1988 B.L. Holian performed NEMD simulations on fcc structures oriented in $\langle 100 \rangle$ and discovered three regimes a system can be in when shocked [8]. These are as follows in the decreasing order of shock strength or strain ϵ .

- Steady overdriven plastic wave
- Steady plastic wave preceded by a non-steady elastic precursor
- Unsteady elastic wave

According to the equation 1.5 the strain rate imposed by the shock depend on the width of the shock. Therefore in order to sample lower strain rates in shock wave simulations, the shock thickness needs to be increased. However, this cannot be done arbitrarily as the thickness of the shock depends on the dissipative processes present. The width of shock waves in single crystals is controlled primarily by phonon dissipation which is small corresponding to short widths, which translate to very high strain rates ($\sim 10^{10} \text{ s}^{-1}$).

To sample strain rates of the order of 10^7 s^{-1} , the shock width is of the order of microns which translates into simulations comprising billions of atoms to achieve these strain rates. Simulations of this scale are currently unrealistic.

Maillet et al proposed an alternative method termed the Hugoniosat [9], that is an equilibrium MD method for simulating shocked states behind the shock front. In this method, initially, the system is compressed instantaneously either uniaxially or isotropically to a predetermined final volume, and then the temperature is relaxed to the final internal energy of the shock Hugoniot by time-reversible integral feedback. A drawback of the method, however, is the way in which the uniaxial deformation is applied to the system. Since the system is instantaneously compressed, the initial temperature and stresses can become unrealistically large, in particular at very high compressions (strains) and the initial imposed strain rates are enormous: of the order of the (final strain)/(simulation time step) ($\sim 10^{14} \text{ s}^{-1}$).

An improvement to the Hugoniosat was formulated by Ravelo et al [7] in which the system is not compressed instantaneously to a final volume but rather gradually to the desired longitudinal stress, again via integral feedback. This method captures the finite strain-rates that a system

undergoes during shock compression. The formulation of this method is discussed in the section 2.2 of this thesis.

In this study, a new formulation, Quasi-Isentropic Compression (QIC) was implemented into the MD code SPaSM [10]. The details of this new formulation and its implementation into SPaSM are discussed in section 2.2.1. With this novel functionality of the code, strength properties can be studied to much lower strain rates, down to lower than 10^9s^{-1}

1.4.4 Dislocations and Their Effect on Material Strength

Strain rate dependence of material strength is controlled by the production of defects and dislocations. Dislocations are abrupt changes in the regular ordering of atoms, along a line (dislocation line) in the solid. Edge dislocations occur when an extra plane is inserted. The dislocation line is at the end of the plane. 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.

Dislocations move along planes in closed packed crystal structures. The rate at which a material deforms plastically depends on the mobility of the dislocations. Thus, strengthening the material consists in hindering dislocation motion.

The density of the defects ρ in the system and the mobility of the dislocations v can be associated with the plastic strain rate $\dot{\epsilon}$ according to (Orowan equation),

$$v = \frac{\dot{\epsilon} M}{\eta \rho b} \quad 1.8$$

Here M is a geometrical factor, η is a scaling parameter of order unity and b is the Burger's vector. According to the equation 1.8, for fixed mobility, the density of dislocation is linearly proportional to the strain-rate, thus, faster the compression, more defects will be generated in the

material, causing dislocations to move slower. This can be explained if defects are considered as energy barriers for dislocation motion. Figure 3 shows shear stress response over a wide range of average dislocation velocities obtained from experiments and MD simulations.

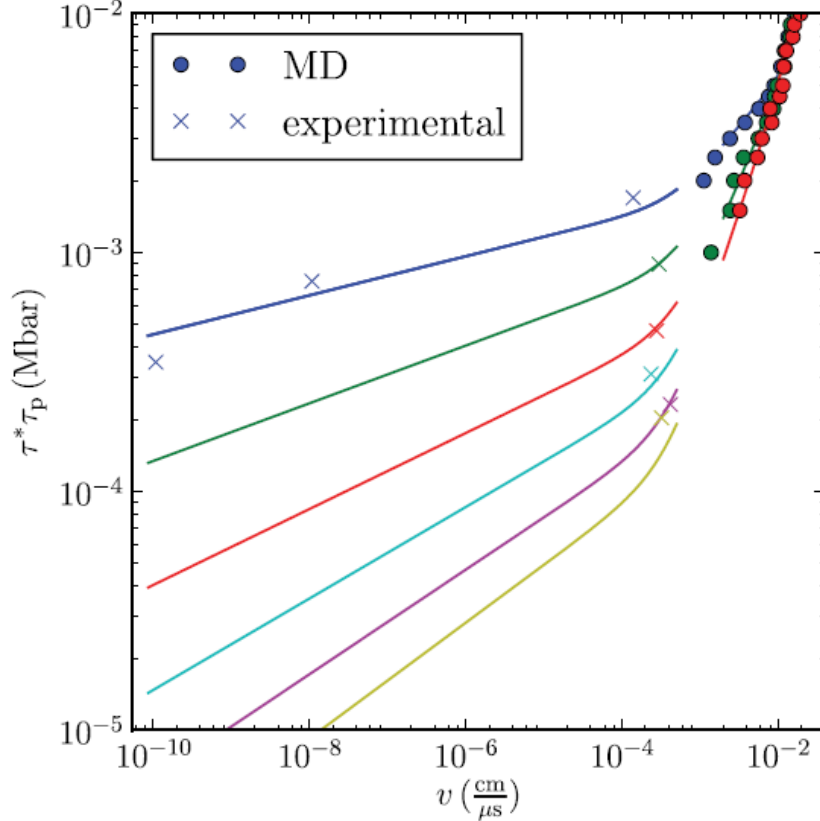


Figure 3 Shear stress response behavior over a wide range of average dislocation velocity. Different color schemes are used for temperatures. For MD data, red, green, and blue line are drawn for temperature of 300 K, 700 K, and 1000 K respectively. And for experimental data, yellow, magenta, cyan, red, green blue lines are correspond to temperatures at 298 K, 473 K, 673 K, 873 K, 1072 K, and 1273 K respectively. This figure was obtained from [11]

1.5 Outline of the Thesis

The history and evolution of MD simulations will be discussed in the chapter 2 along with techniques employed to study material strength under dynamic loading such as quasi-isentropic compression (QIC) and the shock wave propagation in solids. Chapter 3 contains a discussion of

the proxy applications and importance of them for improving performance of well-established MD codes. Performance analysis of a newly implemented neighborlist in the OpenCL version of the MD proxy application CoMD is also included. Our study of strength of Copper and Tantalum is discussed in chapter 4. Finally, chapter 5 contains the future work that is being proposed.

Chapter 2: Quasi-Isentropic Implementation in SPaSM

In this chapter background of method employed in MD code for solving different problems and a new implementation of QIC equations of motion into SPaSM will be discussed.

2.1 MD Solvers

Equation 2.2 is the differential form of the Newton's second law and the same as the equation 2.1. 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, δt must be significantly smaller than the time scale of vibrational period of the atoms.

$$f = ma = -\nabla(u) \quad 2.1$$

$$\dot{v} = \frac{f}{m} \quad 2.2$$

$$\dot{r} = v \quad 2.3$$

The general scheme of MD simulations can be specified stepwise:

- Predict the position, velocities, at time $t + \delta t$, using the values of these quantities and forces at time t .
- 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.

The accuracy and stability of a simulation algorithm is measured by local and global truncation error. Any small perturbation applied at a given time to a classical trajectory will cause it to diverge from its original over the time steps of the calculation.

It is obvious that none of the integration algorithms would generate essentially exact solution over the time steps of the calculation. Fortunately, exact solution of the classical trajectory is not necessary to generate states of the system sampled from ensembles. But, it is important to conserve quantities that are naturally conserved for the given system, such as energy, momentum. A good algorithm allows larger time step to be used while conserving the energy.

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

2.1.1 Verlet Integration

Verlet integration [12] 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 $r(t - \delta t)$ can be stated as in equations in 2.4 and 2.5 respectively.

$$r(t + \delta t) = r(t) + \delta t v(t) + (1/2) \delta t^2 a(t) + \dots \quad 2.4$$

$$r(t - \delta t) = r(t) - \delta t v(t) + (1/2) \delta t^2 a(t) - \dots \quad 2.5$$

Summing equation 2.4 and 2.5, the equation 2.6 can be obtained in which the velocity was eliminated.

$$r(t + \delta t) = 2r(t) - r(t - \delta t) + \delta t^2 a(t) + O(\delta t^4) \quad 2.6$$

The velocities can be obtained using the central difference theorem and can be given as in equation 2.7. 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.

$$v(t) = \frac{r(t + \delta t) - r(t - \delta t)}{2\delta t} \quad 2.7$$

It is important to notice that the local error of the equation 2.6 is of the order of δt^4 while the local error of equation 2.7 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 $v(t)$), which make it time-reversible.

2.1.2 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 2.8 and 2.9 respectively.

$$r(t + \delta t) = r(t) + \delta t \times v\left(t + \frac{1}{2} \delta t\right) \quad 2.8$$

$$v\left(t + \frac{1}{2} \delta t\right) = v\left(t - \frac{1}{2} \delta t\right) + \delta t \times a(t) \quad 2.9$$

Selection of δt is vital in MD as the global error of the result is dominated by δt . Thus, δt should be small enough to snap motion of particles with enough resolution while number of time steps to achieve desired state of the system is kept at an optimal level.

2.2 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 among scientists as a model for calculating atomic-interactions, since that method was well suited for the computing machines available then. Therefore, the inter-atomic interaction in metals is mostly influenced by the local electron density and cannot be realistically described by considering only the pairwise-additive forces [13].

Later, more accurate but simple methods for modeling inter-atomic interactions were introduced for metallic systems taking local electron density into 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) [14]

- Finnis Sinclair [15]
- Glue Model [16]

Among these, EAM and FS are the most popular and widely used methods for atomic interaction models in MD, since they consume the least amount of time while retaining a preferred degree of accuracy.

EAM Model Potential

EAM is a semi-empirical 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. Each atom is influenced by a locally uniform electron gas where total density for N ions in an arbitrary volume V with atomic mass m is given by $\rho = N m/V$. The energy per atom E_i is summed to give total potential energy of the system.

$$E_{tot} = \sum_i E_i \quad 2.10$$

The energy per atom E_i is given by

$$E_i = F(r_i) + \frac{1}{2} \sum_j \phi(r_{ij}), \quad 2.11$$

where $F(r_i)$ 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 neighbor atoms j .

The electron density function (ρ_i) is given by

$$\rho_i = \sum_{i \neq j} f(r_{ij}) \quad 2.12$$

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

2.3 Ensembles

An ensemble is a collection of all microstates of a system that is consistent with constraints 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-isobaric (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.

2.3.1 Microcanonical Ensemble (NVE)

In microcanonical 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.

2.3.2 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 [17] is the most widely used thermostat for NVT ensembles and the one used in SPaSM.

A canonical ensemble uses a thermodynamic friction coefficient ζ that evolves with time in order to minimize the difference between statistical temperature and the instantaneous kinetics of the system. Thus, equation 2.2 is modified to equation 2.13 for α direction.

$$\dot{v}_\alpha = \frac{F_\alpha}{m} - v_T \zeta v_\alpha \quad 2.13$$

Such that

$$\dot{\zeta} = v_T \left(\frac{T}{T_0} - 1 \right) \quad 2.14$$

Where v_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.

2.4 Quasi-Isentropic Compression (QIC)

An alternative approach known as quasi-isentropic compression: a shock-less ramp compression, in which non-steady-state waves are tailored to have a rise not as sharp as in a shock wave resulting in a lower temperature for same compression. Ideal isentropic loading is a state where the entropy remains constant and it is a shock-less thermodynamics state.

In order to simulate such a process, the MD equations of motion are modified to include a strain-rate variable as proposed by Ravelo et al [18].

$$\dot{r}_{\alpha i} = v_{\alpha i} + \dot{\epsilon}_\alpha r_{\alpha i} \quad 2.15$$

$$\dot{v}_{\alpha i} = \frac{F_{\alpha i}}{m_i} - \dot{\epsilon}_{\alpha} v_{\alpha i} \quad 2.16$$

$$\dot{L}_{\alpha} = \dot{\epsilon}_{\alpha} L_{\alpha} \quad 2.17$$

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

Figure 4 shows a set of possible strain rate functions that can be implemented easily.

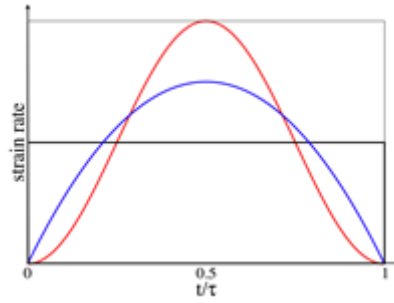


Figure 4 Different strain rate functions that can be used to model strain rate function. Black line represents constant strain rate, blue line represents parabolic strain rate function, and, red line represents strain rate function modeled with $\sin^2 \theta$.

The equations 2.14 - 2.16 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:

$$\dot{E} = \dot{Q} - \dot{W} = \sum_i m_i v_i \dot{v}_i + \sum_{i,j} \frac{\partial U}{\partial r} \dot{r} = \sum_i m_i v_i \dot{v}_i - \sum_i F_i \dot{r}_i \quad 2.18$$

Using equations 2.15 and 2.16 for \dot{r} and \dot{v} ,

$$\dot{E} = \sum_i^N m_i v_i \left(\frac{F_i}{m_i} - \dot{\epsilon} v_i \right) - F_i \dot{r}_i \quad 2.19$$

$$\dot{E} = \sum_i^N v_i F_i - m_i \dot{\epsilon} v_i v_i - F_i \dot{r}_i \quad 2.20$$

By substituting v_i from equation 2.15

$$\dot{E} = \sum_i^N \left[\left(\dot{r}_{\alpha i} - \dot{\epsilon}_{\alpha} r_{\alpha i} \right) F_{\alpha i} - m_i \dot{\epsilon}_{\alpha} v_{\alpha i} v_{\alpha i} - F_{\alpha i} \dot{r}_{\alpha i} \right] \quad 2.21$$

$$\dot{E} = -\dot{\epsilon}_{\alpha} \sum_i^N \left[\left(r_{\alpha i} F_{\alpha i} + m_i (v_{\alpha i})^2 \right) \right] \quad 2.22$$

The stress tensor is given by

$$\sigma_{\alpha\beta} = \frac{\sum_i^N \left[\left(r_{\beta i} F_{\alpha i} + m_i v_{\alpha i} v_{\beta i} \right) \right]}{V} \quad 2.23$$

By substituting stress tensor in equation 2.22

$$\dot{E} = -\dot{\epsilon}_{\alpha} V \sigma_{\alpha\alpha} = -\sigma_{\alpha\alpha} \dot{V} = -\dot{W} \quad 2.24$$

Equation 2.24 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 rise of the temperature and the plastic work done by the dislocation motion (relaxation).

The change of entropy of a system is given by equation 2.25, where dS the change in entropy, T is the absolute temperature and dQ is the amount of heat released or absorbed. However, due to the dissipation of heat due to plastic deformation entropy can be changed in small amounts, which is negligible. Hence, with the QIC (approximately constant entropy) simulations, higher strains, and, strain rates can be achieved at significantly lower temperatures compared to shock wave simulations with the same amount of strain.

$$dS = \frac{dQ}{T}$$

2.25

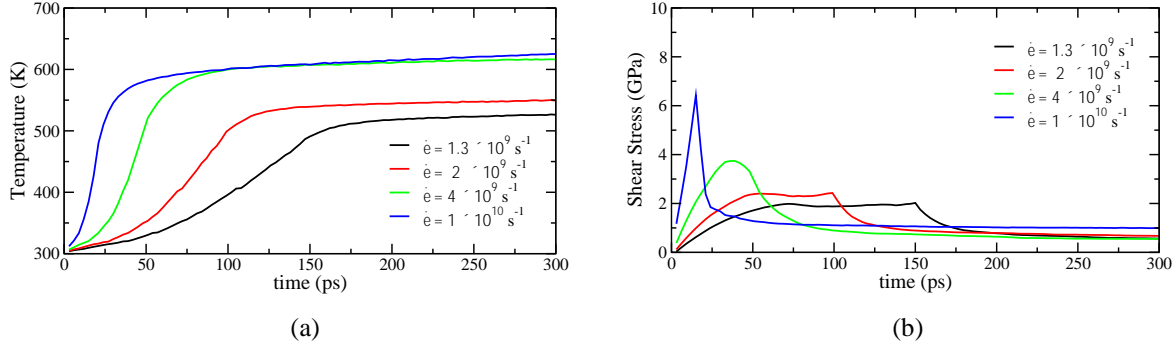


Figure 5 Time profiles of temperature (a) and shear stress (b) obtained from Tantalum QIC simulations at 0.20 strain for different strain rates.

QIC simulations drive the system with the work done on the system by volume change. is different than Hugoniotat in a sense that the Hugoniotat describes the asymptotic state of the material behind the shock whereas quasi-isentropic MD loads the system along an isentrope and results in an increase of the internal energy by the work done on the system (no heat enters or exits the system during the compression). Figure 5 shows time profile of temperature (a) and shear stress (b) obtained from Tantalum QIC simulations at 0.20 strain for different strain rates. In Figure 5 (a), a steep rise in temperature during the compression followed by much slower rise temperature is observed. The slower temperature rise is due to the plastic work done by the moving dislocations. In Figure 5 (b), a sharp peak of shear stress is observed at half way through the compression for high strain rates while lower strain rates don't produce a clear value for maximum shear stress.

2.4.1 Implementation of QIC Equations of Motion in SPaSM

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 2.15 and 2.16 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 2.18 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.

2.5 GPU Execution Model and MD Codes

GPUs are comprised of many stream microprocessors, which contain multiple of processor cores as shown in Figure 6. A block of threads can utilize these processor cores, and shared memory to execute instructions in SIMD (Single Instruction Multiple Data) mode. Further, multiple stream multiprocessors enable the GPU to execute thread blocks from different GPU kernels in parallel. Since GPUs execute enormous number of parallel threads, significant amount of data need to be loaded to the GPUs DRAM for uninterrupted execution of thread blocks. This demands a sophisticated interface between GPU and its DRAM and CPU and system memory. A bidirectional PCIe (Peripheral Component Interconnect Express) is used for the data transferring between CPU and GPU. This data transfer introduces extra overhead in comparison to CPU execution of the program. Thus, for a program to achieve a favorable speed up by executing it on a GPU requires a massive amount of data to be processed by the parallel threads in the GPU. As a result, MD codes are perfect candidates for GPU executions in that perspective.

Not only having the massive number of parallel threads, but, being energy efficient and utilizing space of the chips effectively were also considered in the GPU design. Thus, many of the latency optimization oriented features of the processing units have been removed from GPUs, such

as branch prediction and speculative execution. As a result, programs with a significant amount of diverging branches will perform poorly on GPUs. Performance of the force evaluation kernel of MD codes are affected due to this factor, as the branch that threads in a thread group would take is not guaranteed to be the same.

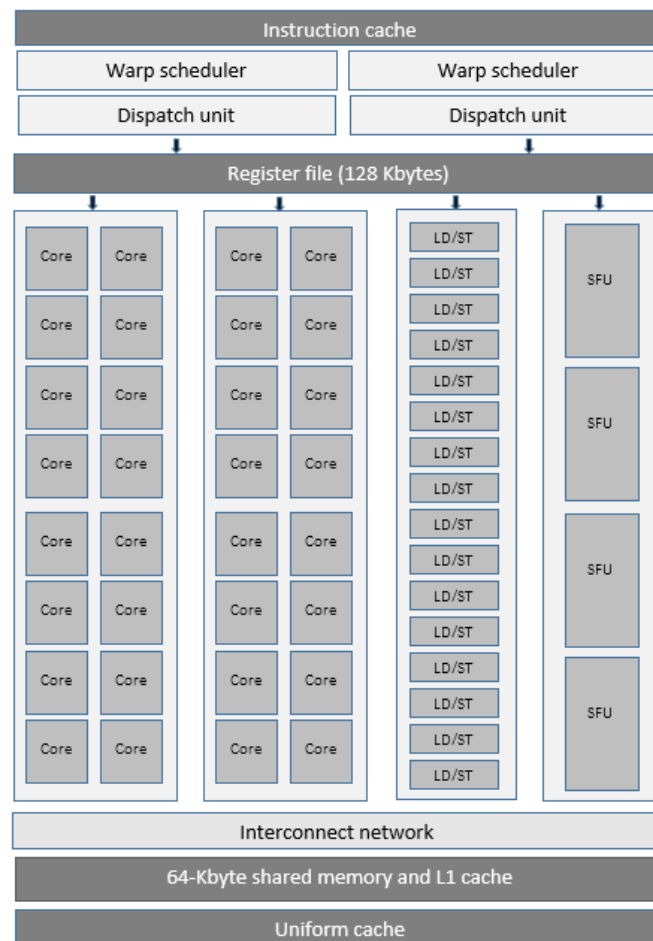


Figure 6 An illustration of a Fermi stream multiprocessor with 32 CUDA cores, 16 load/store units, four special function units, 64kB shared L1 cache, and 128kB register file.

Heterogeneous Computing

CPU and GPU co-processing of applications has become popular since the two architectures have their complementary advantages and disadvantages which allows applications to exploit them to perform better than executing on a single architecture. CPUs were used to

execute serial parts of the application while massively parallel portions are ported to the GPU for optimum throughput. CPU is still the best of the two architecture to execute serial instructions since CPUs are optimized for reducing latency, and executing serial parts in a GPU would be a waste of resources such as memory and energy.

Identifying segments of the code that can effectively utilize a large number of threads in the GPU is important, since the CPU, which handles control flow branching better than GPU, could be the better choice for some algorithms than the device, or the amount of parallelism that could be exploited is not enough to hide the extra overhead when executing on the device.

Unfortunately, almost all the practical MD applications are comprised of segments that are not thread-safe and/or computations that are heavily diverging depending on the data being processed by the threads. Thus, the attention paid for identifying parallelizable components of the applications is important for better performance improvements and preserved accuracy. The speed-up that is achievable from heterogeneous computing depend on the portion of the application that can be executed in parallel and the amount of parallelism available.

Chapter 3: Proxy Applications for Molecular Dynamics

As result of the effort of Department of Energy and ExMatEx program to achieve exascale computational intensity for MD codes, number of different proxy applications was introduced to the scientific community and hardware vendors. These applications serve as a vehicle for the collaboration of scientific application users/developers and hardware vendors in order to implement better algorithms for immerging architectures and adjust new architectures for better execution of scientific algorithms, if it is possible. Co-design Molecular Dynamic (CoMD) is a simple and self-contained proxy application developed for the production level classical MD codes.

3.1 Co-design Molecular Dynamics (CoMD)

The CoMD proxy application represents the typical workload and use cases of MD simulations of material dynamics and represents the fundamental workflow in such simulations from problem setup, equilibration, time integration, and, analysis/visualization. CoMD was mainly based on the production code SPaSM with the intension of the algorithmic improvements and optimizations will eventually be incorporated into the production level classical MD codes such as ddcMD [19], and SPaSM [10]. Further, CoMD has been a favorite among the scientists that study the performance and the energy consumption of MD code on different computer architectures [20], [21].

Latest CoMD version available is 1.1 and, several variants of CoMD has been developed for modern computer architectures and programing models based on a simple serial code. Thus, there are CoMD version for Massage Passing Interface (MPI), OpenMP, Open Computer Language (OpenCL), and, CUDA.

3.1.1 CoMD Data Structures

Domain Decomposition

Parallel execution of CoMD with MPI divide the simulation box among processors equally. Thus, each processor will be assigned a part of the initial lattice (a cube or a cuboid) with equal dimensions in all three directions and compute atom statistics only for the atoms inside the geometry that belongs to the processor. This type of decomposition is called spatial decomposition. Since, spatial decomposition is widely used in classical MD, this decomposition method was introduced into CoMD. An inevitable disadvantage of this method is that the extra amount of communication introduced in the force evaluation steps of the code.

Data Layout

Local domain of the simulation cell that belong to a given processor is then divided in to smaller box called linkcells. The linkcell could be a cube or cuboid depending on the dimension of the local domain. The size of the linkcell in each direction is selected to be close to the cutoff distance in inter atomic potential but greater than it. This method allows to find all the neighbor atoms that contribute for the force evaluation of all the atom in a linkcell can be found within the 26 neighbor linkcells.

Atom details can be layout in computer memory in two different methods; Structure of Arrays (SoA), and, Array of Structure (AoS). Selecting suitable data layout method is mainly depend on the memory access pattern of the underling computer architecture. In SoA method, atoms' unique data fields are stored in separate arrays, for example, x-coordinates of all the atoms in the domain are consecutively stored in a double (or single) precision array etc.

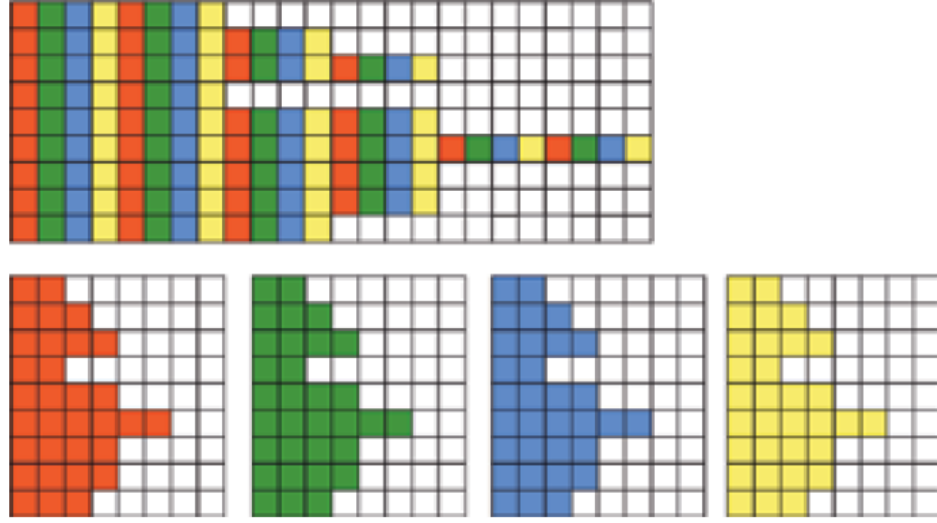


Figure 7 Data layout of Array of Structures (shown in top) vs Structure of Arrays (shown in the bottom) [22].

Interatomic Interactions

EAM and Lenard-Jones (LJ) potential types are currently available to calculate inter atomic interactions in CoMD. When evaluating LJ potential, the functional form of the potential is evaluated. In contrast, values of the functions that are used to calculate EAM potential are stored in tabular format in data files and read into floating point arrays before force evaluation is started.

Values for EAM potential are stored in tabular format in data files and read into floating point arrays before force evaluation is started. Thus EAM evaluation incurs memory accesses into random locations of these arrays, which will increase the cache miss rate and as a result decrease the performance in the EAM kernel.

As a result of evaluating three kernels per atom pair when EAM potentials are used, the evaluation of inter-atomic interactions using EAM potentials occupies approximately 90% of the execution time of a time step. Thus, according to the Amdahl's law (Eq. 3.1, where $S(n)$ is the speed up, P is the portion of the code that can be executed in parallel, and n is amount of parallelism), it is essential to improve the performance of the force evaluation using EAM. Algorithmic improvements, improved data layout methods, and modified EAM potential tables are proven to improve performance of the EAM force evaluation.

$$S(n) = \frac{1}{(1-P) + P/n} \quad 3.1$$

Neighborlists

The Linkcell approach of data layout incurs exhaustive search for neighbor atoms within the 27 neighbor linkcells (including the linkcell that the reference atom belongs to). This search decrease performance of the force evaluation routines as it is performed in three different kernels per time step. Neighborlist is a method of storing all the neighbor atoms of an atom in a dedicated array so that, once the neighborlist is updated and valid it can be used to refer neighbor atoms without searching for them.

When selecting atoms to be in the neighborlist a cutoff distance is used. This cutoff distance needs to be greater than the cutoff distance of the model potential in order to eliminate frequent neighborlist updates due to movements of the atoms. The added distance to the model potential cutoff is called skin-distance.

3.1.2 GPU Versions of CoMD

CoMD has two implementations that can be executed on a GPU, and these implementations are written in OpenCL and CUDA. Since OpenCL codes can be compiled for CPUs as long as the libraries and proper drivers are installed, the OpenCL version of CoMD can be executed even without a GPU.

CUDA Implementation of CoMD

As a part of the Co-Design effort, NVIDIA developed a CoMD version [23] highly optimized for NVIDIA GPUs. 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 [24] 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 neighbor list and potential table value, the code was able to increase the memory throughput.

OpenCL Implementation of CoMD

Similar to the CUDA-CoMD, This version of CoMD was also build upon the basic CoMD 1.1v. However, force evaluation in the OpenCL version wasn't implemented targeting a particular hardware 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). Users are granted with the flexibility to select the storage method of the atom data (either SoA or AoS) in the main memory at the compile time of the code depending on the memory access pattern of the architecture.

3.1.3 CoMD Performance Analysis

MD codes spend enormous amount of their execution time in the atomic force evaluation. In the case of CoMD, 90% of the main-loop time is spent on the EAM force evaluation. According to Amdahl's law, improving EAM force evaluation is important for improving overall performance of the code.

Binary Lattice vs Single Lattice.

Timing 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 8 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 [25]. Thus, the Cu-Ag binary EAM potential is 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 number of operations and the memory accesses in the binary force evaluation. Further, binary lattice simulations had higher number of linkcells compared to the single lattice while the number of atoms per linkcell was less than the single lattice due to increased lattice constant of the binary system. Performance of $L1_2$ force evaluation was then compared against a single lattice of copper using the same EAM potential but a different force kernel, which assumes only one atomic type.

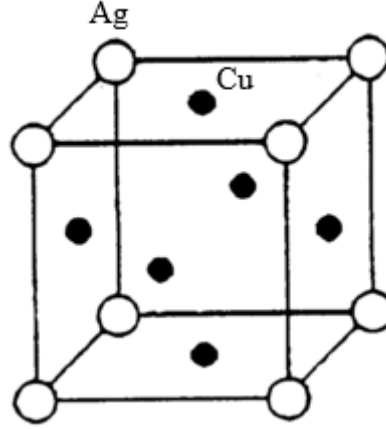


Figure 8 Unit cell of an $L1_2$ structure with Cu atoms in the faces and the Au atoms in the corners of the fcc structure

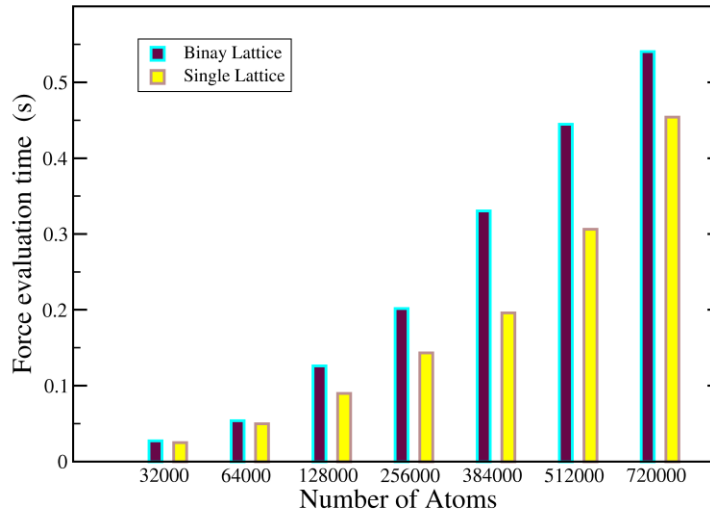


Figure 9 Force evaluation time for binary lattice and single lattice. It is clearly noticeable that the binary lattice spent more time for evaluation of EAM forces due to the increased number of calculation in the binary force evaluation.

Figure 9 and Figure 10 show the force evaluation and execution times of the main loop per iteration respectively. The data was gathered on an Intel Xeon X5355 CPU. The binary lattice spent more time on atomic interaction 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 monoatomic Cu lattice in Co-MD.

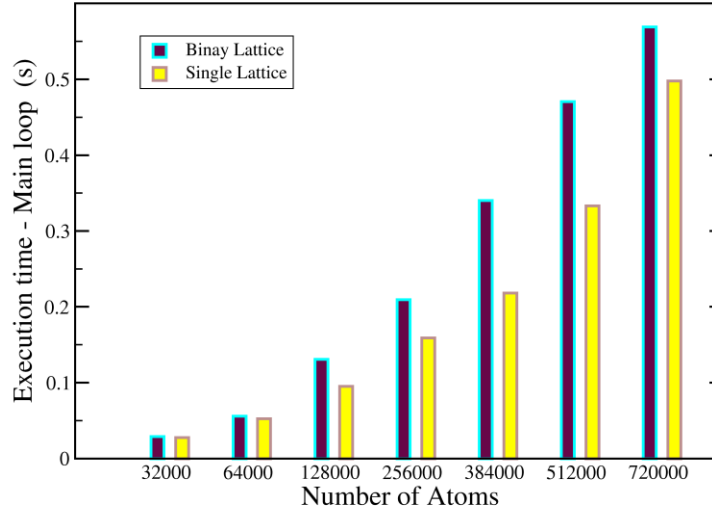


Figure 10 Execution time of the main loop for binary and single lattice for different number of atoms.

Performance Improvements by Implementing Neighborlists for CoMD-OpenCL

The improvements of force evaluation of OpenCL version of CoMD was studied by implementing neighborlists for storing atom neighbors within a given cutoff distance. This cutoff was kept 20% larger than the cutoff distance (skin-distance) of the potential to eliminate frequent updating of the neighborlist. Thus the requirement for updating the neighborlist was set to be an atom moving a half of the skin-distance from where it was at the last neighborlist update.

The performance improvements studied by using NVIDIA Tesla C2070 GPUs. According to Figure 11 and Figure 12 it is clearly noticeable that the neighborlist improved the performance of EAM force evaluation. The speedup of the EAM kernels was approximately 2.

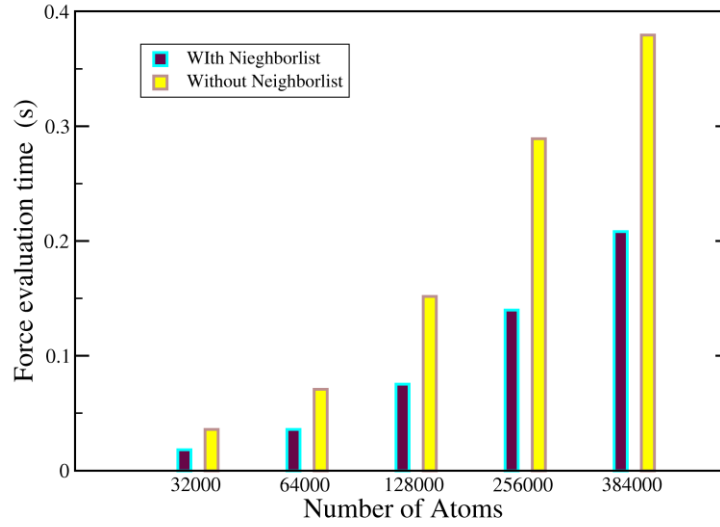


Figure 11 Execution time of EAM force evaluation kernels with and without using neighborlist per iteration for different number of atoms is shown in this graph.

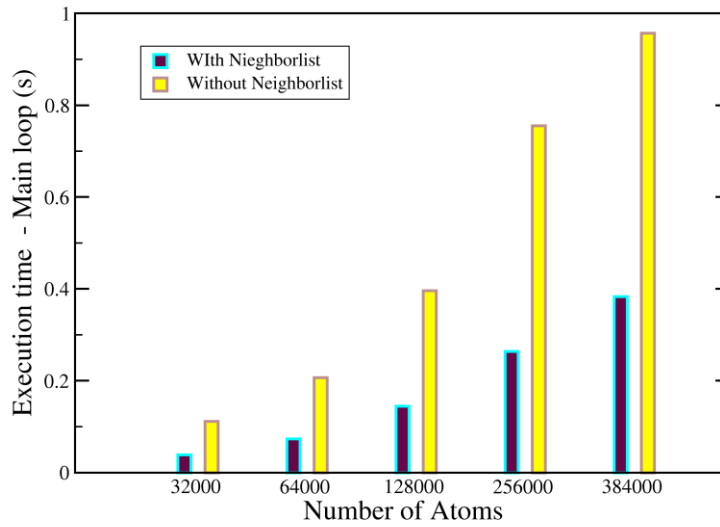


Figure 12 Execution time of the main loop with and without using neighborlist per iteration for different number of atoms is shown in this graph.

Chapter 4: Large-scale MD Study of Material Strength

4.1 Simulation Details and Analysis Tools

Large-Scale MD simulations are used to model quasi-isentropic compression (QIC) in defective copper and tantalum crystals. The atomic interactions were modeled employing embedded-atom method (EAM) potentials (Cu [26], and Ta [27]). Quasi-isentropic uniaxial compression is achieved by incorporating a strain rate function in the positions and velocities equations of motion. In this new formalism the change in internal energy is exactly equal to the work done in compression. Deformation mechanisms and material strength for strain rates in the 10^9 - 10^{12} s⁻¹ range for both Cu and Ta defective crystals were studied using QIC simulations.

For this study perfect Cu and Ta crystals with approximately cubical dimensions with four million atoms were introduced with defects by a preliminary processing scheme. First, uniaxial isentropic compression was performed on the perfect crystal up to 0.10 - 0.12 strain, which is sufficient enough for triggering plastic deformation of the sample crystal. A subsequent annealing process was performed for longer time period (approximately 1ns) to achieve thermal equilibrium of the system. The compression increases the pressure inside the sample. Volumetric-isentropic expansion of the sample was performed to attain zero pressure followed by an annealing at high temperature (800K-900K). The high temperature annealing helps to increase the mobility of the dislocations in the sample, thus the annealing process could be completed within a shorter time frame while achieving preferred final defect densities. Finally, the temperature of the sample is set back to 300K followed by another annealing until thermal equilibrium is achieved. Initial defect densities ρ_0 of the samples were in the order of 10^{12} cm⁻². Periodic boundary conditions were maintained for x , y , and, z directions throughout the above steps.

The defective samples were then compressed either with a QIC simulator for different final strains at different strain-rates or NEMD shock wave simulator at different shock velocities. All the Ta simulations were QICs with final strains of 0.16, 0.20, 0.25, while several data points for Cu were obtained from NEMD simulations. Constant engineering strain-rate function was used as the strain-rate generator for the QIC simulations, therefore, the strain-rate values generated for

each time is time independent. Equation 4.1 represents relationship between true strain rate $\dot{\epsilon}_t$ and engineering strain rate, where L is the length of the simulation box, and t is time.

$$\dot{\epsilon}_t = \frac{\dot{L}}{L} = \frac{\dot{\epsilon}_e}{1 + \dot{\epsilon}_e t} \quad 4.1$$

Crystal Structure Analysis and Visualization

DXA Crystal Analysis Tool

The Dislocation Extraction Algorithm - Crystal Analysis Tool (DAX-CAT) [28], which is developed by Alexander Stukowski, can be used for identifying types of dislocations in a given crystal structure and calculating the length of dislocation lines of identified dislocation types. This tool can be executed in parallel using MPI, which became a useful feature since crystal sizes used in this study were large enough to make the time taken for serial analysis of the structures using the tool intolerable.

The DXA-CAT process a given crystal structure in three principal steps:

- i. The Common Neighbor Analysis (CNA) [29] is performed on the structure to identify crystalline atoms that are not defective.
- ii. A closed, orientable, two-dimensional manifold (referred to as interface mesh) is constructed that separates the crystalline atoms from the disordered or defective atoms.
- iii. For each dislocation segment, an initial Burgers circuit is generated on this manifold. The closed circuit is moved in both directions to the two opposing ends of the dislocation segment to capture its shape. While the circuit is advanced in each direction, a one-dimensional line representing the dislocation segment in the output is constructed.

DXA-CAT can analyze any given configuration of atoms regardless their underlying crystal structures (FCC, BCC, or HCP) due to the CNA performed in the first step.

Visualization

SPaSM is equipped with the extension called MDRender, which is useful for real-time visualization of the structures being simulated, and, it was employed for the visualization of the crystal structures and defects in this study.

$$central_symmetry_i = \left(\sum_j^n r_{ij} \right)^2 \quad 4.2$$

Where r_{ij} is the vector pointing at j^{th} atom from i^{th} atom, n is the number of nearest neighbors for the given structure.

First, MDRender calculates the central-symmetry parameter (equation 4.2) for every atom in the system, and categorizes atoms with central-symmetry value less than five to be perfect (belong to either FCC or BCC structure), and others to be defective (at least one of the neighbor atom is not in the preferred position or missing). After identifying the central-symmetric atoms, Common Neighbor Analysis is performed on those atoms to determine correct crystal structure (either FCC or BCC).

After identifying the correct structure, the second phase of analysis is to determine the local orientation of the atomic structure with respect to a reference direction. This is done by evaluating a three-dimensional orientational order parameter whose components represent the angle of the orientation of the perfect atoms' centrosymmetric nearest neighbors (four or six pairs, depending on being either BCC or FCC structure respectively) with respect to those in ideal (100), (110), and (111) orientations, with respect to a fixed reference direction (generally, this direction is chosen to be the direction of the compression or shock direction, which is z). The normalized triad of the inverse of these distances (d_{100}^{-1} , d_{110}^{-1} , d_{111}^{-1}) are mapped onto an RGB color triangle with the primary

colors red, green, and blue at the corners. Equation (4.2) represents the numerical calculation of this “Orientation Imaging Map” (OIM) for a digital 8-bit color scheme (Figure 13).

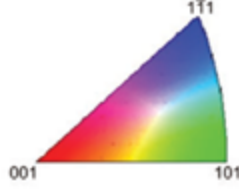


Figure 13 Orientation Imaging Map (OIM) used for both FCC and BCC materials.

$$q_{oim} = (R, G, B) = \left[\min \left(255, \frac{\sqrt{d_{100}^{-1}}}{d_{norm}} \right), \min \left(255, \frac{\sqrt{d_{110}^{-1}}}{d_{norm}} \right), \min \left(255, \frac{\sqrt{d_{111}^{-1}}}{d_{norm}} \right) \right] \quad 4.3$$

Where $d_{norm} = \sqrt{d_{100}^{-1} + d_{110}^{-1} + d_{111}^{-1}}$. If an atom has $d_{100} \ll d_{110}, d_{111}$, it will be considered as oriented in (100) direction and colored with bright saturated red. With this color scheme it is also possible to get intermediate colors. Analysis and visualization of dislocations, grains, and structural changes of crystal structures heavily benefited from this color-coded orientation mapping.

Further, MDRender accepts parameter files as a way of creating the visualization environment, such as angle of rotation of the perspective view and cutoff values for specified variables. These files can be changed to achieve the desired perspective of the images and types of atoms need to be displayed.

4.2 Analysis and Results

The compressed samples were analyzed using the DXA Crystal Analyze Tool with a database of BCC structural defects for Tantalum and a database of FCC structural defects for Copper. Figure 14 shows the time profile of the dislocation density (ρ_D) during and after the QIC

compression. According to the figure ρ_D increases until the compression stops at 50ps and then starts decaying. This is evidence for the relaxation of the shear stress by moving dislocations.

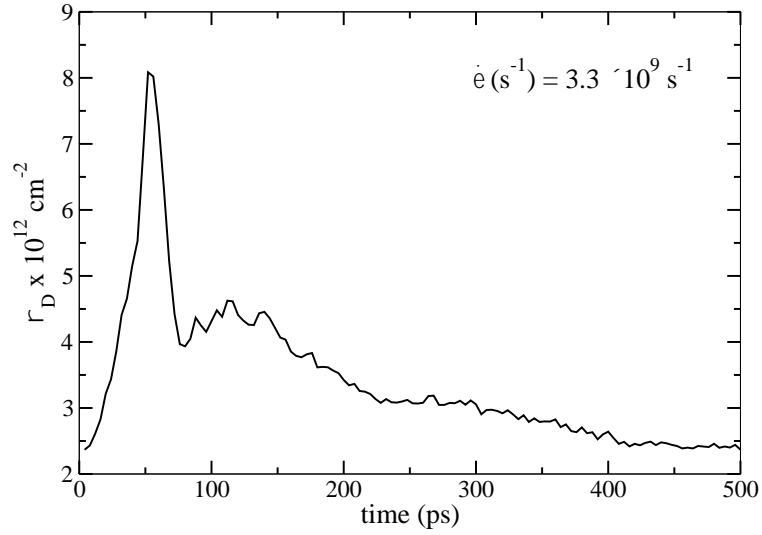


Figure 14 Time profile of dislocation line density for a Ta sample compressed at constant strain rate of $3.3 \times 10^9 \text{ s}^{-1}$.

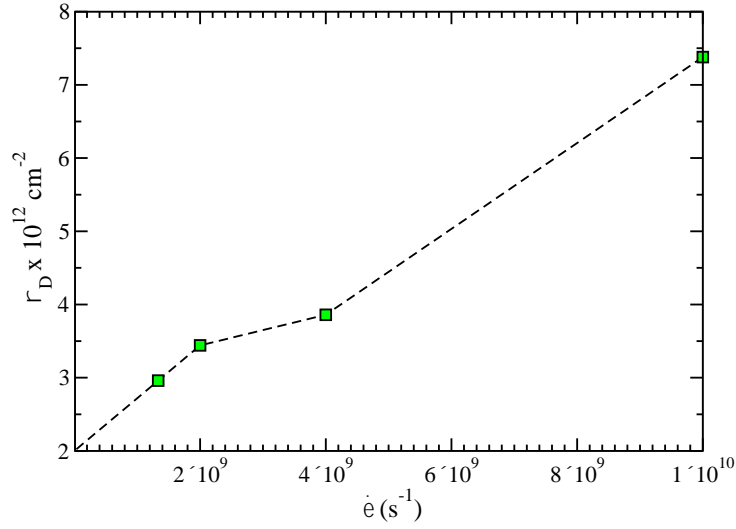


Figure 15 Dislocation line density at 300ps as a function of applied strain-rate for Tantalum QIC simulation at a fixed strain of 0.20.

Figure 15 shows ρ_D as function of applied strain rates for a fixed final strain of 0.20. The figure confirms that ρ_D is proportional to the strain rate. Figure 16 is comprised of the snapshots taken at 250ps for Tantalum QIC simulations along with the corresponding strain-rates and the $\frac{1}{2}\langle 111 \rangle$ dislocation density. Atoms with value of central symmetry greater than 5 are shown in the top row of frames, and color of the atoms correspond to the central symmetry value. Atoms in the bottom row of frame are colored according to the local orientation of atoms with respect to the direction of compression, red being (100) oriented, and blue being (111) oriented.

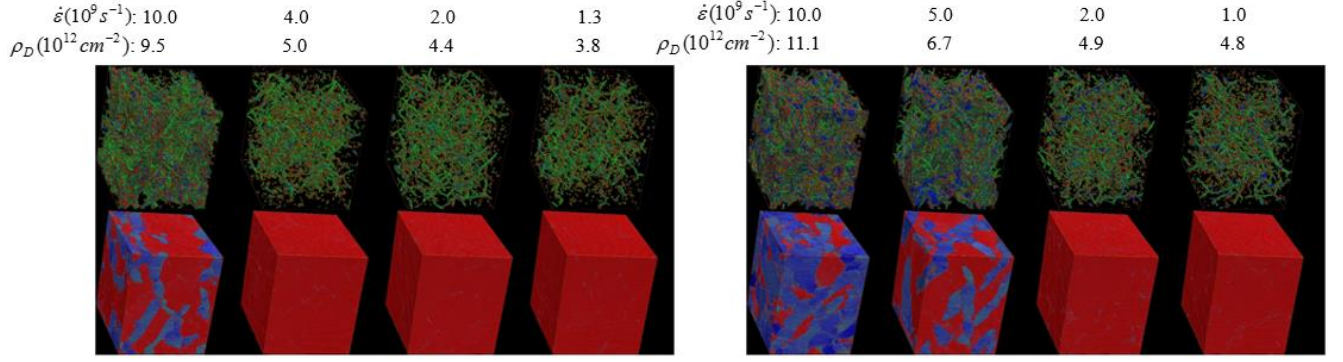


Figure 16 Snapshots at 250ps of Tantalum sample compressed quasi-isentropically to 20% (left), and 25% (right), at different strain-rates: 10^9 - $10^{10} s^{-1}$. Top row of frames shows only defective atoms; bottom row of frames shows only bcc atoms colored according to their local orientation with respect to the compression direction (horizontal axis): red = (100), blue = (111).

Figure 17 shows strain rate dependence for Tantalum QIC data (green squares) along with the PTW model predictions (red line). The difference between QIC data and model prediction is due to QIC data points are presented by the maximum shear stress values while average of shear stress was presented by the model. The black line is a fit to QIC data using a generalized Lorentzian, which is given by equation 4.4.

$$\tau(\epsilon, \dot{\epsilon}) = \tau_0(\epsilon) + \tau_1(\epsilon) \frac{(\dot{\epsilon} t_0)}{\left[1 + (\dot{\epsilon} t_0)^2\right]^{\beta/2}} \quad 4.4$$

Where $\tau_0 = 0.50$ GPa; $\tau_1 = 3.85$ GPa; $t_0 = 269$ ps; $\beta = 0.55$

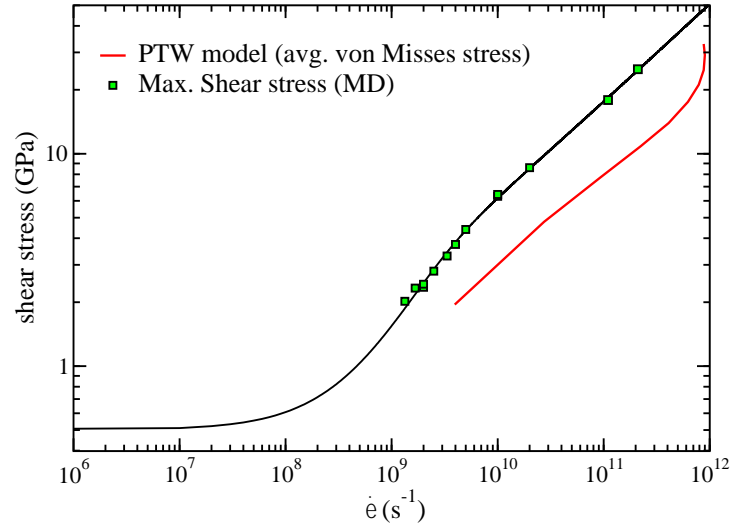


Figure 17 Strain-rate dependence of yield strength of Tantalum from QIC simulations compared with predictions of the PTW continuum model [6] (red line). At high strain rates, the shear stress follows a power law in strain rate with an exponent of 0.45. The solid line is a fit to QIC data using a generalized Lorentzian

Figure 18 shows the MD data for Copper along with the experimental data and the model predictions. The dashed line shows a generalized Lorentzian fit to MD data.

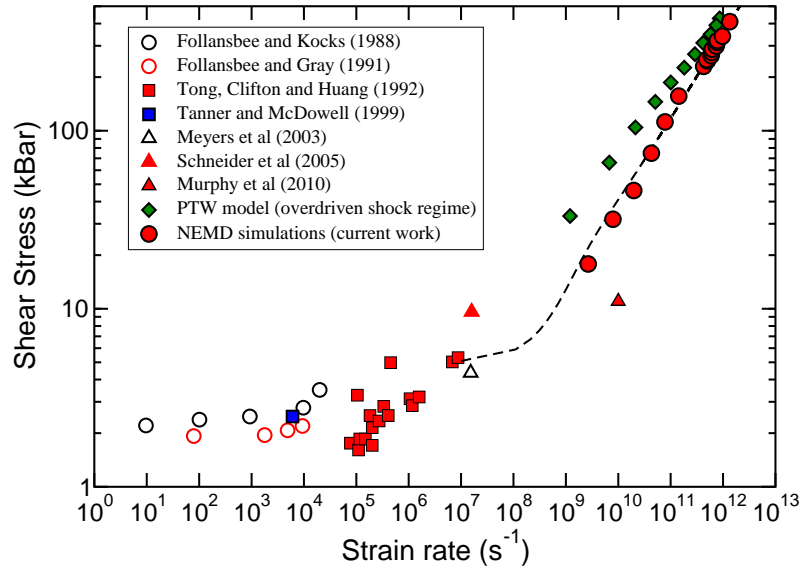


Figure 18 Shear stress response as a function of applied strain rate for Copper. MD data (red circles) are plotted along with experimental data and model predictions. The dashed line is a fit to the MD data.

Chapter 5: Future Work

Strain Rate Dependence of Material Strength Using Polycrystalline Samples

In order to sample lower strain rates ($<10^9 \text{ s}^{-1}$) using QIC simulations, the initial sample defect densities will need to be lowered. In this study strain rates of the order of 10^9 s^{-1} were achieved for samples with defect densities in the range of $10^{12} - 10^{13} \text{ cm}^{-2}$. According to Figure 5 (b) the shear stress response of the material did not reach a clear maximum value. This artifact is a result of high initial defect densities of the sample causing lower strain rates to produce less number of new defects than it requires relaxing the accumulated shear stress. Thus, for lower strain rate QIC simulations samples with very low initial defect densities are essential. In future studies We will be investigating the use of polycrystalline samples to overcome this issue. Figure 19 shows an example of a Tantalum polycrystalline sample.

Desired polycrystalline sample for the future study will accommodate at least five grains for each Cartesian direction, thus approximately 125 grains will be created in the sample. This many grains are necessary due to periodic boundaries applied in the sample. Given a size of a grain is in the order of 20nm, the samples will be approximately $1 \times 10^6 \text{ nm}^3$ in volume. Thus, samples comprising roughly 100millions of atoms will be required.



Figure 19 A polycrystalline sample. This image was obtained from [27].

This study will contribute to better understanding of strain rate dependence of material strength at lower strain rates (10^8s^{-1} and below) that never achieved using MD simulations. Thus, validation of constitutive models in this regime will be possible.

References

- [1] M. P. Allen and D. J. Tildesley, Computer simulation of liquids, Oxford university press, 1989.
- [2] A. Rahman, "Correlations in the motion of atoms in liquid argon," *Physical Review*, vol. 136, no. 2A, p. A405, 1964.
- [3] J. Barojas, D. Levesque and B. Quentrec, "Simulation of diatomic homonuclear liquids," *Physical Review A*, vol. 7, no. 3, p. 1092, 1973.
- [4] "Proxy Apps overview," ExMatEx, [Online]. Available: <http://www.exmatex.org/proxy-over.html>.
- [5] R. Ravelo, Holian, B. L., T. C. Germann and P. S. Lomdahl, "Constant-stress Hugoniot method for following the dynamical evolution of shocked matter," *Physical Review B*, vol. 70, no. 1, p. 014103, 2004.
- [6] F. J. Zerilli and R. W. Armstrong, "The effect of dislocation drag on the stress-strain behavior of F.C.C. metals," *Acta Metallurgica et Materialia*, vol. 40, no. 8, pp. 1803-1808, 1992.
- [7] D. L. Preston, D. L. Tonks and D. C. Wallace, "Model of plastic deformation for extreme loading conditions," *Journal of Applied Physics*, vol. 93, no. 1, pp. 211-220, 2003.
- [8] B. L. Holian, "Modeling shock-wave deformation via molecular dynamics," *Physical Review A*, vol. 37, no. 7, p. 2562, 1988.
- [9] J. B. Maillet, M. Mareschal, L. Souillard, R. Ravelo, P. S. Lomdahl, T. C. Germann and B. L. Holian, "Uniaxial Hugoniot: A method for atomistic simulations of shocked materials," *Physical Review E*, vol. 63, no. 1, p. 016121, 2000.
- [10] P. S. Lomdahl, P. Tamayo, N. Jensen and D. M. Beazley, "50 GFlops molecular dynamics on the Connection Machine-5," in *Supercomputing'93. Proceedings*, 1993.
- [11] N. R. Barton and M. Rhee, "A multiscale strength model for tantalum over an extended range of strain rates," *Journal of Applied Physics*, vol. 114, no. 12, p. 123507, 2013.
- [12] L. Verlet, "Computer experiments on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules," *Physical review*, vol. 159, no. 1, p. 98, 1967.
- [13] B. L. Holian, A. F. Voter, N. J. Wagner, R. J. Ravelo, S. P. Chen, W. G. Hoover, C. G. Hoover, J. E. Hammerberg and T. D. Dontje, "Effects of pairwise versus many-body forces on high-stress plastic deformation," *Physical Review A*, vol. 43, no. 6, p. 2655, 1991.
- [14] M. S. Daw and M. I. Baskes, "Semiempirical, quantum mechanical calculation of hydrogen embrittlement in metals," *Physical review letters*, vol. 50, no. 17, p. 1285, 1983.
- [15] M. W. Finnis and J. E. Sinclair, "A simple empirical N-body potential for transition metals," *Philosophical Magazine A*, vol. 50, no. 1, p. 45, 1984.
- [16] F. Ercolessi, E. Tosatti and M. Parrinello, "Au (100) surface reconstruction," *Physical review letters*, vol. 57, no. 6, p. 719, 1986.
- [17] S. Nosé, "A molecular dynamics method for simulations in the canonical ensemble," *Molecular physics*, vol. 52, no. 2, p. 255, 1984.

- [18] R. Ravelo, B. Holian and T. Germann, "High strain rates effects in quasi-isentropic compression of solids," in *AIP Conference Proceedings*, 2009.
- [19] J. N. Glosli, D. F. Richards, K. J. 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 *ACM/IEEE conference on Supercomputing*, 2007.
- [20] S. J. Pennycook, C. J. Hughes, M. Smelyanskiy and S. A. 2. I. 2. I. Jarvis, "Exploring SIMD for Molecular Dynamics, Using Intel® Xeon® Processors and Intel® Xeon Phi Coprocessors," in *Parallel & Distributed Processing (IPDPS), 2013 IEEE 27th International Symposium*, 2013.
- [21] 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.
- [22] J. Mohd-Yusof, S. Swaminarayan and T. C. Germann, "Co-Design for Molecular Dynamics: An Exascale Proxy Application," 2013. [Online]. Available: http://www.lanl.gov/orgs/adts/publications/science_highlights_2013/docs/Pg88_89.pdf.
- [23] NVIDIA/CoMD-CUDA, [Online]. Available: <https://github.com/NVIDIA/CoMD-CUDA>.
- [24] NVIDIA-CUB, [Online]. Available: <https://nvlabs.github.io/cub/>.
- [25] H. H. Wu and T. D. R., "Cu/Ag EAM potential optimized for heteroepitaxial diffusion from ab initio data," *Computational Materials Science*, vol. 47, no. 2, pp. 577-583, 2009.
- [26] Y. Mishin, M. J. Mehl, D. A. Papaconstantopoulos, A. F. Voter and J. D. Kress, "Structural stability and lattice defects in copper: Ab initio, tight-binding, and embedded-atom calculations," *Physical Review B*, vol. 63, no. 22, p. 224106, 2001.
- [27] R. Ravelo, T. C. Germann, O. Guerrero, Q. An and B. L. Holian, "Shock-induced plasticity in tantalum single crystals: Interatomic potentials and large-scale molecular-dynamics simulations," *Physical Review B*, vol. 88, no. 13, p. 134101, 2013.
- [28] A. Stukowski and K. Albe, "Extracting dislocations and non-dislocation crystal defects from atomistic simulation data," *Modelling and Simulation in Materials Science and Engineering*, vol. 18, no. 8, p. 085001, 2010.
- [29] J. D. Honeycutt and H. C. Andersen, "Molecular dynamics study of melting and freezing of small Lennard-Jones clusters," *Journal of Physical Chemistry*, vol. 91, no. 19, p. 4950, 1987.
- [30] B. L. Holian and G. K. Straub, "Molecular dynamics of shock waves in one-dimensional chains," *Physical Review B*, vol. 18, no. 4, p. 1593, 1978.
- [31] A. Paskin, A. Gohar and G. J. Dienes, "Simulations of shock waves in solids," *Journal of Physics C: Solid State Physics*, vol. 10, no. 19, p. L563, 1977.
- [32] "Proxy Apps Overview," ExMatEx, 2016. [Online]. Available: <http://www.exmatex.org/proxy-over.html>.
- [33] P. S. Follansbee and U. F. Kocks, "A constitutive description of the deformation of copper based on the use of the mechanical threshold stress as an internal state variable," *Acta Metallurgica*, vol. 36, no. 1, pp. 81-93., 1988.
- [34] P. S. Follansbee and G. T. Gray, "Dynamic deformation of shock pretrained copper," *Materials Science and Engineering*, vol. 138, no. 1, pp. 23-31, 1991.

- [35] W. Tong, R. J. Clifton and S. Huang, "Pressure-shear impact investigation of strain rate history effects in oxygen-free high-conductivity copper," *Journal of the Mechanics and Physics of Solids*, vol. 40, no. 6, pp. 1251-1294., 1992.
- [36] A. B. Tanner and D. L. McDowell, "Deformation, temperature and strain rate sequence experiments on OFHC Cu," *International Journal of Plasticity*, vol. 15, no. 4, pp. 375-399, 1999.
- [37] M. A. Meyers, F. Gregori, B. K. Kad, M. S. Schneider, D. H. Kalantar, B. A. Remington, G. Ravichandran, T. Boehly and J. S. Wark, "Laser-induced shock compression of monocrystalline copper: characterization and analysis," *Acta Materialia*, vol. 51, no. 5, pp. 1211-1228, 2003.
- [38] M. S. Schneider, B. Kad, D. H. Kalantar, B. A. Remington, E. Kenik, H. Jarmakani and M. A. Meyers, "Laser shock compression of copper and copper–aluminum alloys," *International journal of impact engineering*, vol. 32, no. 1, pp. 473-507, 2005.
- [39] W. Murphy, A. Higginbotham, G. Kimminau, B. Barbre, E. Bringa, J. Hawreliak, R. Kodama, M. Koenig, W. McBarron, M. Meyers and B. Nagler, "The strength of single crystal copper under uniaxial shock compression at 100 GPa," *Journal of Physics: Condensed Matter*, vol. 22, no. 6, p. 065404, 2010.

Appendix

Source Code for QIC Implementation

Position Update

```
void integrate_adv_coord(void)
{
    int i;
    // pointer for holding particle array
    Particle *pt;
    Real efac1,efac2, edot;

    // check if current time is less than compression time(Tramp) and
    // the system is not being annealed during this timestep
    if (Deformation == 1 && Anneal == 0) {
        // check if compression should be stopped before next
        // time, and if true
        // set variable "Deformation" to 0,
        if ((TotalTime+Dt/2) > Tramp)
            Deformation = 0;
        // calculate strain rate for current timestep
        edot = edotfun(0.5*Dt);
        // calculate and store factors that will be reused
        efac1 = (1.0 + edot * Dt/2)/(1.0 -edot*Dt/2);
        efac2 = Dt/(1.0 -edot * Dt/2);
    }

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

    // update particle positions
    // Maxsize is a global variable that holds the mximum number of
    // particles allowed in a processor
    for (i = 0; i < Maxsize; i++, pt++) {
        pt->r.x = (pt->r.x + Dt * pt->s.x);
        pt->r.y = (pt->r.y + Dt * pt->s.y);

        // check if the system is compressed in this timestep,
        // if true, update
        // positions with applied strain rate, else update normally
        if (Deformation ==1 && Anneal ==0)
            pt->r.z = (efac1 * pt->r.z + efac2 * pt->s.z);
        else
            pt->r.z = (pt->r.z + Dt * pt->s.z);
    }

    // update length of the system in z-direction (direction
    // of compression)
    if (Deformation == 1 && Anneal == 0) {
        Pmin_z *= efac1;
        Pmax_z *= efac1;
    }
}
```

```

    // redistribute particles among cells
    SPaSM_subdivide();
}
}

```

Velocity Update

```

void integrate_adv_velocity(void)
{
    int i;
    // pointer for holding particle array
    Particle *pt;
    Real invmass, vx, vy, vz, ke_old, ke_new;
    Real vfac1, vfac2, edot;

    // check if current time is less than compression time(Tramp) and
    // the system is not being annealed during this timestep
    if (Deformation==1 && Anneal==0) {
        if ((TotalTime+Dt) > Tramp)
            Deformation=0;
        edot=edotfun(Dt);
        vfac1=(1.0 - edot*Dt/2)/(1.0 + edot*Dt/2);
        vfac2=Dt/(1.0 + edot*Dt/2);
    }

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

    if (Anneal == 1) {
        // Zero out velocities of ALL atoms when the TOTAL kinetic energy drops
        ke_old = ke_new = 0.0;
        for (i = 0; i < Maxsize; i++, pt++)
            if (pt->type >= 0) {
                invmass = 1.0 / Masses[pt->type];
                vx = pt->s.x;
                vy = pt->s.y;
                vz = pt->s.z;
                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 = (pt->s.z+Dt*invmass*pt->f.z);
                ke_old += Masses[pt->type] * (vx*vx + vy*vy + vz*vz);
                ke_new += Masses[pt->type] * (pt->s.x * pt->s.x +
                    pt->s.y * pt->s.y +
                    pt->s.z * pt->s.z);
            }

        // barrier before calculating total kinetic energy
        SPaSM_sync_with_nodes();
        ke_old = SPaSM_reduce_double(ke_old, SPaSM_combiner_add);
        ke_new = SPaSM_reduce_double(ke_new, SPaSM_combiner_add);
        if (ke_new < ke_old) {
            pt = (Particle *) Cells[0][0][0].ptr;
            for (i = 0; i < Maxsize; i++, pt++)
                if (pt->type >= 0)

```

```

        pt->s.x = pt->s.y = pt->s.z = 0.0;
    }
}
else { // if not annealing
    for (i = 0; i < Maxsize; 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);

        // if system is being compressed in current time step
        // calculate z-velocity with according to the strain rate
        // applied in current time step
        if (Deformation==1 && Anneal==0){
            pt->s.z = (vfac1 * pt->s.z + vfac2 * invmass*pt->f.z);
        }
        else
            // else calculate z-velocity in normal way
            pt->s.z = (pt->s.z+Dt*invmass*pt->f.z);
    }
}

```

Strain Rate Calculation

```

Real
edotfun(Real deltt)
{
    Real pi = acos(-1);
    Real tt;
    Real erate;

    tt=(TotalTime+delt)/Tramp;

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

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

        // Sin^2 true strain rate
        else if (Isenfun==3) erate=2*edotavg*(sin(pi*tt)*sin(pi*tt));

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

    return erate;
}

```

Curriculum 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, he worked as a research assistant under supervision of Dr. Ramon Ravelo.

Email: mabeywardhana@miners.utep.edu