8-7-2010

A study of grain rotations and void nucleation in aluminum triple junctions using molecular dynamics and crystal plasticity

Matthew William Priddy

A STUDY OF GRAIN ROTATIONS AND VOID NUCLEATION IN ALUMINUM TRIPLE JUNCTIONS USING MOLECULAR DYNAMICS AND CRYSTAL PLASTICITY

By

Matthew William Priddy

A Thesis
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirements
for the Degree of Master of Science
in Civil Engineering
in the Department of Civil and Environmental Engineering

Mississippi State, Mississippi

August 2010

A STUDY OF GRAIN ROTATIONS AND VOID NUCLEATION IN ALUMINUM TRIPLE JUNCTIONS USING MOLECULAR DYNAMICS AND CRYSTAL PLASTICITY

By

Matthew William Priddy

Approved:

Philip M. Gullett Assistant Professor of Civil and Environmental Engineering (Director of Thesis)	Mark F. Horstemeyer Center for Advanced Vehicular Systems Chair and Professor of Mechanical Engineering (Committee Member)
Esteban B. Marin Adjunct Research Professor of Mechanical Engineering (Committee Member)	James L. Martin Professor and Kelly Gene Cook, Sr. Chair (Graduate Coordinator of the Department of Civil and Environmental Engineering)
Sarah A. Rajala Earnest W. and Mary Ann Deavenport, Jr. Chair (Dean of Bagley College of Engineering)	

Name: Matthew William Priddy

Date of Degree: August 7, 2010

Institution: Mississippi State University

Major Field: Civil Engineering

Major Professor: Dr. Philip M. Gullett

Title of Study: A STUDY OF GRAIN ROTATIONS AND VOID NUCLEATION IN ALUMINUM TRIPLE JUNCTIONS USING MOLECULAR DYNAMICS AND CRYSTAL PLASTICITY

Pages in Study: 115

Candidate for Degree of Master of Science

This study focuses on molecular dynamics (MD) simulations, coupled with a discrete mathematical framework, and crystal plasticity (CP) simulations to investigate micro void nucleation and the plastic spin. The origin and historical use of the plastic spin are discussed with particular attention to quantifying the plastic spin at the atomistic scale. Two types of MD simulations are employed: (a) aluminum single crystals undergoing simple shear and (b) aluminum triple junctions (TJ) with varying grain orientations and textures undergoing uniaxial tension. The high-angle grain boundary simulations nucleate micro voids at or around the TJ and the determinant of the deformation gradient shows the ability to predict such events. Crystal plasticity simulations are used to explore the stress-state of the aluminum TJ from uniaxial tension at a higher length scale with results indicating a direct correlation between CP stress-states and the location of micro void nucleation in the MD simulations.

Key words: molecular dynamics, crystal plasticity, plastic spin, continuum theory, multiscale modeling, constitutive modeling

DEDICATION

I would like to dedicate this work to my parents, Richard and Anita Priddy, for their continuous love and unwavering support of the advancement of my education. I am humbled by their constant generosity and sacrifice. They taught me, among many things, to do what you love and love what you do. Thank you for being the best role models any child could ask for.

I would like to thank my wife, Lauren, for her perpetual love and constant support. I would be lost without her beside me. She is my example of excellence that I strive to reach on a daily basis.

I would also like to thank my family and friends for their expression of love and words of support throughout my life. There is no doubt each one of them has touched me in a unique way and aided in my advancement as a person.

ACKNOWLEDGEMENTS

I would like to extend my sincere gratitude to all of the people who helped me through the completion of this thesis. First, I would like to thank Dr. Philip Gullett for taking a chance on me four short summers ago as an undergraduate researcher. I would like to thank him for his constant guidance and encouragement as my advisor. To Dr. Donald Ward, thank you for teaching me the ins and outs of molecular dynamic simulations and answering my many questions. I would like to thank Dr. Mark Horstemeyer and Dr. Esteban Marin for providing their expert knowledge and advice throughout this entire process. I would also like to thank my friends and fellow students Barrett Hardin and Kyle Crosby for their efforts to answer my questions about Abaqus as well as a variety of other subjects.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

FIGURES Page

CHAPTER I

INTRODUCTION

1.1 Motivation

An important part of multiscale modeling, which investigates a material at multiple length scales, is determining mechanisms of failure at the nano-length scale via atomistic simulations such as methods like molecular dynamics (MD). An example of the use of multiscale modeling is the development of ductile, lightweight alloys to replace heavier metals and to lower CO_2 emissions. The use of ductile, lightweight structural alloys such as magnesium is becoming increasingly popular in a variety of industries, with the automobile industry aggressively pursuing their use in components formerly composed of heavier steel or aluminum. Modeling the mechanical behavior of lightweight alloys is challenging due to (i) the complex behavior of the material and (ii) the complication of relating information between multiple length scales.

One of the challenges is linking discrete MD information to continuum information. The difficulty lies in that the discrete quantities (i.e. velocity, position, and force) is different from the continuum quantity (stress and strain). For example, the most commonly used measure of deformation is strain, which takes on a different meaning at the nanoscale. In a continuum body, strain is a

function of the motion. In particular, the function defines strain as the gradient of the displacement. In a discretized body, this definition directly correlates to the change in distance between two atoms, which is empty space. However, even with these complications to overcome, using the continuum framework for a discretized body can be beneficial for visualization purposes and for linking the atomic scale with the macroscale. For example, Gullett *et al.* (2008) used a kinematic algorithm for computing the deformation gradient and strain tensors from the total atomic motion. This is significant because strain tensors are the first step to understanding the full nature of plastic deformation at an atomistic level.

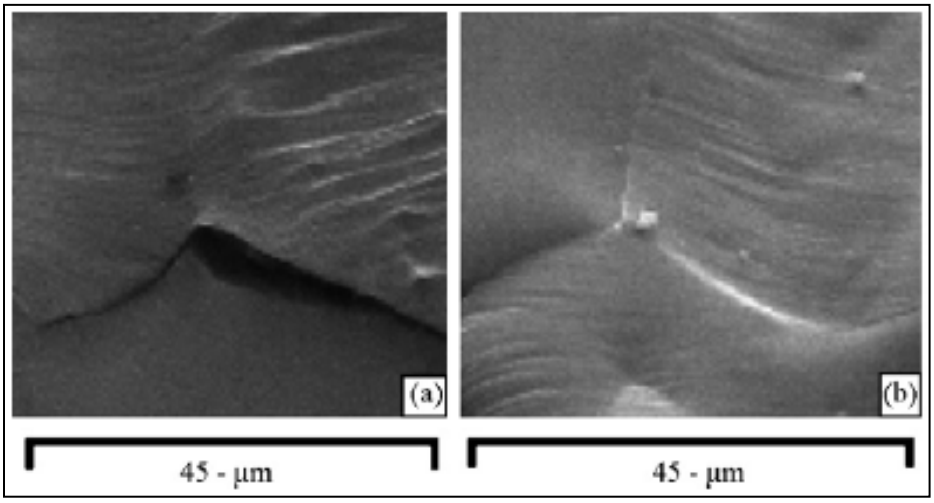

Figure 1.1 Two GBTP junctions where a (a) micro void formed and (b) micro void did not nucleate in a specimen strained to 15% true strain. The tensile axis is vertical with respect to the images (Querin *et al.*, 2007).

One phenomena that is not well understood, examined by Querin *et al.* (2007), involves a grain boundary triple point (GBTP) junction, referred to here as a triple junction (TJ) in AA6022, an aluminum alloy. Querin observed a micro void at the TJ for some orientations while other orientations did not exhibit a micro void, as shown in Figure 1.1. Because nucleation of voids is one of the primary contributors to material behavior, the void nucleation mechanisms need to be better understood to enhance bulk properties of materials.

1.2 Research Objectives

The overall goal of this research is to determine a method for calculating the plastic spin at the atomistic level for the purpose of multiscale-based constitutive modeling.

The pursuit of this goal is made possible with two distinctive objectives of research connected by constitutive modeling. The first objective is (i) to quantify void nucleation in molecular dynamic simulations of TJ using a discrete mathematical framework. Void nucleation is a central part of determining material damage and plays an important role in failure models. The second objective is (ii) to explore the possibility of calculating quantities integral to continuum-based failure models, such as the plastic spin. The plastic spin is an important measure used in constitutive modeling due to its ability to capture the evolution of texture and deformation-induced anisotropy (Horstemeyer, 1995).

1.3 Thesis Organization

The thesis begins with an introduction to and explanation of the research problem. Chapter 2 introduces the advancement of research at the nanoscale, discusses the fundamentals of molecular dynamics simulations, and introduces the kinematics of continuum-based motion. In addition, previous research of the plastic spin and plans for measuring the plastic spin at the nanoscale are discussed in Chapter 2. In Chapter 3, EAMpost, an MD post-processing program which uses a discrete gradient operator to calculate strains and other continuum metrics at the nanoscale, is introduced and validated. Chapter 4 examines the molecular dynamic study of void nucleation in aluminum triple junctions. Chapter 5 introduces crystal plasticity and examines a crystal plasticity study of void nucleation in aluminum triple junctions and their comparison with the MD results in Chapter 4. Finally, conclusions and future work are discussed in Chapter 6.

CHAPTER II

BACKGROUND

The advancement of areas of research such as medicine, electronics, and materials, where the size of products is decreasing to the nanoscale, has provided the scientific community with an array of new research endeavors. The field of nanotechnology is a blend of engineering and science used to advance understanding and materials at length scales that are not visible to the human eye. The increase in interest of the nanoscale is correlated with an increase in computational power and modeling techniques. Liu *et al.* (2004) forsees the advancement research at the nanoscale can have huge implications on areas such as national defense and homeland security. One of the most important traits of a present day researcher is to combine new knowledge with accepted understanding to advance the information in multiple fields. Accordingly, nanotechnology and the advancement of modeling techniques is a great innovation, but it is limited while it stands alone.

2.1 Nanoscale Hurdles

Studies at the nanoscale via atomistic simulations have a multitude of limitations, which include computing power, simulation time, and simulation size

(Buehler, 2004). These limitations severely lessen the number of problems atomistics can be used to solve. There are a couple of reasons computing power is a limiting factor for atomistic simulations. We will see, in Section 2.2, there are many calculations needed to march a single atom through time. As you increase the size of the simulation, the number of calculations made per time step increase. Therefore, computing processors are needed to make the number of calculations and computer memory is needed to store the data from the calculations transmitted to the processors. Another difficult coupled with computing power is storage space for the simulation data. After the completion of the atomistic simulation, data (position, velocity, centrosymmetry parameter, etc.) is saved for each atom at specified time increments. This data can consume large amounts of space depending on the size and duration of the simulation.

The other limiting factors of atomistic simulations are the size and duration of the simulation. As shown in Figure 2.1, the range of molecular dynamics simulations encompasses lengths on the order of microns and time durations on the order of nanoseconds.

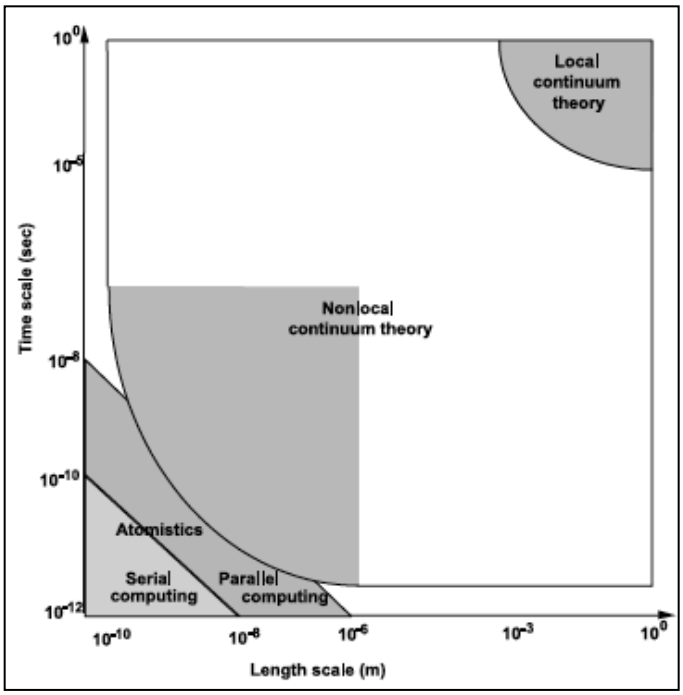

Figure 2.1 Schematic of various theories and the extents of atomistic simulations (Horstemeyer *et al.*, 2001).

For example, simulations of 1 billion atoms would be considered extremely large for current atomistic simulations. However, a system of 1 billion atoms of aluminum is a cube with sides of 0.4 microns (μm) ($1x10^{-6}$ m) in length. Couple the size of this simulation with 1 million time steps of 0.001 picoseconds (ps) ($1x10^{-12}$ sec), and the simulation is 100x smaller than the diameter of a human hair (50 – 100 μm) and lasts for 1 ns, or the amount of time it take light to travel 1 foot (1.017 ns). However, specific physical phenomena can be studied and understood using atomistic simulations as long as these limitations are well understood and taken into account during the analysis process.

2.2 Molecular Dynamics

A dynamical system is defined as a system with the capability of motion in which quantities and dynamic variables have a value at specific instants of time (Logan, 2006). They are governed by the laws of motion, which are functions of time, and satisfy initial conditions. At the root of understanding dynamical systems is molecular dynamics, which utilizes Hamilton's principle and the equations of motion. Hamilton was able to define a dynamical system such that it could be understood over a defined range of time. The dynamical system satisfies a set of differential equations, which are functions of time, and initial conditions. Hamilton's principle was elegant, yet powerful, and the results from it were not fully understood until more recent ventures with the onset of computer simulations.

2.2.1 Background

Molecular dynamic simulations are used to study many-body interactions between atoms and/or molecules. Initial studies with MD occurred in the 1960's with researchers constructing physical experiments and analyzing elaborate systems composed of rubber balls attached with metal rods (Frenkel *et al.*, 2002). As the use of computers became more widespread, researchers turned to simulations for understanding molecular interaction because of their ability to solve large systems of equations quickly.

Steve Plimpton at Sandia National Laboratories created the molecular dynamics software used for this research, Large-scale Atomic/Molecular

Massively Parallel Simulator (LAMMPS) (Plimpton, 1995). Today, LAMMPS has become one of the more popular open-source molecular dynamics code used because of its' ability to run on a plethora of operating systems and ease of customization. Also, molecular dynamics has become a widely used method to analyze and understand large-scale atomic interactions. Its further development is providing researchers with an opportunity to better understand complex material behavior.

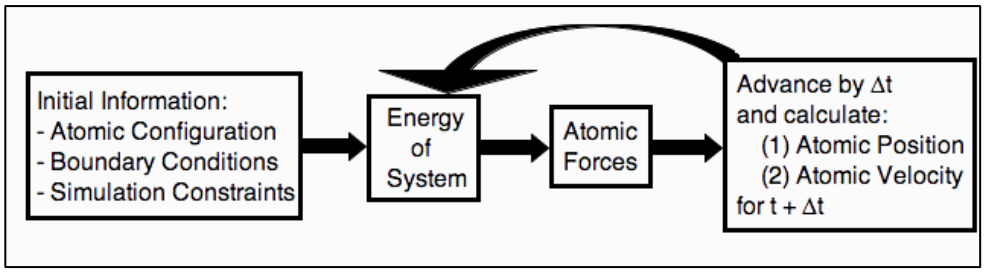

Figure 2.2 Process which MD simulations step through and determine atomic positions and velocities.

LAMMPS calculates the position and velocity of atoms through an explicit process known as the velocity Verlet algorithm (Swope *et al.*, 1982; Verlet, 1967), as shown in Figure 2.2. The initial energy of the system is determined based on simulation characteristics such as the configuration of the atomic structure, boundary conditions of the simulations, and constraints such as temperature. The kinetic energy is determined from the mass of the atoms and their initial velocities while the potential energy is determined from their positions and interaction potentials. With these quantities, the total energy of the system

can be calculated. Given the energy of the system, the forces applied to the atoms can be determined. The forces on the atoms are defined as the derivative of the potential energy function. Then, the forces are applied over a time, Δt, from which new atomic positions and velocities are calculated.

2.2.2 Velocity Verlet Algorithm

The velocity Verlet algorithm determines the velocity and position at time $t + \Delta t$ given the initial position, $\mathbf{x}(t)$, and velocity, $\mathbf{v}(t)$. The position of the atom at $t + \Delta t$ is estimated as

$$\mathbf{x}(t+\Delta t)=\mathbf{x}(t)+\mathbf{v}(t)\Delta t+\tfrac{1}{2}\mathbf{a}(t)(\Delta t)^2, \tag{2.1}$$

and the mid-step velocity is estimated as

$$\mathbf{v}(t+\tfrac{1}{2}\Delta t)=\mathbf{v}(t)+\tfrac{1}{2}\mathbf{a}(t)\Delta t, \tag{2.2}$$

where the acceleration at time t is computed from

$$\mathbf{a}(t)=\frac{F(t)}{m}. \tag{2.3}$$

Given the updated atomic positions, the updated forces, $F(t + \Delta t)$, are found using an interatomic potential and an updated acceleration, $\mathbf{a}(t + \Delta t)$, is calculated using the relationship in Equation 2.3. The new velocity is found using the mid-step velocity and determined to be

$$\mathbf{v}(t+\Delta t)=\mathbf{v}(t+\tfrac{1}{2}\Delta t)+\tfrac{1}{2}\mathbf{a}(t+\Delta t)\Delta t. \tag{2.4}$$

The integration process is repeated for every time step with units of Δt.

2.2.3 Embedded-Atom Method Potential

Molecular dynamic codes use interaction potential models to calculate the energy, force, and stress of a simulation. One of the more commonly used potentials for metals or alloys is the embedded-atom method (EAM) potential. Developed by Daw and Baskes (1984), the EAM potential includes an embedding energy formulation proposed by Friedel (1952) and advanced by Stott and Zaremba (1980) and the pair interaction defined by Jones (1924a, 1924b). The pair interaction is a function of the distance between two atoms and provides a fast and accurate approximation for fully enclosed atoms. However, with the introduction of boundaries or surfaces, the pairwise potential does not provide accurate energy calculation and is better supplanted by a many-body potential, which maintains the speed while providing the increased level of functionality (Gullett *et al.*, 2004). The EAM potential defines the energy for an atom as the sum of embedding energy and the pair potential energy. The total energy, E, for the ith atom is defined as

$$E = \sum_i F_i\left(\sum_{j\neq i}\rho_j\left(r_{ij}\right)\right) + \tfrac{1}{2}\sum_{j\neq i}\phi_{ij}\left(r_{ij}\right), \tag{2.5}$$

where the embedding energy, F, multiplied by the electron density, ρ, due to the neighboring jth atoms is summed with the potential energy term, ϕ, due to the neighboring jth atoms.

2.2.4 Virial Stress

The virial stress tensor (Clausius, 1870; Maxwell, 1870), W, is defined from the energy calculations as

$$W_k^{mn} = \frac{1}{V_i} \sum_{j \neq i}^{N} f_{ij}^m \cdot r_{ij}^n , \tag{2.6}$$

where the total stress tensor is the summation (over N atoms) of the force vector, f, multiplied by the displacement vector, r, for each atom pair, ij, divided by the volume of the ith atom, V. The global stress is the over the continuum is the volumetric average of the virial stress for each atom,

$$\sigma = \frac{1}{N} \sum_{k=1}^{N} W_k . \tag{2.7}$$

Even though the intended use of the virial stress was to relate bulk averages and the values of an arbitrary point and time have no physical meaning, the virial theorem provides the most consistent expression for relating forces and motion within an atomic system to a continuum stress. (J. A. Zimmerman *et al.*, 2002).

2.2.5 Centrosymmetry Parameter

Another important measure widely used in MD simulations of metals is the centrosymmetry parameter (Kelchner *et al.*, 1998). All metals have a repeating lattice structure and the centrosymmetry parameter measures the deviation from the lattice structure. For face-center cubic (FCC) materials, the centrosymmetry parameter is expressed as

$$C_i^{FCC} = \sum_{j=1}^{6} \left| r_{i,j} + r_{i,j+6} \right|^2 , \tag{2.8}$$

where the centrosymmetry parameter for the ith atom is the sum of the squares of pairs of opposite nearest neighbors, twelve atoms total and six pairs. For an atom in a perfect FCC structure, the centrosymmetry value is equal to zero. As the centrosymmetry value increases, so does the deformation of the lattice.

2.3 Continuum Deformation Measures

The continuum description used throughout this text distinguishes the reference and current configuration. A body is said to be in a reference configuration at time $t=t_0$ and a point on the body is identified by a vector $\mathbf{X}=(X_1, X_2, X_3)$. As time passes and motion occurs, the body moves to a current configuration and a point on the body is identified by a new vector $\mathbf{x}=(x_1, x_2, x_3)$. The reference and current configuration can be uniquely mapped by a time-dependent motion function, known as χ and expressed as

$$\mathbf{x} = \chi(\mathbf{X},t). \tag{2.9}$$

The spatial gradient of the motion, known as the deformation gradient (Holzapfel, 2000), is expressed as

$$\mathbf{F} = \frac{\partial \chi}{\partial \mathbf{X}} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}}. \tag{2.10}$$

The deformation gradient can be used to define the rotation and stretch effects about a point. The displacement field in the reference configuration, expressed as

$$\mathbf{U}(\mathbf{X},t)=\mathbf{x}(\mathbf{X},t)-\mathbf{X}, \tag{2.11}$$

is the difference between the original position, $\mathbf{X}$, in the undeformed configuration and the new position, $\mathbf{x}$, in the deformed configuration. The displacement field can also be written in terms of the current configuration, which is expressed as

$$\mathbf{u}(\mathbf{x},t)=\mathbf{x}-\mathbf{X}(\mathbf{x},t). \tag{2.12}$$

It can be shown that $\mathbf{U}(\mathbf{X},t) = \mathbf{u}(\mathbf{x},t)$, but for brevity, we will just mention it is true. Therefore, the displacement fields can be written in terms of the reference or current configuration and they are equal. Taking the time derivative of the material configuration displacement field yields

$$\mathbf{V}(\mathbf{X},t)=\frac{\partial \mathbf{U}(\mathbf{X},t)}{\partial t}, \tag{2.13}$$

but by using the above statement of $\mathbf{U}(\mathbf{X},t) = \mathbf{u}(\mathbf{x},t)$, then Equation 2.13 can be rewritten as

$$\mathbf{v}(\mathbf{X},t)=\frac{\partial \mathbf{u}(\mathbf{x},t)}{\partial t}. \tag{2.14}$$

Defining the velocity gradient in the current configuration provides much more flexibility for subsequent derivations, as shown below. The velocity gradient, $\mathbf{L}$, is

$$\mathbf{L}=\frac{\partial \mathbf{v}}{\partial \mathbf{x}}=\frac{\partial \dot{\chi}(\mathbf{X},t)}{\partial \mathbf{X}}\frac{\partial \mathbf{X}}{\partial \mathbf{x}}, \tag{2.15}$$

or simply expressed as

$$\mathbf{L}=\dot{\mathbf{F}}\mathbf{F}^{-1}, \tag{2.16}$$

where the velocity gradient is the product of the material time derivative of the deformation gradient with the deformation gradient. The velocity gradient can be decomposed into a symmetric tensor, $\mathbf{d}$, and an anti-symmetric tensor, $\mathbf{w}$, and expressed as

$$\mathbf{L} = \mathbf{d} + \mathbf{w} . \tag{2.17}$$

The symmetric tensor is commonly referred to as the stretch tensor, written as

$$\mathbf{d} = \frac{1}{2}\left(\mathbf{L} + \mathbf{L}^{\mathrm{T}}\right), \tag{2.18}$$

and the anti-symmetric tensor is the spin tensor, expressed as

$$\mathbf{w} = \frac{1}{2}\left(\mathbf{L} - \mathbf{L}^{\mathrm{T}}\right). \tag{2.19}$$

Calculations of strain are very important in engineering applications, because they can typically be used across length scales to describe the deformation of a body. Two common strain measures (Ogden, 1984), the Lagrangian-Green (or material) strain, expressed as

$$\mathbf{E} = \frac{1}{2}\left(\mathbf{F}^{\mathrm{T}}\mathbf{F} - \mathbf{I}\right), \tag{2.20}$$

and the Eulerian (or spatial) strain, expressed as

$$\mathbf{e} = \frac{1}{2}\left(\mathbf{I} - \mathbf{F}^{-\mathrm{T}}\mathbf{F}^{-1}\right). \tag{2.21}$$

2.4 Material Response Modeling

The mathematical description of continuum behavior is fundamentally nonlinear in both geometry and material properties. For this reason, analytic

studies generally focused on linearized geometry and isotropic linear-elastic material behavior. The theory of elasticity states the deformation of solid materials is induced by the application of mechanical and thermal forces (Barber, 2002). One of the restrictions of elasticity is the assumption of perfectly elastic loading and the small strain assumption. Real materials are not perfectly elastic under all loading conditions nor do they only undergo small strains. However, before the advanced computational power provided by computers, little was understood of material behavior beyond the elastic region.

The onset of computer simulations has created an upsurge of research on material responses past the elastic region. Computers have allowed us to pursue the understanding of materials with highly nonlinear behaviors such as polymers and biological materials. For example, time and temperature dependent plasticity models were originally created to model materials undergoing large strains (Bammann, 1984). Models such as this were the beginning of constitutive models known as internal state variable (ISV) models. ISV models use stress-strain data from mechanical testing (uniaxial tension, compression, and torsion) to determine constants to fit constitutive equations in an effort to match computational solutions with experimental data to predict a material's behavior under certain loading conditions.

Today, such models have been expanded to include rate dependence and recovery mechanisms (Marin *et al.*, 2006). Also included in this model is a porosity-based isotropic damage variable to model the ductile failure mechanisms due to the nucleation, growth, and coalescence of micro voids.

These models lend themselves to multiscale modeling because of the multiple length scales involved in material failure. There are two types of approaches for multiscale modeling: (1) the hierarchical approach and (2) the concurrent approach. The hierarchical approach determines causality at lower length scales and integrates them into macroscale models (Horstemeyer *et al.*, 2003). In contrast, the concurrent approach includes multiple length scales in a single simulation (Tadmore, 1996).

For an example of the use of multiscale modeling, McDowell (2000) states plasticity occurs over an expansive collection of length scales and is affected by dislocation generation and kinetics. Therefore, the models used to simulate plastic deformation should also involve multiple length scales. This is important for the research discussed here because the study of void nucleation at the atomistic level is an important aspect of the multiscale models.

2.4.1 Work Towards Plastic Spin

Plastic deformation of crystalline materials is related to changes in the crystal lattice such as dislocation glide or twinning that lead to changes in the texture (Bunge *et al.*, 1997). Bunge also stated that glide and rotation occur simultaneously in a polycrystalline material so continuity of the plastic deformation is maintained over the grain boundaries. The total of all crystal orientations is called the texture of the material and the rotation of grains leads to texture changes also referred to as deformation texture. Texture is important because it can lead to anisotropy in many of a material's physical properties.

Within the theoretical framework of continuum mechanics, one of the quantities commonly associated with texture evolution is the plastic spin. It is an important quantity in constitutive modeling because of its ability for analyzing texture and deformation-induced anisotropy in macroscale unified creep-plasticity and mesoscale polycrystalline plasticity models (Horstemeyer, 1995).

2.4.2 Deformation Gradient Decomposition

For elasto-plastic modeling at finite deformations, the total deformation gradient is commonly written as the product of two components (Bilby *et al.*, 1957; Kröner, 1959; Lee, 1969). It is broken up into a reference configuration, labeled Ω_o, an unstressed intermediate configuration, labeled Ω_i, and the current configuration, labeled Ω_c. The reference configuration is mapped to the intermediate configuration with the plastic deformation gradient, $\mathbf{F}_p$. and the intermediate configuration is mapped to the current configuration with the elastic deformation gradient, $\mathbf{F}_e$. This is illustrated in Figure 2.3 and expressed as

$$\mathbf{F}=\mathbf{F}_e\mathbf{F}_p \,. \tag{2.22}$$

The multiplicative decomposition of the deformation gradient is motivated by the kinematics of single crystals where dislocations move along fixed slip systems through the crystal lattice (Steinmann, 1996). Steinmann also stated the intermediate configuration is incompatible because some dislocations do not completely transverse the crystal resulting in plastic deformation that is not uniform.

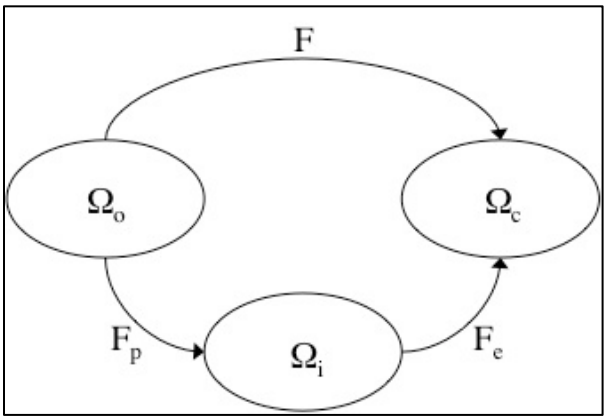

Figure 2.3 Diagram drawing of the decomposition of the deformation gradient into elastic and plastic parts.

The deformation gradient can be decomposed into a variety of ways, therefore it must follow that values in terms of the deformation gradient also have the ability to be decomposed. Mandel (1973) first proposed the decomposition of the spin tensor into elastic and plastic rotations. This decomposition provides an intermediate, unstressed configuration where the elastic deformation has been removed, as illustrated by Ω_i in the figure above. However, the unstressed configuration is not uniquely defined because an arbitrary rigid-body rotation can be superimposed on the configuration and it stays unstressed (Aravas 1994). Mandel (1971) also proposed a local triad of direction vectors, which are embedded in the material substructure, to relate the orientation of the intermediate configuration to a fixed cartesian coordinate system.

The plastic rotation proposed by Mandel (1973) is also known as the plastic spin. Defined as the anti-symmetric portion of the plastic velocity gradient, the plastic spin is directly related to texture. Specification of this

quantity from the continuum level provides no clean path of evolution because it inherently lives in an intermediate configuration. Quantifying the plastic spin would allow researchers to investigate grain rotations, which play a significant role in large deformations. For example, Horstemeyer *et al.* (2002) used experimental data and MD to illustrate the plastic spin played a role in dislocation nucleation when torsion was applied to single crystal copper. In the past 20 years, many bright and inquisitive minds have determined formulations for the plastic spin under certain loading conditions and material behaviors. Therefore, a discussion has been created recently in the literature in regards to the plastic spin, its true meaning, and possible formulations. However, in these discussions, some ambiguity associated with term 'plastic spin' and its qualitative meaning has surfaced. The next section is an attempt to clear up these ambiguities and what it means for the determination of the plastic spin at the atomistic level.

2.4.3 Dafalias and the Plastic Spin

Dafalias (1998) defines the antisymmetric portion of the plastic velocity gradient as the 'plastic material spin', but the plastic spin refers to the addition of the plastic material spin and the constitutive spin. The "plastic spin", as coined by Dafalias (1984, 1985), is the rotation rate of a reference system. This total rotation rate can be decomposed into the antisymmetric portion of the velocity gradient (flow-induced spin) and the relative rotation rate (texture spin) as considered by Teodosiu (1989) and Lippmann (1995).

Following Mandel (1971, 1973) and Kratochvil (1971, 1973), Dafalias (1984) chose direction vectors that coincide with a fixed orientation relative to the global coordinate system, making them an isoclinic configuration. After working through the continuum mechanics done by Aravas (1994), one will find the plastic spin, defined as a purely plastic quantity (in the intermediate configuration), is

$$W_i^p = \tfrac{1}{2}\left(\dot{F}^p\left(F^p\right)^{-1} - \left(F^p\right)^{-\mathrm{T}}\left(\dot{F}^p\right)^{\mathrm{T}}\right). \tag{2.23}$$

He, like Mandel and Kratochvil, believed constitutive relations in a macroscopic formulation must include the plastic rate of deformation and the plastic spin. Dafalias introduced a constitutive equation for the plastic spin to obtain an appropriate corotational rate for kinematic hardening (1983). This additional equation was found using the representation theorem, which defines the stress by a response function that is material frame-indifferent (Holzapfel, 2000). The importance of this equation is it does not include the evolution of other variables; it is self-containing. However, studies in the 1980's were limited to isotropic materials and did not expand to anisotropic materials until later. Mandel (1971) and Kratochvil (1973) found the plastic spin is identically zero for the deformation of isotropic materials. At this time, the majority of macroscopic plasticity theories did not explicitly state the plastic spin despite its significance in microscopic formulations (Dafalias *et al.*, 1990).

Equation 2.23 led Dafalias to the conclusion that the plastic spin is equal to the continuum spin in the intermediate configuration because the substructure spin is zero due to the definition of the direction vectors not varying with time.

For a rigid plastic material, Dafalias clearly defines the plastic spin as a function of the continuum spin, W, and the substructure spin, ω as illustrated by

$$W^p = W - \omega \,. \tag{2.24}$$

By rewriting Equation 2.24, the spin of the substructure can be found in continuum models provided the continuum spin is known and a constitutive equation for the plastic spin is given. The spin of the substructure can be used to determine the evolution of anisotropy.

2.4.4 Others and the Plastic Spin

Other camps of understanding the plastic spin exist and came to much different conclusions in regards to the creation of constitutive equations. Schieck *et al.* (1995) chronologically lists the progress of the plastic spin calculation and the ambiguity associated with the quantity and its worth to constitutive modeling. For example, Nemat-Nasser (1990) states the plastic spin is not an independent quantity and as such does not need to be explicitly defined in constitutive modeling. Nemat-Nasser provided an alternate kinematic function for the plastic spin defined in terms of the stretch rate, which was also supported by Stumpf (1990). Nemat-Nasser came to this conclusion by defining the deformation gradient with the decomposition laid out be Lee (1969) and then using the polar decomposition of the elastic and plastic deformation gradients in his theory. His expression is

$$\begin{aligned}\left(I^p II^p - III^p\right)\hat{W}^p &= I^{p^2}\left(U^p\hat{D}^p - \hat{D}^pU^p\right)\\ -I^p\left(U^{p^2}\hat{D}^p - \hat{D}^pU^{p^2}\right) &+ U^p\left(U^p\hat{D}^p - \hat{D}^pU^p\right)U^p\end{aligned}, \tag{2.25}$$

where $\hat{W}^p$ is the plastic spin, $\hat{D}^p$ is the plastic deformation rate, U^p is the total plastic stretch tensor, and I^p, II^p, and III^p are the basic invariants of U^p (Nemat-Nasser, 1990).

Also, Kröner (1981) states that during elastic deformation the lattice stretches and contracts while neighboring particles remain neighboring particles and during plastic deformation the lattice structure is not changed. For clarity, the body referenced in Figure 2.4 is made up of many infinitesimal elements, as shown in Figure 2.4(a). When the body is plastically loaded, the intermediate state of the elements is shown in Figure 2.4(b). However, the elements are not compatible with neighboring elements, so an additional deformation must take place for the final state of the body to occur in Figure 2.4(c). Therefore, Kroner's theory suggests there are two elastic deformations present: (1) an internal elastic deformation that is equal and opposite to the plastic deformation to make the elements compatible and (2) the elastic deformation that relaxes the system to a low energy configuration. Schieck *et al.* (1995) also discussed the decomposition of the deformation gradient and the various configurations associated with the decomposition. He concluded it is not possible to determine, uniquely, the stress-free intermediate configuration.

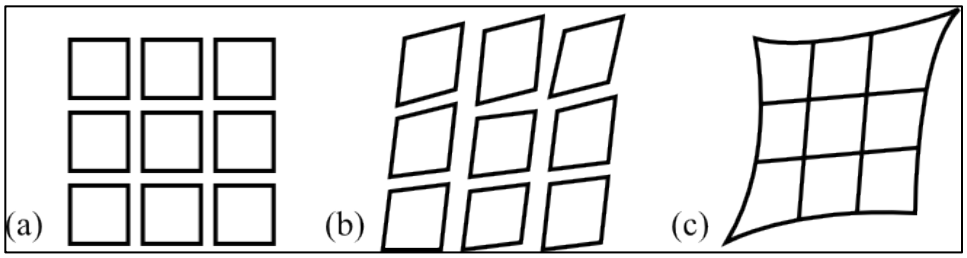

Figure 2.4 A body comprised of elements (a) before deformation, (b) after deformation but separated, and (c) after deformation and joined together (Kröner, 1981).

Kroner's findings are supported by Prantil *et al.* (1993), who discusses the creation of a constitutive model for planar polycrystal where, during typical metal forming operations, plastic deformation takes place primarily by crystallographic slip (Kocks, 1975). But, for up to moderately large strains, the reorientation of crystals is the dominant factor affecting the developing anisotropy (Prantil *et al.*, 1993). The distribution of the reorientation of crystals is commonly referred to as texture. Prantil assumes a homogeneous deformation throughout the body to ensure compatibility, which allows the deformation gradient of each crystal to be equal to the deformation gradient of the body. However, it also violates the intergranular equilibrium, meaning if identical deformation occurred through all of the crystals, they would not fit together after deformation (Taylor, 1938).

The study of the plastic spin by various researchers has led to a plethora of statements and equations to relate the plastic spin to anisotropy and texture. However, none of those studies have explicitly discussed the application of the plastic spin at the nanoscale, which creates additional complexity.

2.4.5 Intermediate Configuration in Atomistics

In all of the plastic spin formulations previously mentioned, the plastic spin was determined for the intermediate or stress-free configuration. One of the major complications of determining the plastic spin at the atomistic level is to determine a stress-free configuration. In order to investigate the nature of the plastic spin we proposed developing a series of molecular dynamic (MD) simulations and examining the associated deformations.

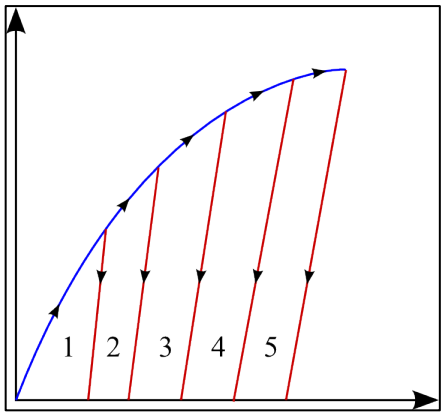

Figure 2.5 Stress-strain curve that depicts the projected load-unload path for determination of the plastic spin.

The proposed method of determining the plastic spin, $\mathbf{w}_p$, is as follows. The specimen will be incrementally loaded, indicated by the blue line, and then fully unloaded, indicated by the red lines. A collection of load-unload increment curves will be collected to make a complete stress-strain curve, as illustrated in Figure 2.5. The loading of the specimen is comprised of elastic and plastic deformation. With loading the specimen, we can calculate the total deformation

gradient, $\mathbf{F}$, which leads to the determination of the total velocity gradient expressed in Equation 2.16. To determine the total velocity gradient, the material time derivative of the deformation gradient is also necessary. This value can be approximated in a couple of different ways, one being a first-order linear approximation of the deformation gradient (Tucker *et al.*, 2010), expressed as

$$\dot{\mathbf{F}}=\frac{\mathbf{F}_{\text{current}}-\mathbf{F}_{\text{past}}}{t_{\text{current}}-t_{\text{past}}}=\frac{\Delta\mathbf{F}}{\Delta t}. \tag{2.26}$$

The decomposition of the velocity gradient is expressed as

$$\mathbf{L}=\mathbf{L}_e+\mathbf{L}_p. \tag{2.27}$$

When the specimen is unloaded, only elastic deformation is removed, thus leaving a sample with only plastic deformation present. During unloading, the elastic deformation gradient would be removed and the plastic deformation gradient can be found. The elastic deformation gradient can be found given $\mathbf{F}$ and $\mathbf{F}_p$ along with Equation 2.22. With all the deformation gradients accounted for, the decomposed velocity gradients can be determined. The expression for the elastic velocity gradient, $\mathbf{L}_e$, is

$$\mathbf{L}_e=\dot{\mathbf{F}}_e\mathbf{F}_e^{-1}, \tag{2.28}$$

and the expression for the plastic velocity gradient, $\mathbf{L}_p$, is

$$\mathbf{L}_p=\mathbf{F}_e\dot{\mathbf{F}}_p\mathbf{F}_p^{-1}\mathbf{F}_e^{-1}. \tag{2.29}$$

Then, the anti-symmetric portion of plastic velocity gradient, $\mathbf{w}_p$, can be found using an expression similar to Equation 2.19 which is

$$\mathbf{L}_p=\mathbf{d}_p+\mathbf{w}_p. \tag{2.30}$$

However, there are difficulties with this plan. The main obstacle is the ability to reach the intermediate configuration by simply unloading the atomistic simulation to a relaxed state and the validity of the proposed plan. It is clearly shown in earlier sections the elastic and plastic deformations are applied simultaneously, thus making the intermediate configuration a fictitious configuration. For example, separately $\mathbf{F}_e$ and $\mathbf{F}_p$ are incompatible fields. But, join them together to get the total deformation gradient, and they obey the compatibility equation, $\mathrm{F} \times \nabla = 0$. Zimmerman's (2009) recent work studies nanoscale thin films of copper to determine the deformation gradient and the curl of the deformation gradient. The film was biaxially loaded and unloaded, similar to the proposed work here. The assumption made was the loading induced elastic and plastic deformation, while the unloading removed the elastic loading. Therefore, only plastic deformation should be left. However, the plastic deformation satisfied the compatibility equation, thus leading to Zimmerman supporting the 3-term multiplicative of the deformation gradient proposed by Clayton (Clayton *et al.*, 2006), expressed as

$$\mathbf{F}=\mathbf{F}_e\mathbf{F}_i\mathbf{F}_p \,. \tag{2.31}$$

In conclusion, the method proposed to calculate the plastic spin entailed loading a body and calculating $\mathbf{F}$, unloading the body and calculating $\mathbf{F}_p$., and determining $\mathbf{F}_e$ with equation 2.22. This would lead to determining $\mathbf{L}_p$ and ultimately $\mathbf{w}_p$. However, the information provided tells us that simply unloading the body and removing the elastic deformation to reveal the plastic deformation is

not possible. A more elaborate method must be used to determine the plastic spin.

2.4.6 The New Plastic Spin Hypothesis

The new hypothesis proposes determining the elastic and plastic deformation gradients at different locations from the same simulation at the same time. Although load-unload method presently will not work given the information in the previous section, we can make use of a couple of facts about the simulation to determine the elastic and plastic deformation gradients.

While deformation is occurring in a polycrystalline system, deformation occurs in different grains at various rates. The incompatibilities begin to build up and move towards the grain boundaries. However, smaller sections of these grains unload once enough deformation has occurred, leaving only elastic deformation at the center of the sections. Therefore, the plastic deformation gradient can be found at the boundaries of these sections and the elastic deformation gradient can be found at the center of the sections.

The proposed plan is to load the simulation until plastic deformation occurs and calculate the deformation gradient at the center of the sections and at the boundary of the sections in a polycrystalline simulation. The bulk deformation gradient can also be found for the simulation, therefore a comparison of the bulk with the elastic and plastic deformation gradients can take place with Equation 2.22. If Equation 2.22 is satisfied, the plastic spin can be determined in a similar fashion as Section 2.4.5.

A new plan has been proposed for determining the elastic and plastic deformation gradients at the atomistic scale. The validity of the proposed plan has yet to be tested, but it is a step in the right direction for determining the plastic spin in a discrete mathematical framework.

CHAPTER III

EAMPOST

3.1 Background

Providing a link between the atomic scale and the continuum framework is a vital part of the hierarchical approach of material modeling. Many phenomena found in atomistic simulations need to be understood and described in terms of a higher scale. This link would allow us to compare kinematics at the atomic scale with larger scale continuum results. However, relating deformation at the nanoscale to the larger length scales is difficult because there are no intrinsic measures that relate the two.

At the root of the difficulty is a difference in body type. At the macroscale, a continuum body is considered to have continuous mass and volume within an explicitly defined boundary or surface, along with dimensions much greater than atomic spacings. The motion of a continuum body can be explicitly defined by a one-to-one mapping of a point from the reference configuration to a point in the current configuration. In contrast, an atomistic body is discrete and composed of a collection of finite number of particles with a lack of matter between the particles. The motion of a discrete body is characterized by the position and velocity of each particle. Because the continuum measures of deformation rely

on the gradient of a continuous displacement field, they do not apply to the nanoscale in their current state due to the lack of a continuous body.

3.2 Discrete Gradient Operator

The deformation of a discrete body at the nanoscale has been commonly characterized by the centrosymmetry parameter (Kelchner *et al.*, 1998) and slip vector (Zimmerman *et al.*, 2001). However, there is no relationship between these quantities and strain or deformation commonly used in continuum-based analysis. In order to compare the atomistic scale to the continuum scale, a strain measure must be defined for atomistic bodies. There are two options for creating strain measures at the atomistic scale: (1) create a continuous displacement fields by interpolating atomic data or (2) devise a discrete gradient operator. The deformation gradient is essential in calculating strain measures used at higher length scales.

Gullett *et al.* (2008) proposed a discrete deformation gradient which can capture the motion of a body of discrete objects similar to a deformation gradient in a continuous body. One difficulty of the discrete deformation gradient was solving the system of linear equations. For a single atom and a single neighbor, a unique deformation gradient can be found. But, the deformation gradient for a single atom and each of its neighbors may not be identical. An optimal deformation gradient that solves the system of linear equations between an atom and its neighbors is sought and the error between them minimized.

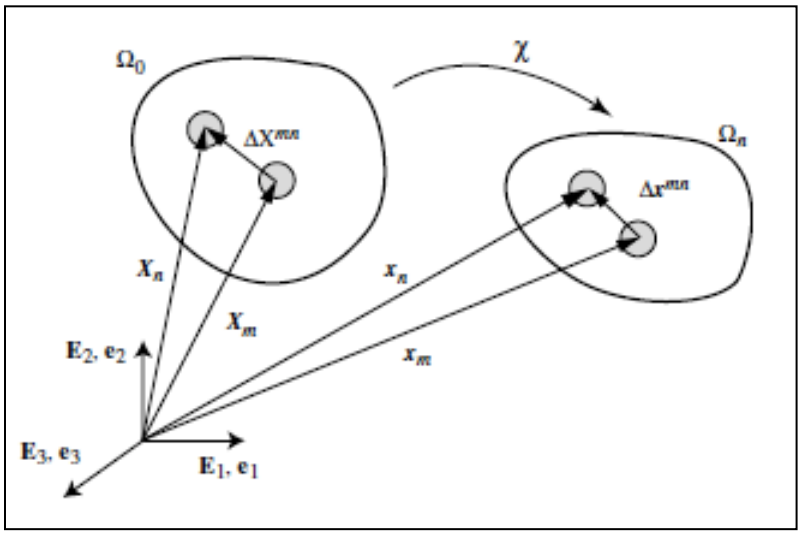

Figure 3.1 The mapping between reference and current configuration for a discrete field (Gullett *et al.*, 2008).

In that work, the discrete operator is developed by first considering the relative position of atoms, m, to neighbors, n is

$$\Delta X^{mn} = X^n - X^m , \tag{3.1}$$

for the reference configuration and

$$\Delta x^{mn} = x^n - x^m , \tag{3.2}$$

for the current configuration. Using the continuum definition of the deformation gradient, the linear mapping is defined as

$$\Delta x^{mn} = \mathbf{F}^m \cdot \Delta X^{mn} , \tag{3.3}$$

where $\mathbf{F}^m$ is the deformation gradient at m for position $\mathbf{x}^m$. Because of the considerable freedom of motion between atoms m and n, the system of equations created by Equation 3.3 do not have a single $\mathbf{F}^m$. So, to find the "best fit" $\mathbf{F}^m$, the

mapping error between m and each n is measured as the L^2-norm of the difference, which is expressed as

$$\phi^{mn} = \left(\Delta x^{mn} - \mathbf{F}^m \cdot \Delta X^{mn}\right)^{\mathrm{T}} \left(\Delta x^{mn} - \mathbf{F}^m \cdot \Delta X^{mn}\right). \tag{3.4}$$

Therefore, the weighted least squares error for m over N number of neighbors can be expressed as

$$\phi^m = \sum_{n=1}^{N} \left(\Delta x^{mn} - \mathbf{F}^m \cdot \Delta X^{mn}\right)^{\mathrm{T}} \left(\Delta x^{mn} - \mathbf{F}^m \cdot \Delta X^{mn}\right) w_n, \tag{3.5}$$

where w_n is a weight function. If we multiply out and take the derivative of Equation 3.5 with respect to $\mathbf{F}$, we can solve for the deformation gradient. This equation is minimized with respect to the components of $\mathbf{F}$ and set equal to zero to yield an expression

$$\mathbf{F}^m = \left[\sum_n \left(\Delta x^{mn}\right)\left(\Delta X^{mn}\right)^{\mathrm{T}} w_n\right]\left[\sum_n \left(\Delta X^{mn}\right)\left(\Delta X^{mn}\right)^{\mathrm{T}} w_n\right]^{-1}. \tag{3.6}$$

The weight function used in this paper is a step function that applies a weight to a neighbor depending on how close it is to atom m. A step function is used instead of the cubic spline because it allows the atoms in a particular grouping to have the same weight, thus not changing an atom's weight because of thermal oscillations, as shown in Figure 3.2. The closer n is to m, the greater the impact n has on the deformation gradient of m. The weight function, w_n, is a series of step-functions (from 1 to 0) used to assign weights to atoms, n, within the cutoff radius. The n atoms are divided into groups based on proximity to m. Each group is assigned a weight value. The radius of n closest to m in each group is used to determine the weight value.

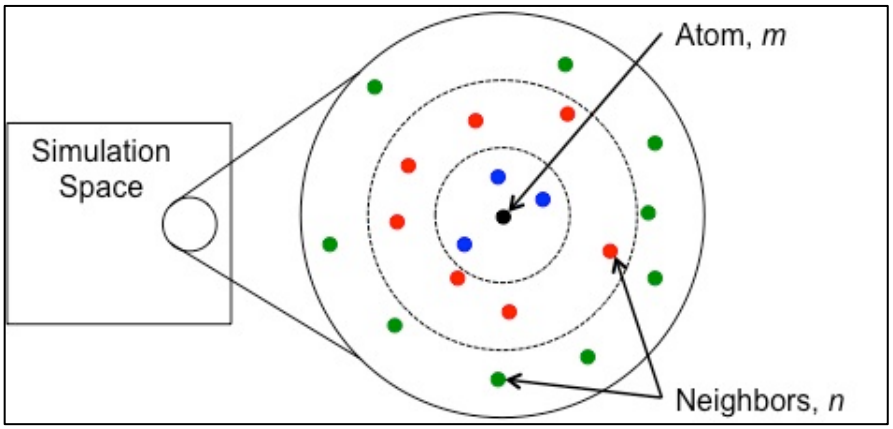

Figure 3.2 The atom, m, and the weight function for its neighbors, n, where atoms of same colors have the same weights.

3.3 Additions and Revisions

For this work, several enhancements were made to EAMpost. EAMpost, written in Fortran, was originally created and used as a post-processing program for the MD software package DYNAMO, written by Stephen Foiles and Murray Daw at Sandia National Laboratories in Livermore, CA (Daw *et al.*, 1993), so the code needed to be updated to read data from LAMMPS. EAMpost was also modified by applying the use of a more sophisticated and less computationally expensive neighbor list routine. Steve Plimpton originally wrote this routine for an earlier molecular dynamics software known as WARP (Plimpton, 1995). Finally, the code was expanded to include various continuum-based measures of deformation.

In its original form, EAMpost was capable of calculating a variety of continuum metrics. Those metrics include the deformation gradient, Green

(Lagrangian) strain, and Almansi strain. For the purposes of quantifying failure and, eventually, the plastic spin, additional continuum metrics were considered and some added to the post-processing program.

3.3.1 Tensor and Scalar Deformation Strain Measures

Upon reviewing the literature, very few studies have been done utilizing local atomic strain measures. Mott *et al.* (1992) proposed a local atomic strain measure for three-dimensional, disordered systems, such as glass. Mott's approach calculated a strain directly from atomistic data using the small strain assumption, previously shown in Equation 2.10. He also utilized the norm of the strain deviator to analyze an atomic deformation strain measure, a dilatation measure defined as

$$\left\|\varepsilon_{dev}\right\|^2 = \tfrac{2}{3}\,\mathrm{tr}\left(\varepsilon_{dev}^{\mathrm{T}}\varepsilon_{dev}\right) \quad , \tag{3.7}$$

where ε_G is the Green's strain and the deviatoric strain, ε_{dev}, is defined as

$$\varepsilon_{dev} = \varepsilon_G - \tfrac{1}{3}\,\mathrm{tr}\left(\varepsilon_G\right)\mathrm{I} \quad . \tag{3.8}$$

A similar measure to Mott's strain tensor is the first invariant of the Green strain tensor. The first invariant is the trace of the Green strain and is also a dilatation measure. It is expressed as

$$I_1 = \mathrm{tr}\left(\varepsilon_G\right). \tag{3.9}$$

Another strain value of interest is the effective strain. The effective strain is a measure commonly used to express bulk strains in stress-strain curves and is expressed as

$$E_{eff} = \sqrt{\tfrac{1}{2}\left(\varepsilon_{11} - \varepsilon_{22}\right)^2 + \left(\varepsilon_{11} - \varepsilon_{33}\right)^2 + \left(\varepsilon_{22} - \varepsilon_{33}\right)^2}\ . \tag{3.10}$$

To better explore the expansive or contractive nature around an atom, the determinant of the deformation gradient is used. The determinant of the deformation gradient, also referred to as the Jacobian (J), is a measure of the change in volume from the reference configuration to the current configuration and is expressed as

$$J = \left|\mathbf{F}\right|. \tag{3.11}$$

3.3.2 Tensor and Scalar Deformation Stress Measures

In addition to strain measures, values of stress were also considered. The hydrostatic stress is explored because of its role in elastic volume changes, which can play a role in the fracture strain of the material (Dieter, 1976). The equation for hydrostatic stress, sometimes referred to as pressure, is

$$\sigma_{hyd} = \tfrac{1}{3}\left(\sigma_{xx} + \sigma_{yy} + \sigma_{zz}\right). \tag{3.12}$$

Another measure of stress commonly used is the effective stress, which has alternate names such as the deviatoric stress or von Mises stress. Although it is more commonly used in bulk stress-strain response, the deviatoric portion of the strain tensor contains the shear stresses, which cause plastic deformation. The effective stress is expressed as

$$\sigma_{eff} = \tfrac{1}{2}\sqrt{\left(\sigma_{xx} - \sigma_{yy}\right)^2 + \left(\sigma_{xx} - \sigma_{zz}\right)^2 + \left(\sigma_{yy} - \sigma_{zz}\right)^2}\ . \tag{3.13}$$

The last stress measure is a ratio of the hydrostatic stress to the deviatoric (effective) stress, also referred to as stress triaxiality. The stress triaxiality has been previously determined as a critical factor of void nucleation and void growth. Triaxiality is defined as the hydrostatic stress divided by the deviatoric stress, or

$$\chi = \frac{\sigma_{hyd}}{\sigma_{dev}}. \tag{3.14}$$

3.4 EAMpost Validation

Once EAMpost was written and compiled to work with LAMMPS dump data, it was ready to be validated. An ideal, discrete particle simulation was performed where the deformation gradient could be found manually. An ideal simulation was not created with MD software, but by building a set of points, defined by a smooth displacement field, and imposing motion based on the points and their position in the field. Simple shear was applied to the points for a desired amount of time to confirm the precision of EAMpost.

The simulation box consisted of 50,000 particles with a spacing of 5.0 Å and a box of dimensions 625 Å (wide) x 500 Å (high) x 20 Å (thick). This created a fictitious set of particles that could be deformed in accordance with simple shear movement. The simulation schematic is shown in Figure 3.3 and the script file is located in Appendix A.

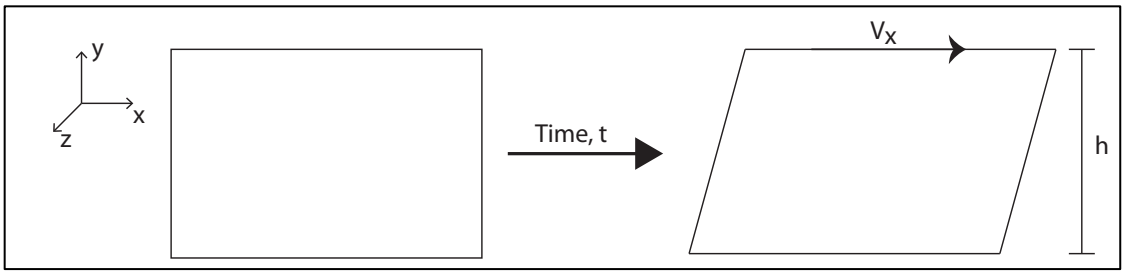

Figure 3.3 Illustration of the simple shear of a box for EAMpost validation.

The applied velocity for the simulation, v_x, was 0.0025 Å/ps with a total simulation time of 50,000 ps. The velocity was applied to particles based on their height in the simulation box. Atoms at the bottom were held fixed (v = 0 Å/ps), atoms at the top exhibited the full velocity (v = 0.0025 Å/ps), and atoms in between had a velocity according to

$$v_i = \frac{v_x y_i}{h}, \tag{3.15}$$

where v_i is the velocity of particle i, h is the height of the simulation, y_i is the height of particle i, and t is the time. The current x-location of the atom could be found according to

$$x_i = x_0 + v_i t\,. \tag{3.16}$$

Simple shear and the values mentioned below come from (Khan *et al.*, 1995). The mapping of reference configuration to current configuration is described by

$$\begin{aligned} x &= X + \frac{vt}{h} Y \\ y &= Y \\ z &= Z \end{aligned}\quad . \tag{3.17}$$

The deformation gradient associated with this motion is

$$\mathrm{F} = \begin{bmatrix} 1 & \frac{v_x t}{h} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.18}$$

The Green strain (E) is given as

$$E = \frac{1}{2}\begin{bmatrix} 0 & \frac{v_x t}{h} & 0 \\ \frac{v_x t}{h} & \left(\frac{v_x t}{h}\right)^2 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{3.19}$$

and the Eulerian strain (e) is

$$e = \frac{1}{2}\begin{bmatrix} 0 & -\frac{v_x t}{h} & 0 \\ -\frac{v_x t}{h} & \left(\frac{v_x t}{h}\right)^2 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{3.20}$$

Also, the velocity gradient is expressed as

$$\mathrm{L} = \dot{\mathrm{F}}\mathrm{F}^{-1} = \begin{bmatrix} 0 & \frac{v_x}{h} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{3.21}$$

which can be decomposed into stretch components,

$$\mathrm{d} = \begin{bmatrix} 0 & \frac{v_x}{2h} & 0 \\ \frac{v_x}{2h} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{3.22}$$

and spin components,

$$\mathrm{w} = \begin{bmatrix} 0 & \frac{v_x}{2h} & 0 \\ -\frac{v_x}{2h} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{3.23}$$

the local particle movement should be equivalent to the bulk system movement and the theoretical values should compare exactly with our discrete deformation gradient. Note that for this linear motion, a single, unique $\mathbf{F}$ exists.

Table 3.1 A comparison of the theoretical and bulk values for simple shear of an ideal, discrete particle simulation.

Timestep	**10,000 ps**		**30,000 ps**		**50,000 ps**	
Value	**Theoretical**	**Bulk**	**Theoretical**	**Bulk**	**Theoretical**	**Bulk**
$\mathbf{F_{12}}$	0.0202	0.0202	0.0606	0.0606	0.101	0.101
$\mathbf{E_{12}}$	0.0101	0.0101	0.0303	0.0303	0.0505	0.0505
$\mathbf{L_{12}}$	2.0×10^{-6}	2.0×10^{-6}	6.1×10^{-6}	6.1×10^{-6}	1.0×10^{-5}	1.0×10^{-5}
$\mathbf{d_{12}}$	1.0×10^{-6}	1.0×10^{-6}	3.0×10^{-6}	3.0×10^{-6}	5.1×10^{-6}	5.1×10^{-6}
$\mathbf{w_{12}}$	1.0×10^{-6}	1.0×10^{-6}	3.0×10^{-6}	3.0×10^{-6}	5.1×10^{-6}	5.1×10^{-6}

As shown in Table 3.1, the computed values for the xy-component of the deformation gradient, Green strain, velocity gradient, stretch, and spin match exactly for time steps of 10,000, 30,000, and 50,000 ps. This shows the code calculated the value properly and implemented the procedure correctly. This also illustrates that a first-order approximation of the deformation gradient is adequate in capturing this motion.

CHAPTER IV

MOLECULAR DYNAMIC SIMULATIONS

4.1 Simple Shear

In this section, local deformation of an atomic system was subjected to simple shear boundary conditions and evaluated. Once EAMpost was validated with an ideal discrete particle simulation, a simple shear simulation was run with LAMMPS to determine the effects of variations of atomic movement.

4.1.1 Simulation Overview

The MD simple shear simulation was setup very similarly to the ideal simple shear simulation used to validate EAMpost. The simulation box consists of 44,513 aluminum atoms with a lattice spacing of 4.05 Å and a box of dimensions 636 Å (wide) x 313 Å (high) x 242 Å (thick). The lattice crystal direction $[100]$, $[011]$, & $[01\bar{1}]$ correspond to the x-, y-, and z-direction, respectively.

The simulation was run using the an EAM potential (Mendelev *et al.*, 2008) to calculate the system energies and determine the interaction forces between atoms. With the interatomic forces calculated, an isothermal-isobaric NPT ensemble, where P stands for pressure and T stands for temperature, was

used to advance the atomic positions and velocities through time. The temperature equlibration and pressure minimization are performed with a Nose/Hoover temperature thermostat (Hoover, 1985) and Nose/Hoover pressure barostat (Hoover, 1986) as implemented by Melchionna (1993). Coefficients for the NPT ensemble included a pressure coefficient, damping coefficient, and drag coefficient of 0.5, 25, and 50, respectively. The coefficients are used to aid in dampening the oscillations of temperature and pressure. LAMMPS uses a velocity Verlet algorithm (Verlet, 1967; Swope *et al.*, 1982) to integrate the equations of motion with a time step of 5 fs.

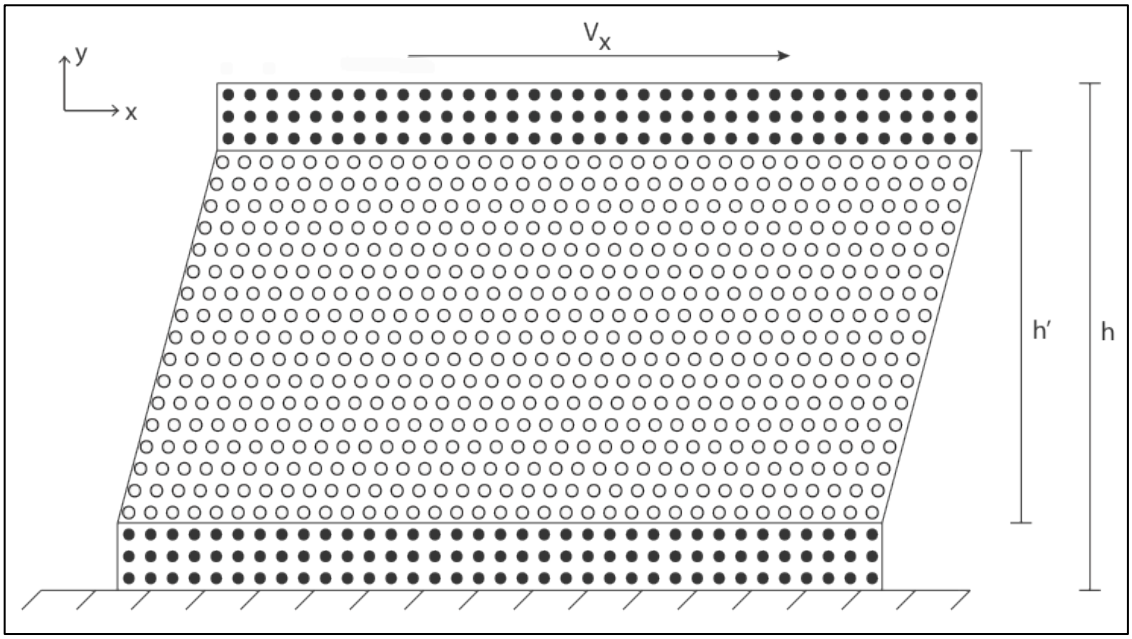

Figure 4.1 Schematic of the molecular dynamic simple shear simulation. The black dots on the top were applied a velocity, the black dots on the bottom were held fixed, and the white dots in the middle were allowed to move freely.

All of the molecular dynamic simulations were run on a Sun X2200 M2 Server containing 512 AMD® Opteron 2.60 GHz quad-core processors with the SuSE SLES 10 operating system.

These simulations were relaxed for 150 ps (30,000 time steps) with a Nose-Hoover thermostat/barostat with a temperature of 0.01K and zero pressure in the z-direction. The LAMMPS input file for the relaxation of this atomic system is located in Appendix B. Three groups were created to apply the simple shear boundary conditions. The top group, shown as the black dots at the top of Figure 4.1 and comprised of 5,329 atoms, were applied a velocity in the x-direction. The bottom groups, shown as the black dots at the bottom of Figure 4.1 and comprised of 5,329 atoms, were held fixed throughout the simple shear simulation. The free atoms group, the white dots between the top and bottom groups, is comprised of 33,855 atoms and were allowed to move freely.

The simple shear simulation was run for 320,000 time steps (1600 ps) with an applied velocity in the x-direction, v_x, of 0.01 Å/ps. This was achieved with a LAMMPS command that allows one to apply a prescribed velocity to a group of atoms regardless of the forces acting on the atoms (Plimpton, 1995). Therefore, the applied velocities of atoms in the top group were 0.01, 0, and 0 Å/ps in the x-, y-, and z-direction, respectively and the bottom group was held fixed in the x-, y-, and z-directions. This LAMMPS input file is located in Appendix C.

In addition to the solid simple shear simulation, a simple shear simulation with a hole was constructed. The aluminum simulation was created in accordance with Figure 4.2, with the only difference in it and the solid simple

shear being the hole and its radius of 12.15 Å. This simulation has all of the specifications of the solid simple shear simulation, which minor changes to the total number of atoms (42,810) and the number of free atoms (32,245).

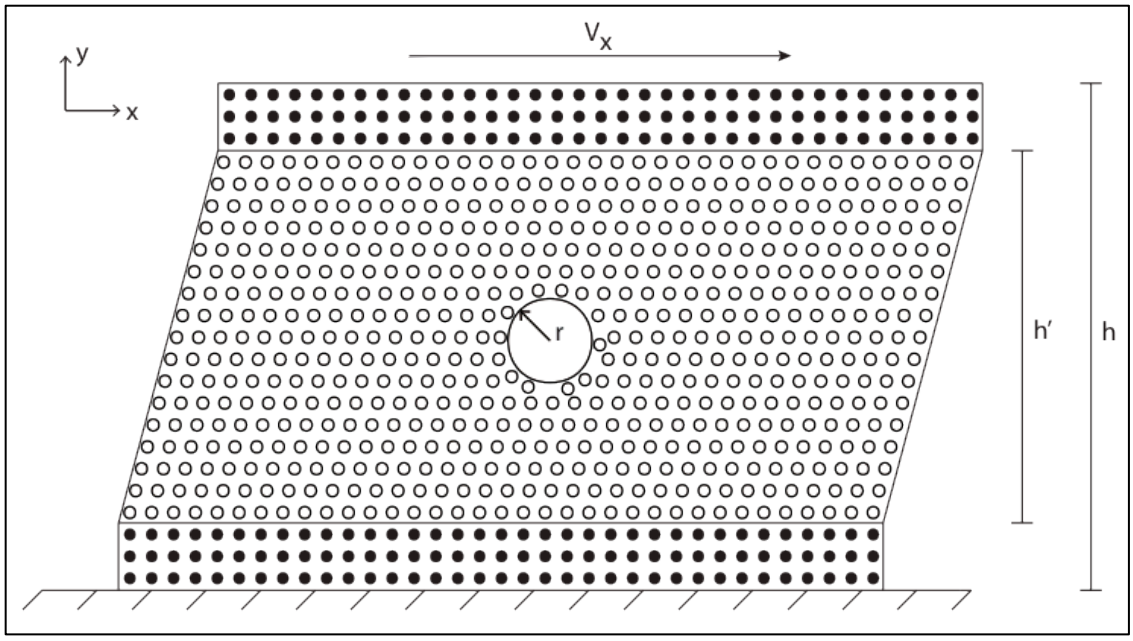

Figure 4.2 Schematic of the molecular dynamic simple shear simulation with a hole. The black dots on the top were applied a velocity, the black dots on the bottom were held fixed, and the white dots in the middle allowed to move freely.

4.1.2 Simple Shear Results

The bulk shear stress versus nominal shear strain curve for the simple shear simulation is shown in Figure 4.3. The stress-strain response is a typical response with a linear behavior and plastic deformation to 20% nominal shear strain. The block of aluminum also endured plastic deformation, so the atomic strain tensors will contain elastic and plastic deformation. It was important to

validate EAMpost in the elastic regime as well as the plastic portion of a materials' deformation. The elastic portion of the stress-strain curve is linear with little deviations, most likely due to the running of the simulation at a temperature of 0.01K thus removing thermal oscillations. We wanted to validate EAMpost with MD software, but we did not want to introduce too much noise in the data due to temperature rescaling.

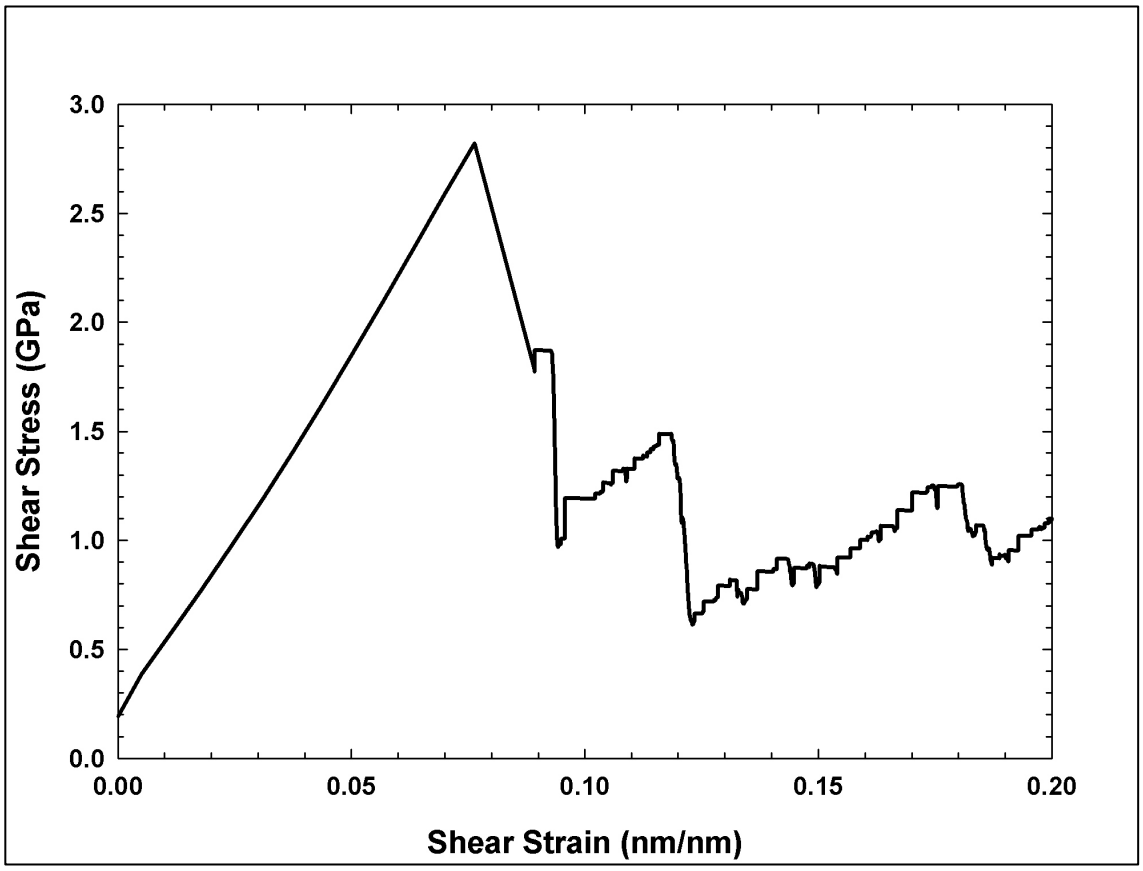

Figure 4.3 The bulk shear stress versus nominal shear strain curve for the aluminum MD simple shear simulation without a hole.

EAMpost was able to capture the entire deformation gradient and subsequent strain tensors with good success. Table 4.1 illustrates the shear (xy) component of the Green strain as calculated using Equation 3.19, the mode of the atomic values, and the mean ± the standard deviation of the atomic values. Table 4.1 illustrates there was a difference ranging from 1.2% difference (100 ps) to 27.4% difference (600 ps) between the nominal value and the average atomic value. However, the nominal is an exact match to the mode value, or the value of highest frequency in the simulation at a specific time step.

Table 4.1 Comparison of the nominal, mode, and average of the atomic values of the shear component of the Green strain for the simple shear simulation without a hole.

Time (ps)	Green Strain (xy) Component (nm/nm)		
	Nominal	**Mode**	**Mean ± Std Dev**
100	0.0083	0.0083	0.0104 ± 0.0028
200	0.0167	0.0168	0.0173 ± 0.0047
400	0.0333	0.0338	0.0312 ± 0.0085
600	0.05	0.0505	0.0448 ± 0.0127

This is further illustrated in Figure 4.4, a histogram that shows the shear component of the Green strain at particular instances in time versus atom count. The majority of the top and bottom group of the simulation, approximately 10,000 atoms, would have a shear strain component close to zero because the atoms are held constant distances from each other over time. That leaves approximately 33,000 atoms to move freely. At a time of 100 ps, the nominal shear strain value is equal to 0.0083 nm/nm. In Figure 4.4a, approximately 14,985 atoms are between the range of shear strain values of 0.008 and 0.0086

nm/nm. Similarly, in Figure 4.3b, approximately 15,334 atoms have a shear strain value between 0.016 and 0.0175 nm/nm while the applied shear strain is 0.0167 nm/nm.

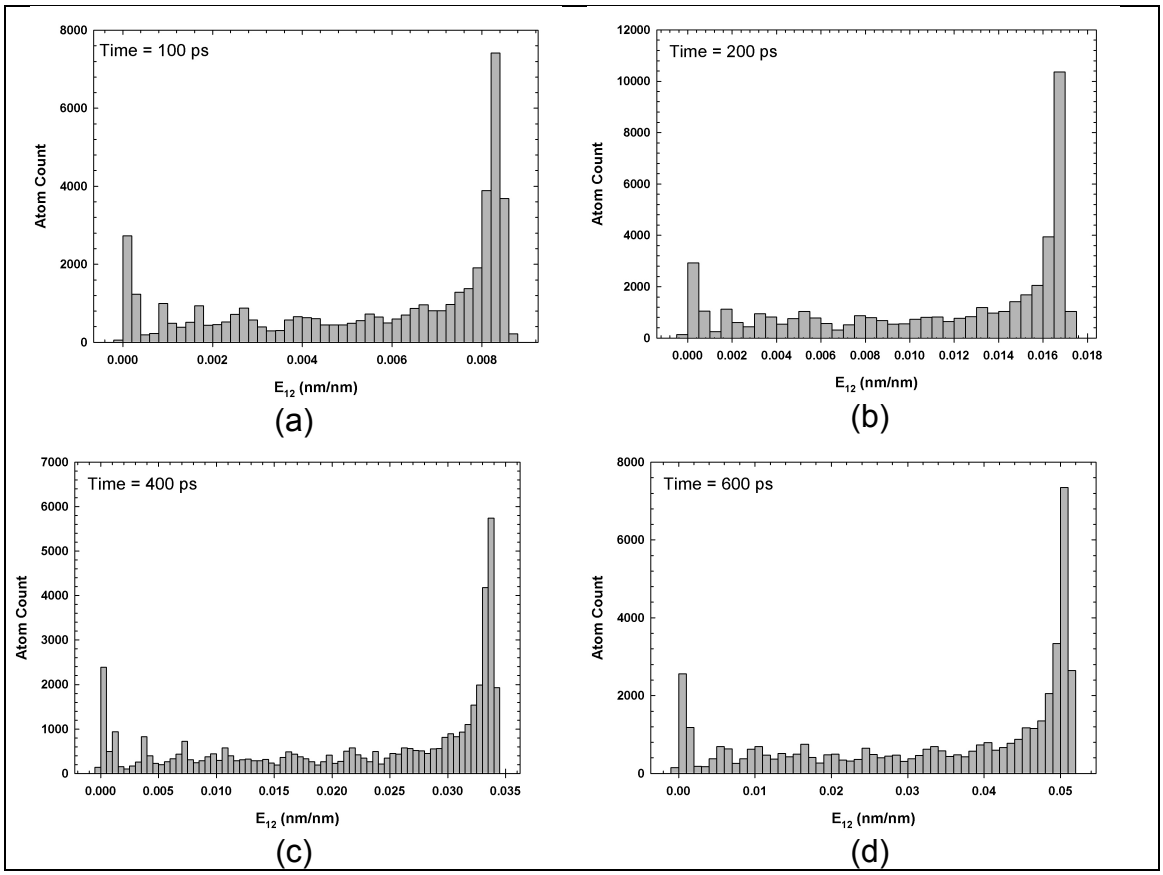

Figure 4.4 Histograms of the shear strain versus atom count at (a) t=100 ps, (b) t=200 ps, (c) t=400 ps, and (d) t=600 ps

Figure 4.5 illustrates the spatial distribution of the shear strain values. In Figure 4.5b, the top and bottom regions are blue, indicating their shear strain values equal zero. But, the center of the simulation box has a constant shear strain value that is consistent with the applied value. Moving further in time,

Figure 4.5d shows a distribution of shear strain values which are constant towards the center but vary at the edges. Although the shear strain values are not constant over the entire simulation, Figure 4.5, supported by Figure 4.4 and Table 4.1, demonstrate the simple shear simulation is moving with motion and boundary conditions that are consistent.

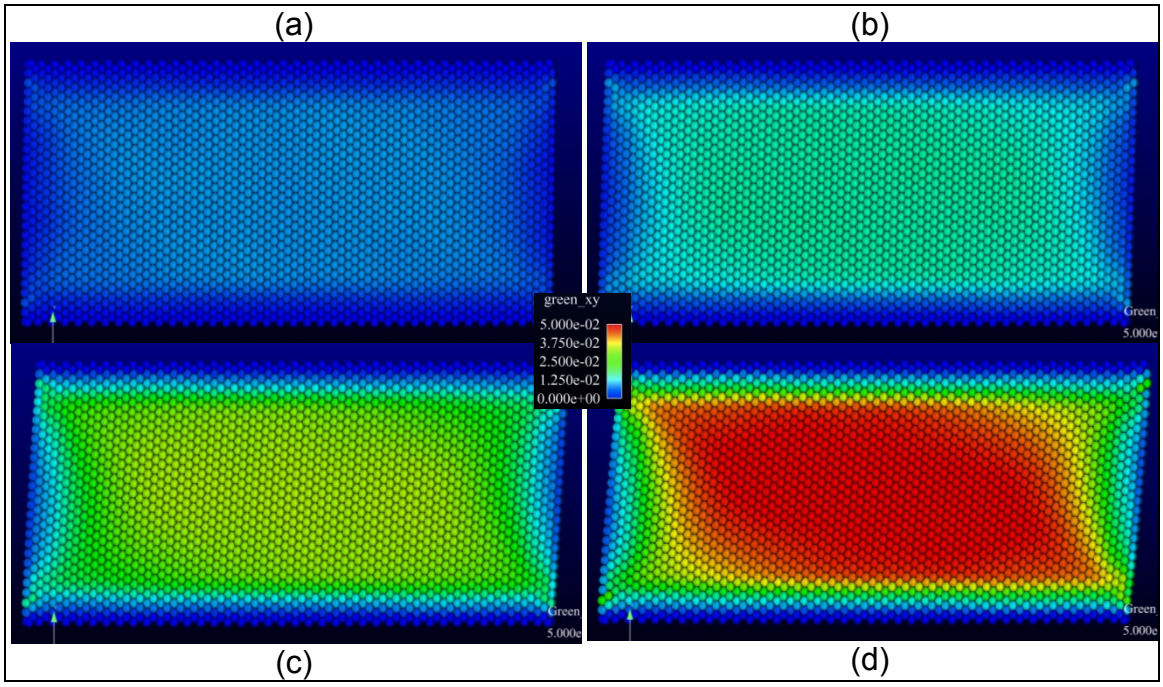

Figure 4.5 The Green shear strain for the simple shear simulation without a hole at (a) t=100 ps, (b) t=200 ps, (c) t=400 ps, and (d) t=600 ps.

The velocity gradient is the next intermediate value in the process of determining the plastic spin. In Figure 4.6, the (xy) component of the velocity gradient is shown for the simple shear simulation without a hole. The values are zero around the top and bottom region with a consistent value in the center of the simulation box. In Figure 4.6d, the velocity gradient slightly increases at the top-

left and bottom-right corner of the simulation box. This coincides with the region in Figure 4.5d where the Green shear strain values increase in both of those corners as well.

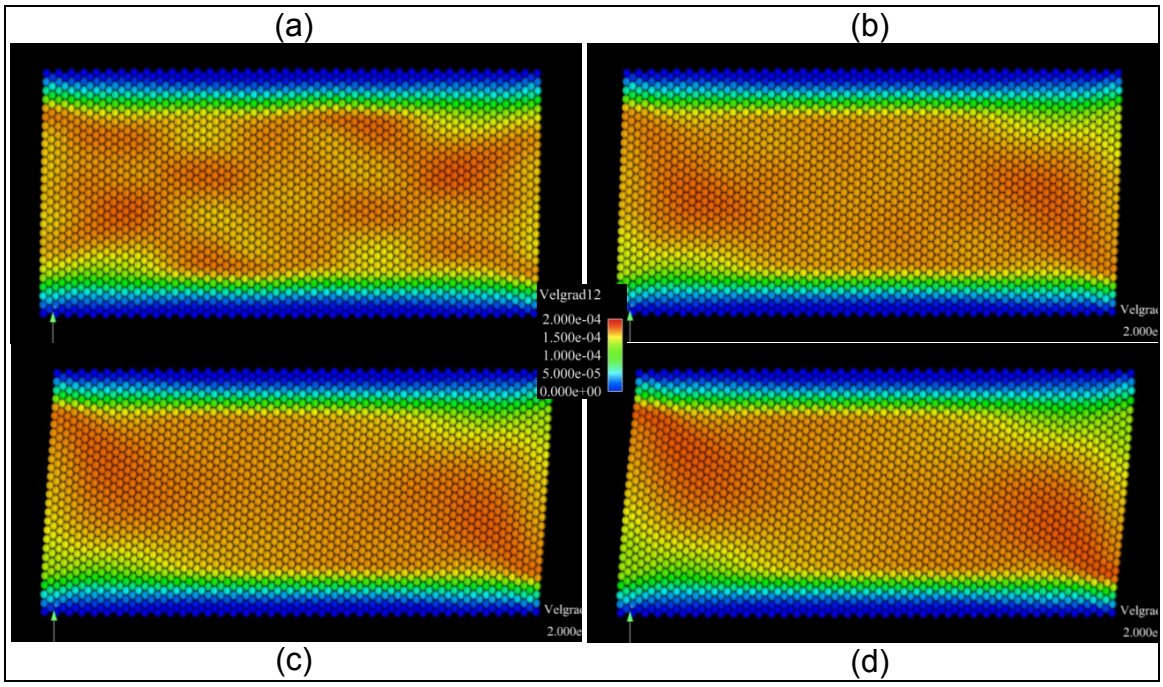

Figure 4.6 The xy-component of the velocity gradient for the simple shear simulation without a hole at (a) t=100 ps, (b) t=200 ps, (c) t=400 ps, and (d) t=600 ps.

The spin tensor for the simple shear simulation without a hole, shown in Figure 4.7, does illustrate an increase in rotations (depicted in yellow) around the front and back edge of the simulation box. These images follow in the fact there are no regions of local deformation in the simple shear simulation without a hole. The regions of blue atoms located on the top and bottom of the simulation box

are due to those atoms being held in a fixed position relative to their neighboring atoms.

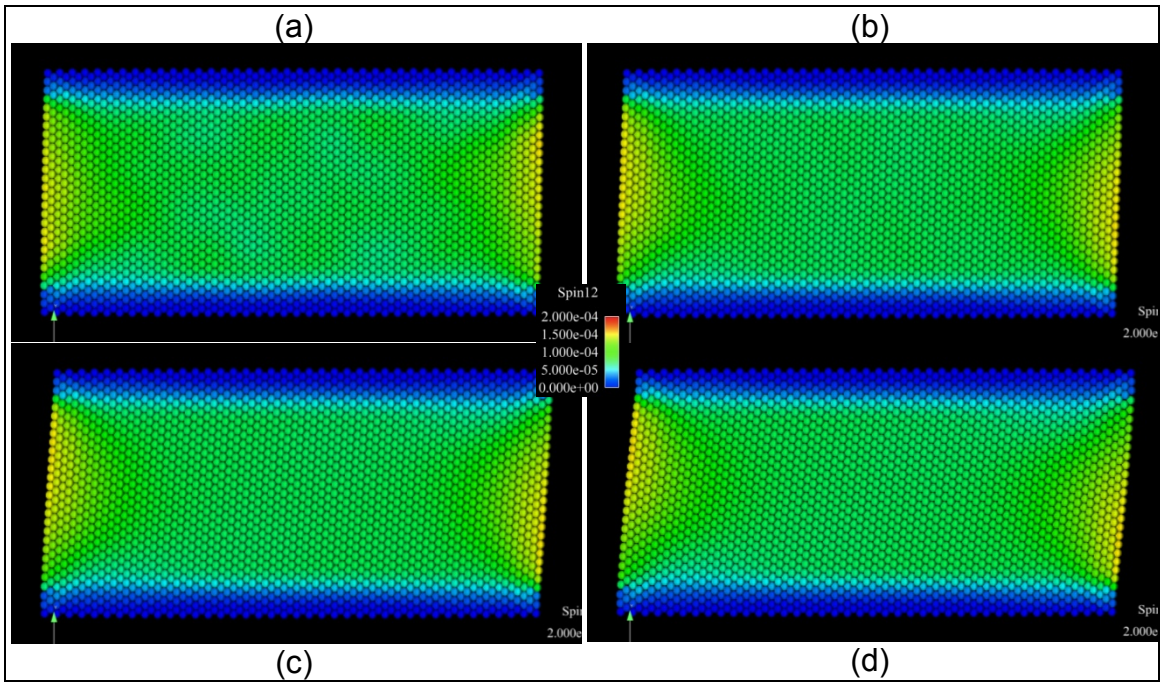

Figure 4.7 The xy-component of the spin tensor for the simple shear simulation without a hole at (a) t=100 ps, (b) t=200 ps, (c) t=400 ps, and (d) t=600 ps.

At this time, it is important to note the use of current parameters and the need for additional measures of deformation. Figure 4.8 illustrates the simple shear simulation with the atoms colored by centrosymmetry parameter. Notice the centrosymmetry values do not deviate from zero, even though it is obvious in Figure 4.5 the material is undergoing deformation. The centrosymmetry value is intended as a measure to show a deviation in the FCC lattice, but as such it does

not provide information relevant to the development of multiscale constitutive models.

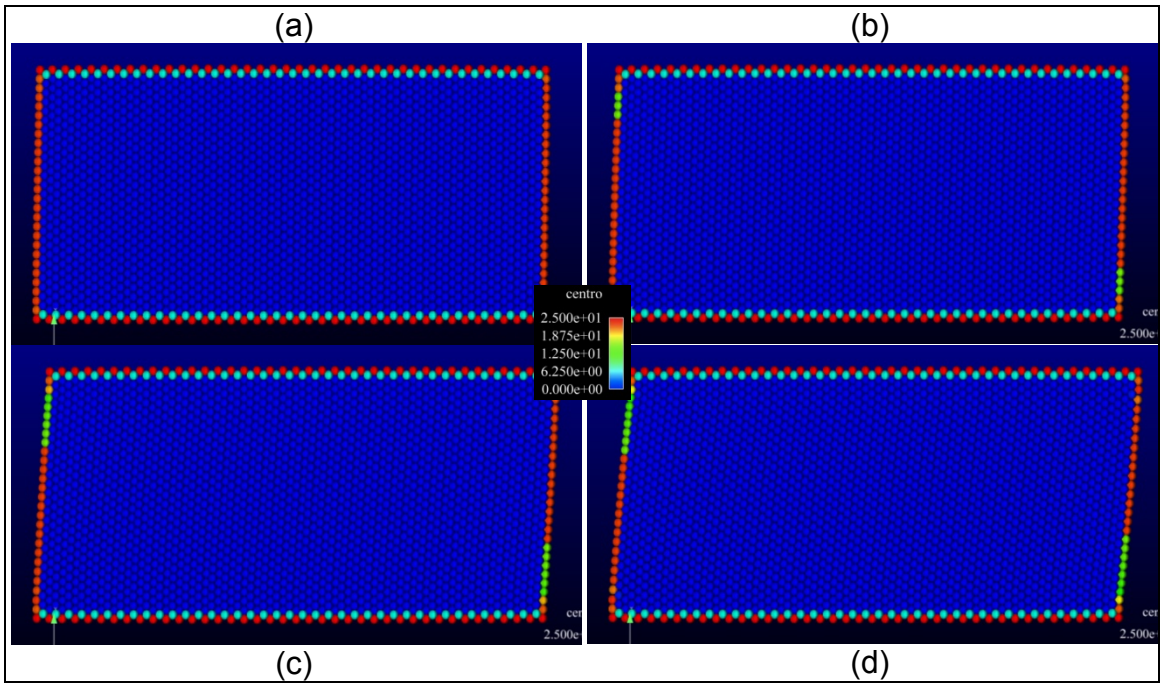

Figure 4.8 The centrosymmetry parameter for the simple shear simulation without a hole at (a) t=100 ps, (b) t=200 ps, (c) t=400 ps, and (d) t=600 ps

4.1.3 Simple Shear with a Hole Results

The bulk shear stress versus nominal shear strain curve for the simple shear simulation with a hole is shown in Figure 4.9 and is very similar to the solid simple shear simulation response. The stress-strain response is a typical response with a linear behavior and plastic deformation to 20% nominal shear strain. The yield stress of the simulation with a hole is approximately 2.5 GPa and the simulation without a hole had a yield stress of approximately 2.8 GPa for

an increase of 0.3 GPa. But, the similarities of the stress-strain curves do not tell the true story of the differences of the atomic response during deformation.

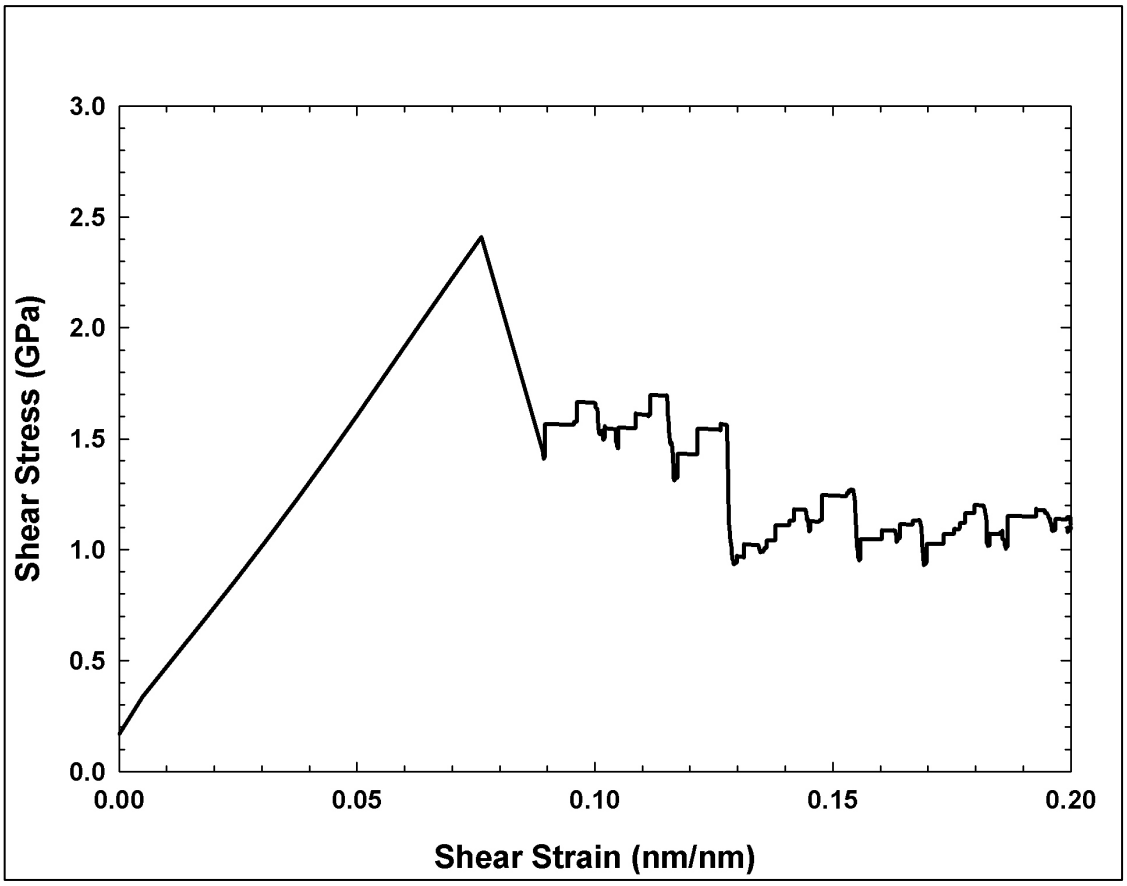

Figure 4.9 The bulk shear stress versus nominal shear strain curve for the aluminum simple shear simulation with a hole.

Figure 4.10 illustrates the following major differences in the simulation with the hole versus the solid simulation. In Figure 4.10, the Green shear strain values are not constant over the middle of the simulation block as in the simulation with a hole. However, the regions around the corners of the simulation box are similar between the simulation with a hole and without a hole.

The absence of atoms at the center causes local motion patterns that are not consistent with simple shear and shown in the simple shear simulation without a hole. Figure 4.10d illustrates low- (shown in yellow) and high-strain (shown in red) regions around the hole. The low-strain regions experience approximately 25% less shear strain than the same region in the simple shear simulation without a hole.

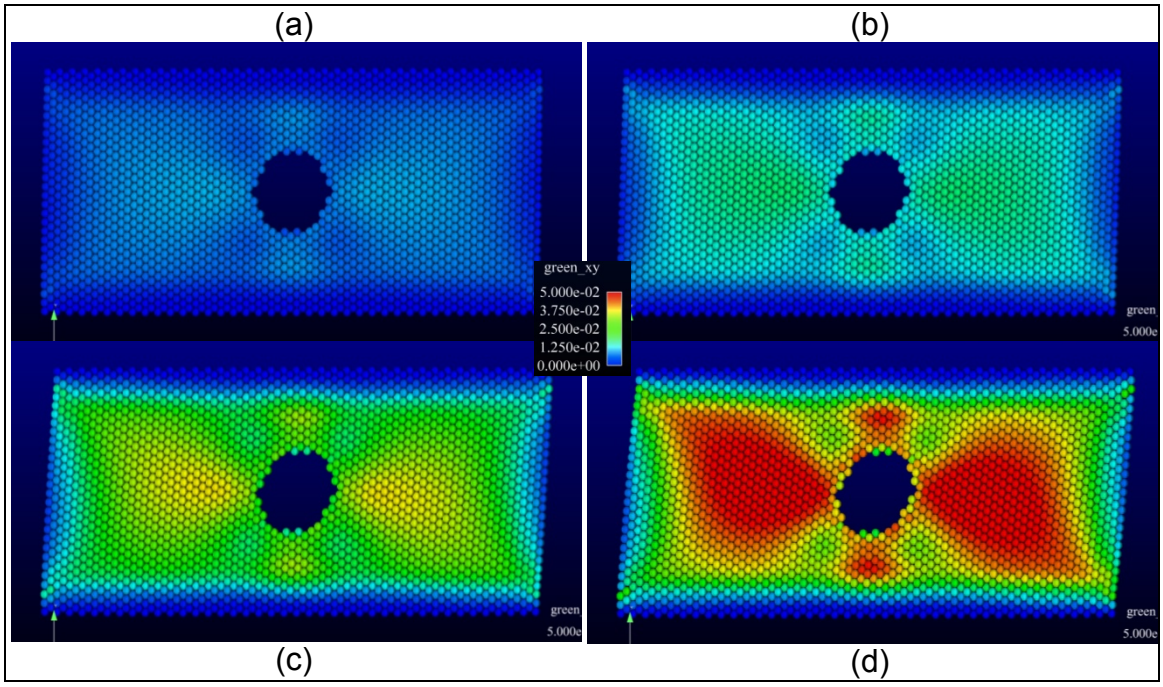

Figure 4.10 The Green shear strain for the simple shear simulation with a hole at (a) t=100 ps, (b) t=200 ps, (c) t=400 ps, and (d) t=600 ps.

Table 4.2 illustrates the shear (xy) component of the Green strain as calculated using Equation 3.19, the mode of the atomic values, and the mean ± the standard deviation of the atomic values for the simulation with a hole. Table 4.2 illustrates there was a difference ranging from 8.4% difference (100 ps) to

21.5% difference (200 ps) between the nominal value and the average atomic value. The mode value is close to zero, indicating the highest frequency of atoms are boundary atoms. In Table 4.2, the standard deviation values are becoming extremely large relative to the mean value. This is an indicator of a wide range of distribution of the atomic values.

Table 4.2 Comparison of the nominal, mode, and average of the atomic values of the shear component of the Green strain for the simple shear simulation with a hole.

Time (ps)	Green Strain (xy) Component (nm/nm)		
	Nominal	**Mode**	**Mean ± Std Dev**
50	0.0042	0.00005	0.0064 ± 0.0019
100	0.0083	0.0001	0.0095 ± 0.0028
200	0.0167	0.00025	0.0163 ± 0.0202

The distribution of the Green shear strain values for the simulation with a hole shows a marked difference than the values for the simulation without a hole. Figure 4.11 illustrates the Green shear strain distribution versus atom count. Figure 4.11c shows approximately 9,967 atoms have a shear strain value below 0.0005. This leads us to believe the boundary atoms are acting the same in this simulation as in the solid block simulation. However, the distribution of the nonzero shear strain values is changed because of the presence of the hole. At a time of 100 ps, the nominal shear strain value is equal to 0.0083 nm/nm. In Figure 4.9b, approximately 2,843 atoms are between the range of shear strain values of 0.008 and 0.0086 nm/nm. In Figure 4.11c, approximately 3,511 atoms have a shear strain value between 0.016 and 0.0175 nm/nm while the applied

shear strain is 0.0167 nm/nm. There is no concentration of atoms at the specific green strain value as in the simulation without a hole. Instead, the atoms are exhibiting a much larger range of shear strain values with no concentration equal to the applied shear strain. Comparing Figure 4.11c to Figure 4.4b, there is a broader distribution of nonzero shear strain values and a greater frequency of the zero shear strain values for the simulation with a hole as compared to the simulation without a hole.

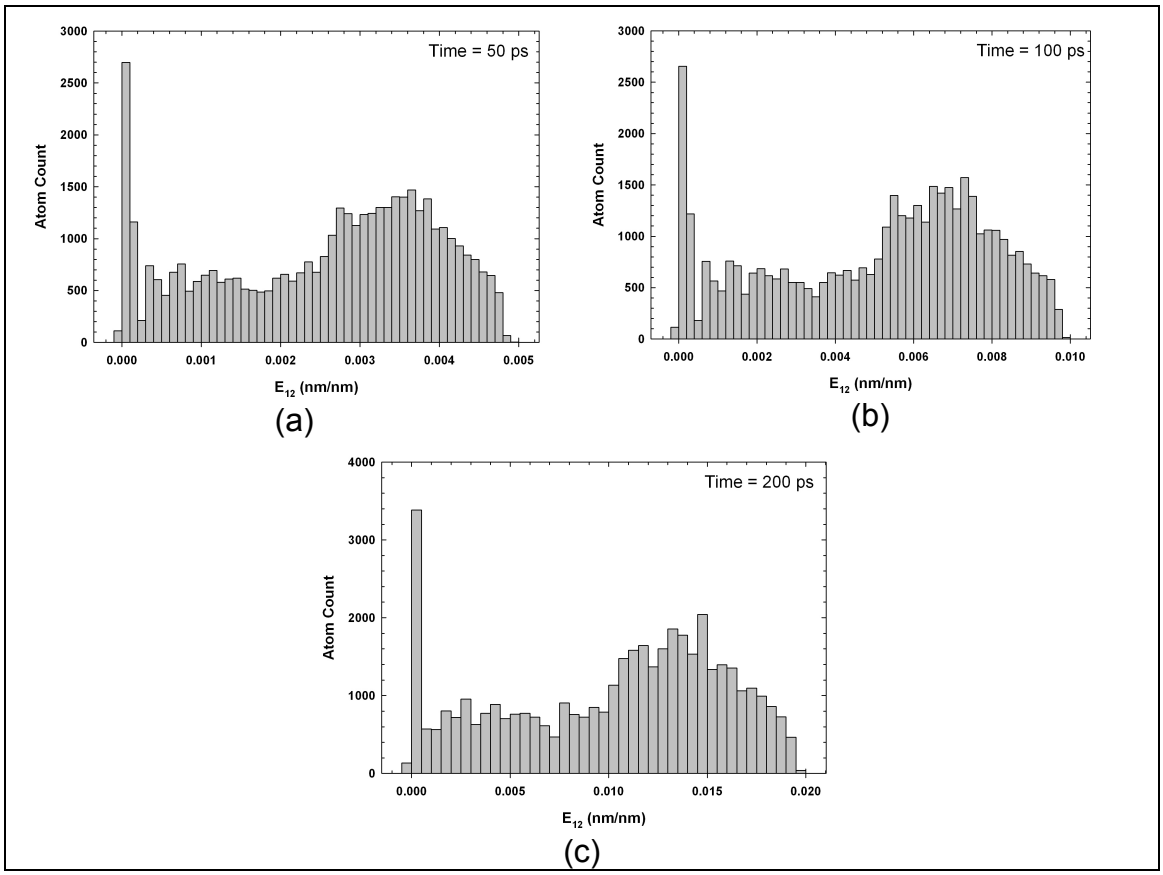

Figure 4.11 Histograms of the shear strain versus atom count for the simple shear simulation with a hole at (a) t=50 ps, (b) t=100 ps, and (c) t=200 ps.

To better explain the deformation of the simulation with a hole, images of the Jacobian or determinant of the deformation gradient are shown in Figure 4.12. In this context, the Jacobian represents the expansive or contractive nature of a region around an atom. A region that has a constant volume will have a Jacobian value equal to one.

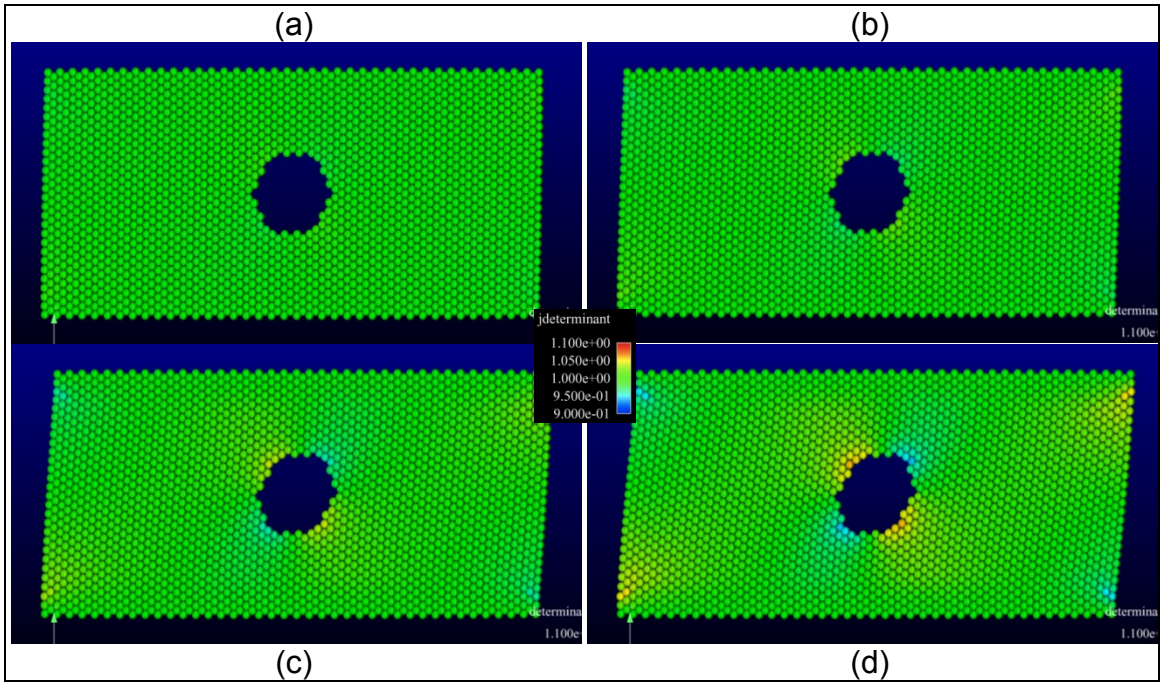

Figure 4.12 The determinant of the deformation gradient for the simple shear simulation with a hole at (a) t=100 ps, (b) t=200 ps, (c) t=400 ps, and (d) t=600 ps.

A value greater than one indicates expansion and a value less than one indicates contraction. In Figure 4.12d, regions on the top left and bottom right of the hole as well as the upper right-hand corner and lower left-hand corner of the simulation box are expanding. In comparison, regions on the top right and

bottom left as well as regions in the upper right-hand corner and lower left-hand corner are contracting. These areas of expansion and contraction are the locations that have varying shear strain values in Figure 4.10d.

Figure 4.13 illustrates the shear (xy) component of the velocity gradient for the simple shear simulation with a hole. With the presence of the hole, the velocity gradient values are not constant throughout the center like shown in Figure 4.6 for the simulation without a hole. As the simulation progresses, the velocity gradient value increases to the left and the right of the hole while it is minimal on the top and bottom of the hole.

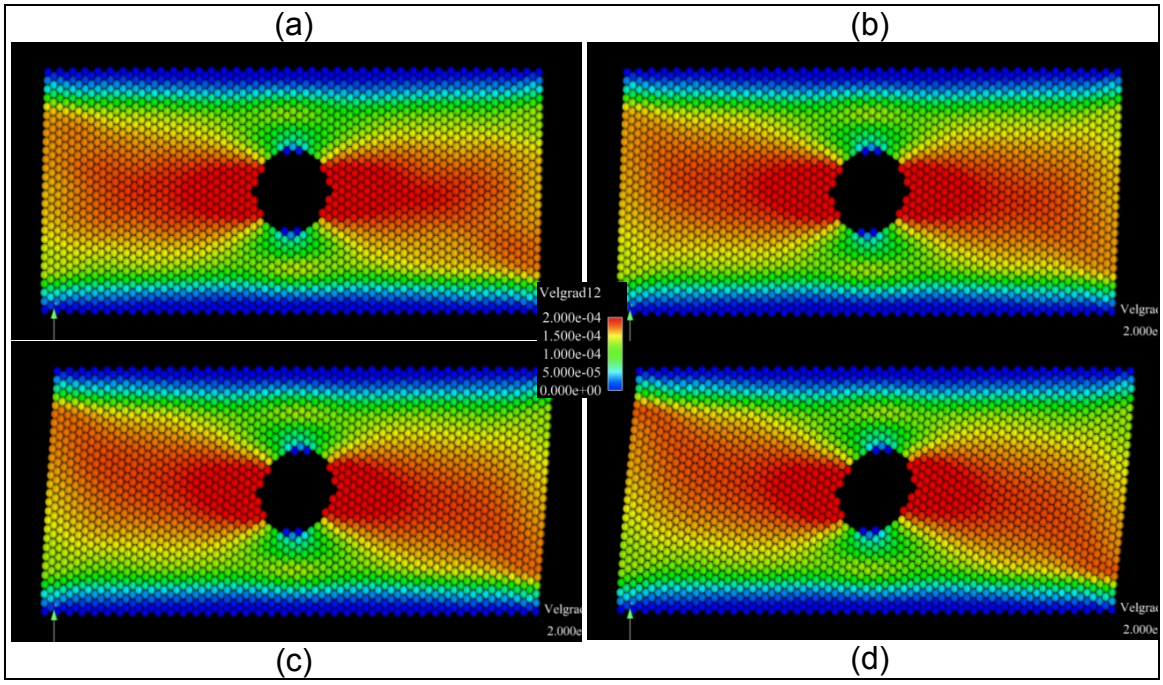

Figure 4.13 The xy-component of the velocity gradient for the simple shear simulation with a hole at (a) t=100 ps, (b) t=200 ps, (c) t=400 ps, and (d) t=600 ps.

Figure 4.14 is the shear component of the total spin for the simulation with a hole. As depicted in Figure 4.13, increased values are located to the left and right of the hole while decreased values are located on the top and bottom of the hole. This indicates minimal rotations on the top and bottom of the hole while the rotations are larger to the left and right of the hole.

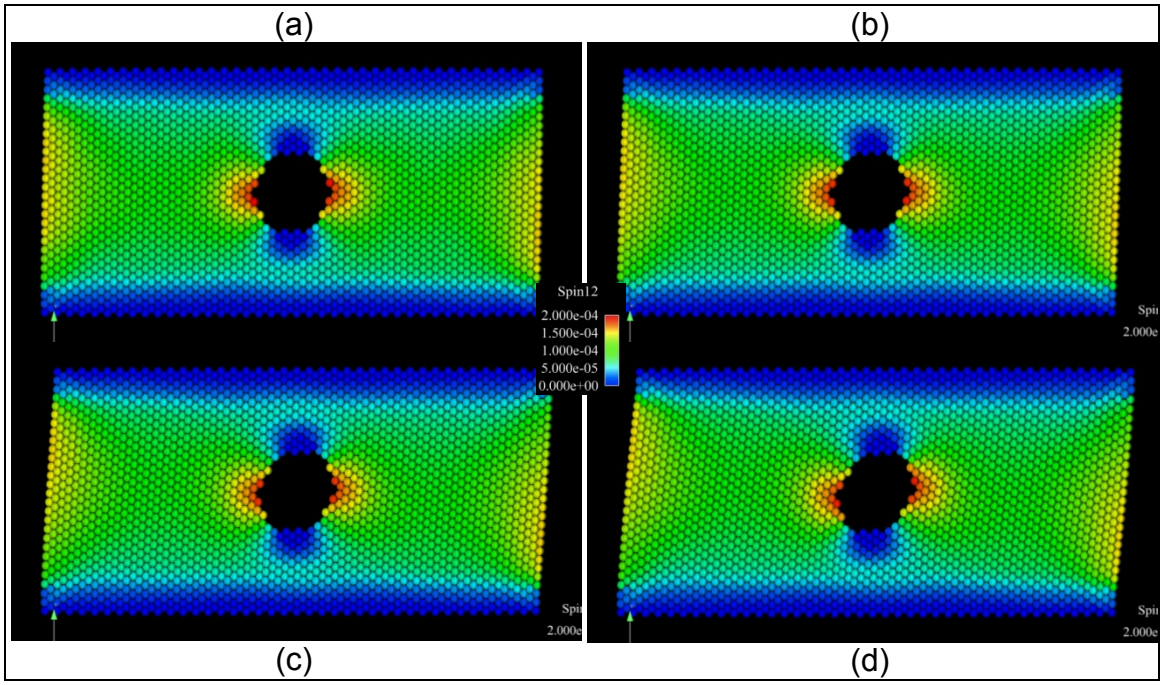

Figure 4.14 The xy-component of the spin tensor for the simple shear simulation with a hole at (a) t=100 ps, (b) t=200 ps, (c) t=400 ps, and (d) t=600 ps.

Again, the centrosymmetry parameter, pictured in Figure 4.15, did not capture the local deformation.

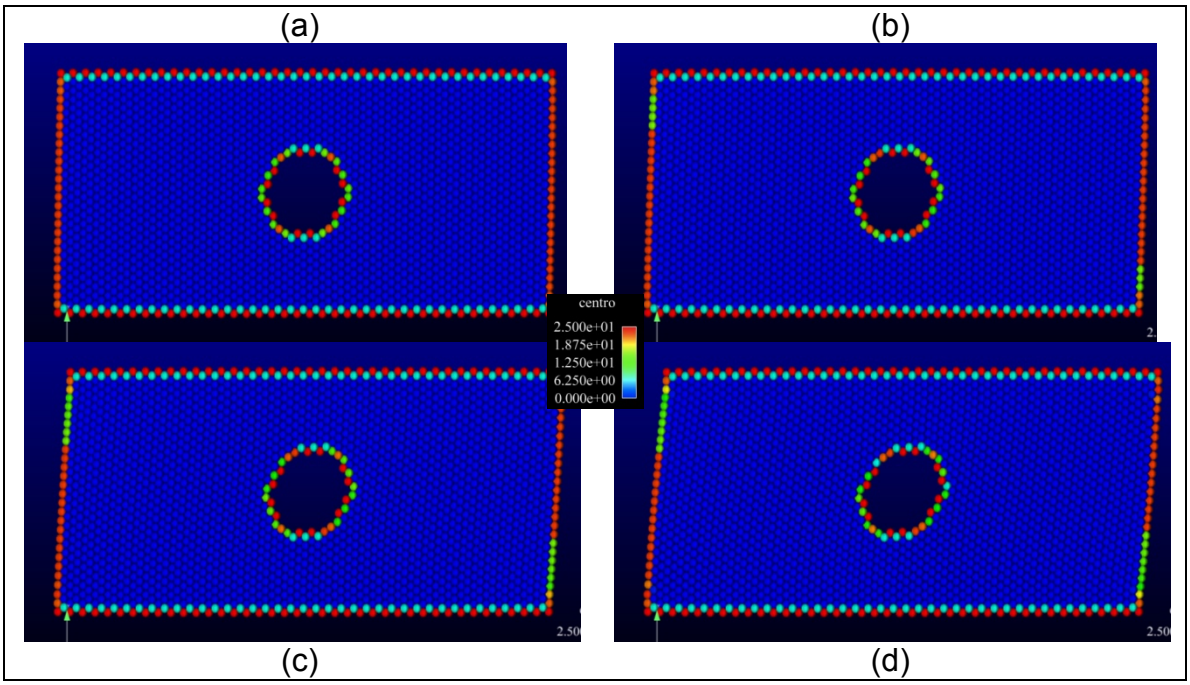

Figure 4.15 The centrosymmetry parameter for the simple shear simulation with a hole at (a) t=100 ps, (b) t=200 ps, (c) t=400 ps, and (d) t=600 ps

4.2 Uniaxial Tension of Aluminum Triple Junctions

The purpose of this project was to assist in characterizing the failure of the TJ with the following objectives: (i) quantify void nucleation in molecular dynamic simulations of TJ using a discrete mathematical framework and (ii) explore the possibility of calculating advanced deformation quantities such as the plastic spin. The groundwork for the item (i) and (ii) have been laid with the theoretical backing and the use of examples shown previously in this chapter. The knowledge and methods are now extended to the problem of which they were originally intended.

4.2.1 Simulation Overview

To further understand void nucleation at different grain boundaries, a suite of simulations with high- and low-angle grain boundary orientations were performed. For this study, high-angle grain boundaries are defined as grains which have a misorientation of 30° to 60° and low-angle grain boundaries have a misoreintation of 5° to 10°. In addition to various grain boundaries, the (110) and (100) textures were used. The introduction of the (100) texture altered the number of slip systems from four to eight (Horstemeyer *et al.*, 2002).

The triple junction structures studied consisted of three columnar grains truncated to isolate a single triple junction. The structures were created using PreWarp, a program that can create complex atomic structures which defines the center of each grain and uses a modified Voronoi construction scheme to determine the grain boundary locations. The grains were grown using the appropriate lattice parameter (4.05 Å for aluminum) and grain orientations. A critical parameter to ensure was the atomic spacing at the grain boundaries. With the construction of multiple grains, areas where grains meet have atoms located in close proximity with no apparent structure. To ensure a reasonable initial structure, the atomic spacing at the grain boundaries was prescribed to equal or exceed the lattice parameter. The PreWarp input file for the creation of the triple junction structures is located in Appendix D and a table of the orientation angles used by PreWarp is located in Appendix E.

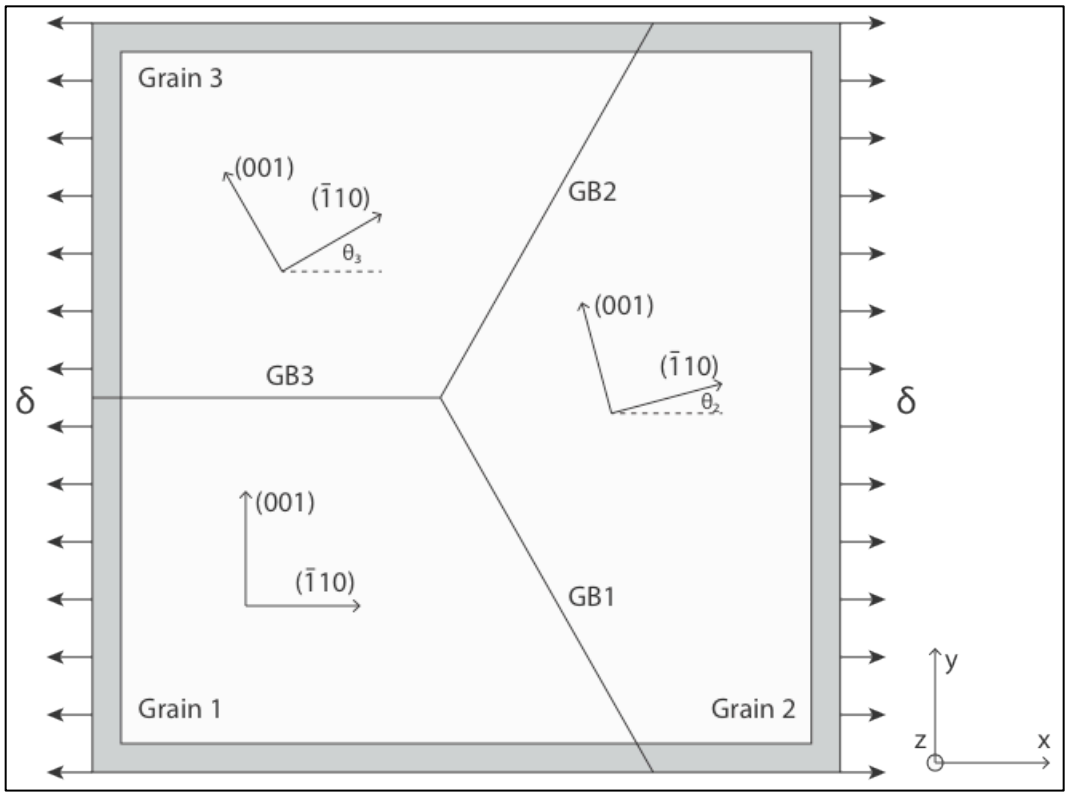

Figure 4.16 The schematic of the aluminum triple junction structure including the grain orientation and boundary conditions.

Each grain portion represented a section of a perfectly hexagonal grain with a longest diagonal of approximately 35 nm. The schematic of the uniaxial simulation is shown in Figure 4.16. The thickness was a multiple of the primitive unit cell for the defined texture to accommodate periodic boundary conditions in the z-direction. The simulations are approximately 530,000 atoms and have a box size of 400 Å (wide) x 400 Å (high) x 56.7 Å (thick). The various orientations for each grain are shown in Table 4.3.

Table 4.3 The orientation angles and texture of the four triple junction simulations.

Name	Rotation Angle			Texture
	Grain 1 (θ_1)	Grain 2 (θ_2)	Grain 3 (θ_3)	
Al_100_High	0	30	60	(100)
Al_100_Low	0	5	10	(100)
Al_110_High	0	30	60	(110)
Al_110_Low	0	5	10	(110)

To simplify the simulation, most structures had (110) textures to create a pseudo two-dimensional plain strain problem. The (110) orientation allowed for a smaller thickness while capturing the correct mechanisms of full three-dimensional atomistic simulations including four active slip systems (Horstemeyer *et al.*, 1999; Yamakov *et al.*, 2003; Yamakov *et al.*, 2002; Yamakov *et al.*, 2001; Zhu *et al.*, 2004). Under uniaxial straining, a 5 Å boundary along the edges of the x-y plane was held fixed in the direction of applied strain. The sample could also shrink in the two unstrained directions in response to Poisson's effects. The uniaxial straining implemented periodic boundary conditions for only the z-direction. By isolating a single triple junction and fixing the exterior boundary, we effectively removed the influence of anisotropic neighboring grains and cumulative grain motion. Analyzing the stress necessary to nucleate damage near a defect became the key focus. Since aluminum is nearly isotropic, the influence of neighboring grains should be mitigated. However, it is necessary to keep in mind we are neglecting some of the grain's ability to rotate.

The same simulation parameters used in the simple shear simulations, such as EAM potential (Mendelev *et al.*, 2008) and NPT ensemble and coefficients, were also used for the triple junction simulations. The relaxation of each simulation occurred by ramping the temperature from 1K to 300K and minimizing the pressure in the y- and z-directions over 50,000 time steps while allowing the pressure in the x-direction to equlibrate. Fixing the outer boundary maintained the square structure of the simulation and held the box size constant in the x-direction, so there were some residual stresses built up with an increase in temperature in the x-direction. The purpose of performing such a relaxation was to retain the polycrystalline structure. The input file for the relaxation of the triple junction structures is located in Appendix F.

After 50,000 time steps, the pressure was minimized in the x- and z-directions while it was allowed to equilibrate in the y-direction without a change in the simulation size in the y-direction for 10,000 time steps. Subsequently, fixing the x- and y-directions of the boundary, each for 10,000 time steps, performed an iterative process such that the stress went to zero in the x- and y-directions. This alternating process was performed until the entire simulation reached 90,000 time steps. The purpose was to “anneal” the structure and relax the grain boundaries and other defects.

Once relaxed, a displacement-controlled deformation, was applied to all atoms establishing uniaxial straining in the direction of deformation at a strain rate of 1×10^{9} sec^{-1}. The total deformation applied to the triple junction structure was a nominal strain of 15% over a time interval of 1,500 picoseconds. The input

file for the uniaxial tension simulation of the triple junction structures is located in Appendix G.

4.2.2 Results

The results illustrate the use of typical atomistic measures of deformation (i.e. centrosymmetry) and compare them with the strain measures introduced in the preceding chapter. The stress-strain response for the four simulations is shown in Figure 4.17. The maximum stresses for the (100) high (3.3 GPa), (100) low (3.1 GPa), (110) high (3.1 GPa), and (110) low (3.0 GPa) had no significant difference. However, the strains where peak stresses occurred for the (100) high (8.4%), (100) low (10.5%), (110) high (6.7%), and (110) low (9.0%). The (100) high-angle simulation had a slightly larger peak stress and corresponding strain than the (110) high-angle simulation. Similarly, the (100) low-angle simulation had a slightly larger peak stress and corresponding strain than the (110) low-angle simulation. The high-angle grain boundary simulations had larger peak stresses and smaller corresponding strains than their respective low-angle grain boundary simulations.

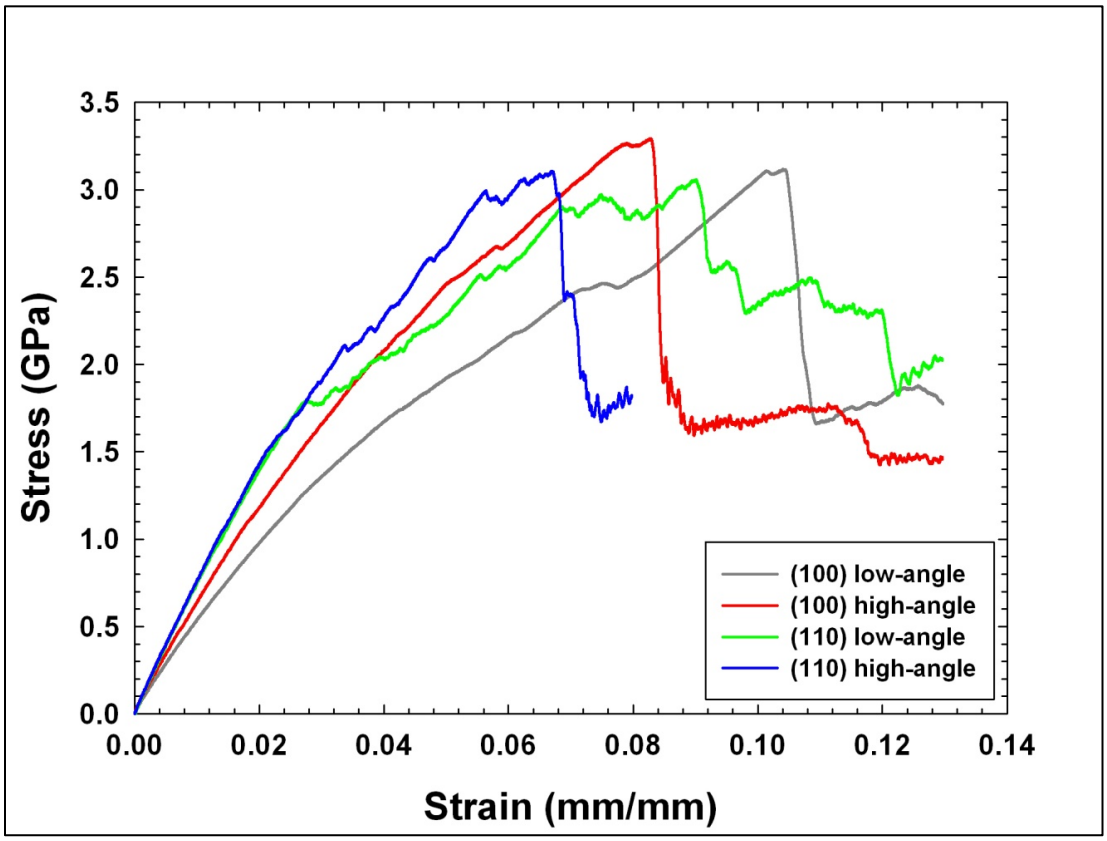

Figure 4.17 The uniaxial stress-strain curves of the four triple junction simulations labeled by their high- and low-angle grain boundaries and their orientations of (100) and (110).

The centrosymmetry parameter for the (100) high-angle grain boundary simulation and (100) low-angle grain boundary simulation at initial strain and before failure is shown in Figure 4.18. Figure 4.18a illustrates the (100) high-angle simulation has clear grain boundaries while Figure 4.18b shows the (100) low-angle simulation is not clearly defined. In Figure 4.18c and 4.18d, the centrosymmetry parameter does not illustrate any deviations from a normal FCC lattice before failure.

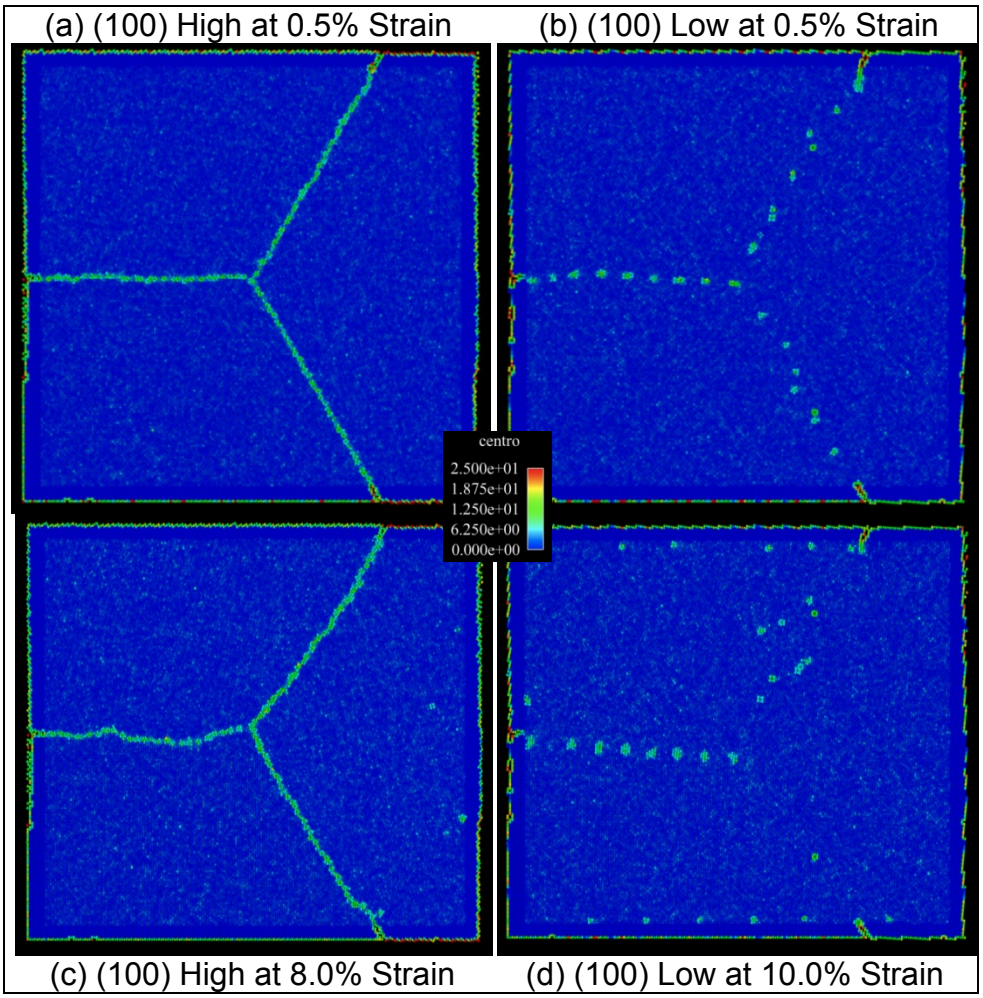

Figure 4.18 The centrosymmetry parameter for the (a) (100) high-angle at 0.5% strain, (b) (100) low-angle at 0.5% strain, (c) (100) high-angle at 8.0% strain, and (d) (100) low-angle at 10.0% strain.

This is also illustrated in Figure 4.19 with the (110) high-angle grain boundary simulation and (110) low-angle grain boundary simulation. Figure 4.19a illustrates the (110) high-angle simulation has definite grain boundaries while the boundary for the (110) low-angle is not clearly defined as shown in Figure 4.19b. However, the centrosymmetry parameter does capture some

dislocations in the (110) simulations directly before failure as shown in Figure 4.19c and 4.19d.

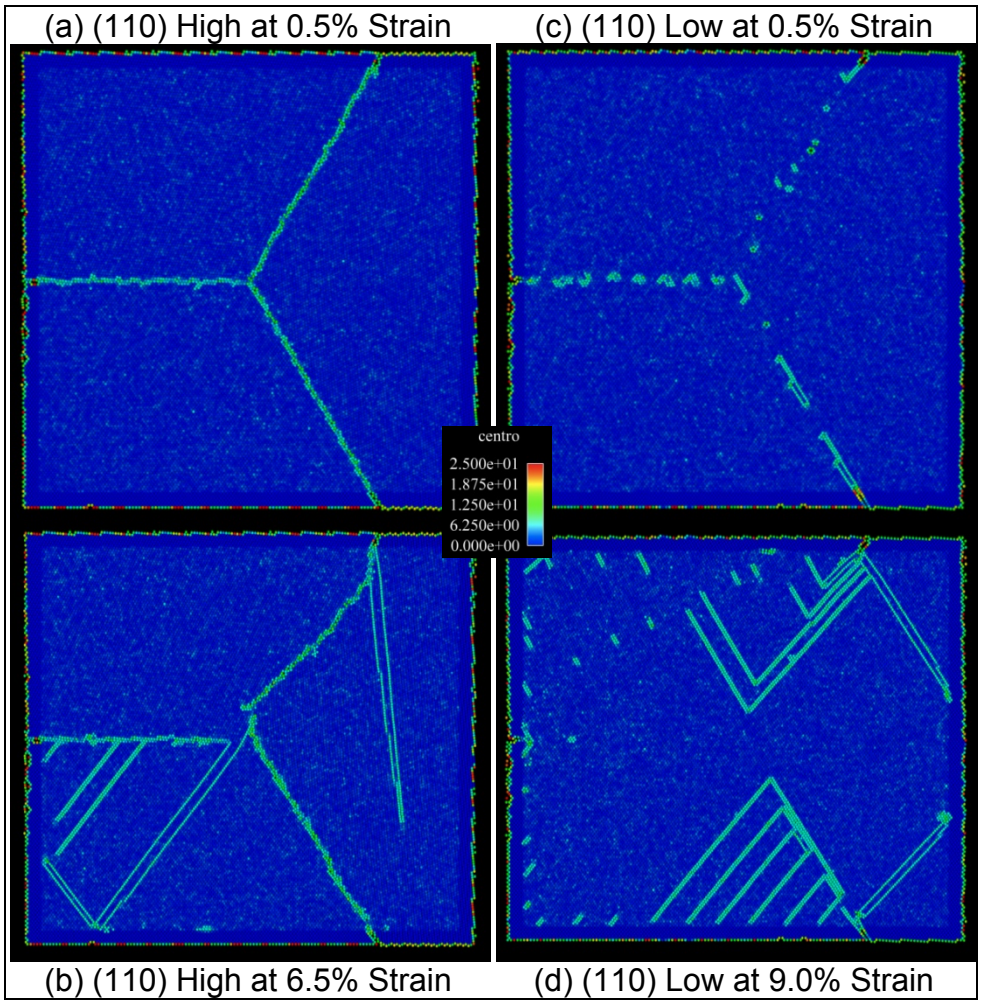

Figure 4.19 The centrosymmetry parameter for the (a) (110) high-angle at 0.5% strain, (b) (110) low-angle at 0.5% strain, (c) (110) high-angle at 6.5% strain, and (d) (110) low-angle at 9.0% strain.

The xx-component of the Green strain was investigated because of its ability to capture local deformation missed by the centrosymmetry parameter. Figure 4.20 depicts the (100) high-angle and low-angle grain boundary

simulation. At initial loading, shown in Figures 4.20a and 4.20b, the xx-component is zero. Directly before failure, shown in Figure 4.20c and 4.20d, dislocations are evident in these simulations.

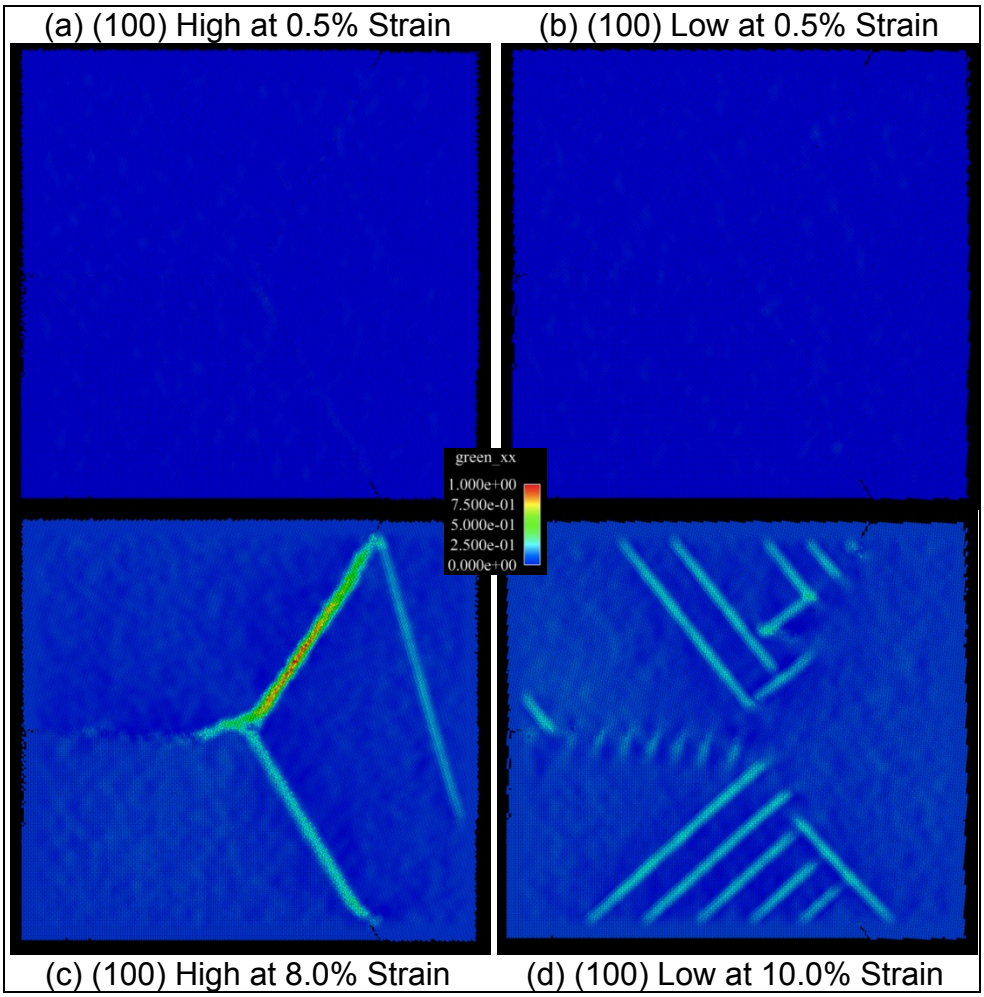

Figure 4.20 The xx-component of the Green strain for the (a) (100) high-angle at 0.5% strain, (b) (100) low-angle at 0.5% strain, (c) (100) high-angle at 8.0% strain, and (d) (100) low-angle at 10.0% strain.

Figure 4.21 depicts the (110) high-angle and the (110) low-angle grain boundary simulation. At initial loading, shown in Figures 4.21a and 4.21b, the xx-

component is zero for the (110) high-angle and (110) low-angle simulations. Directly before failure, shown in Figure 4.21c and 4.21d, dislocations are evident in these simulations that were displayed with the centrosymmetry parameter.

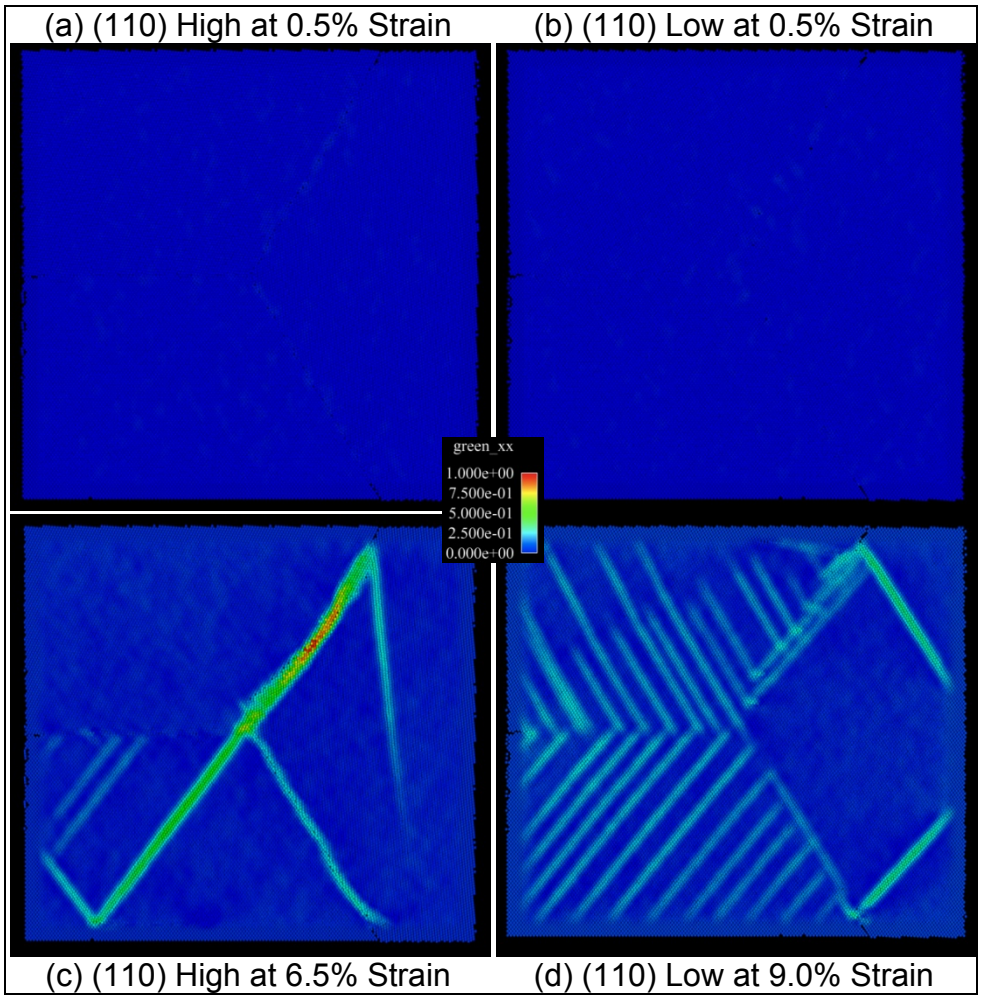

Figure 4.21 The xx-component of the Green strain for the (a) (110) high-angle at 0.5% strain, (b) (110) low-angle at 0.5% strain, (c) (110) high-angle at 6.5% strain, and (d) (110) low-angle at 9.0% strain.

Given the success of the Green strain tensor, it was a natural movement to apply other metrics that involve the deformation gradient. In particular, the use

of the determinant of the deformation gradient was of interest. It has the ability to measure the expansion and contraction around single atoms, which is exactly what happens in the case of void nucleation. Shown below in Figure 4.22 are images of the Jacobian for the (100) high-angle simulation.

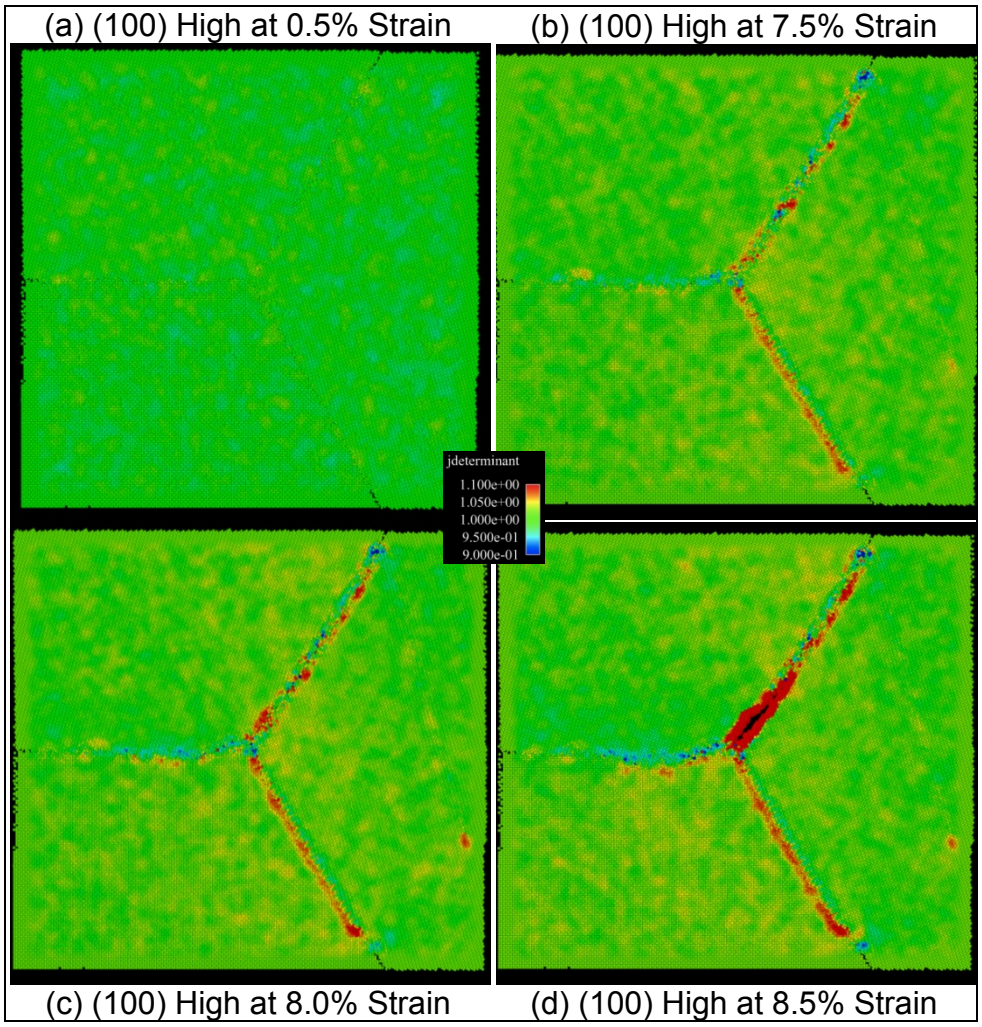

Figure 4.22 The determinant of the deformation gradient for the (100) high-angle simulation at (a) 0.5% strain, (b) 7.5% strain, (c) 8.0% strain, and (d) 8.5% strain.

The Jacobian for the (110) high-angle simulation illustrates expansion around the triple junction, as shown below in Figure 4.23. Figure 4.23c shows an expansive area at the triple junction and Figure 4.23d illustrates a void at that precise location. Interestingly, the increase of the Jacobian along GB2 in Figure 4.23d does not lie on top of the grain boundary.

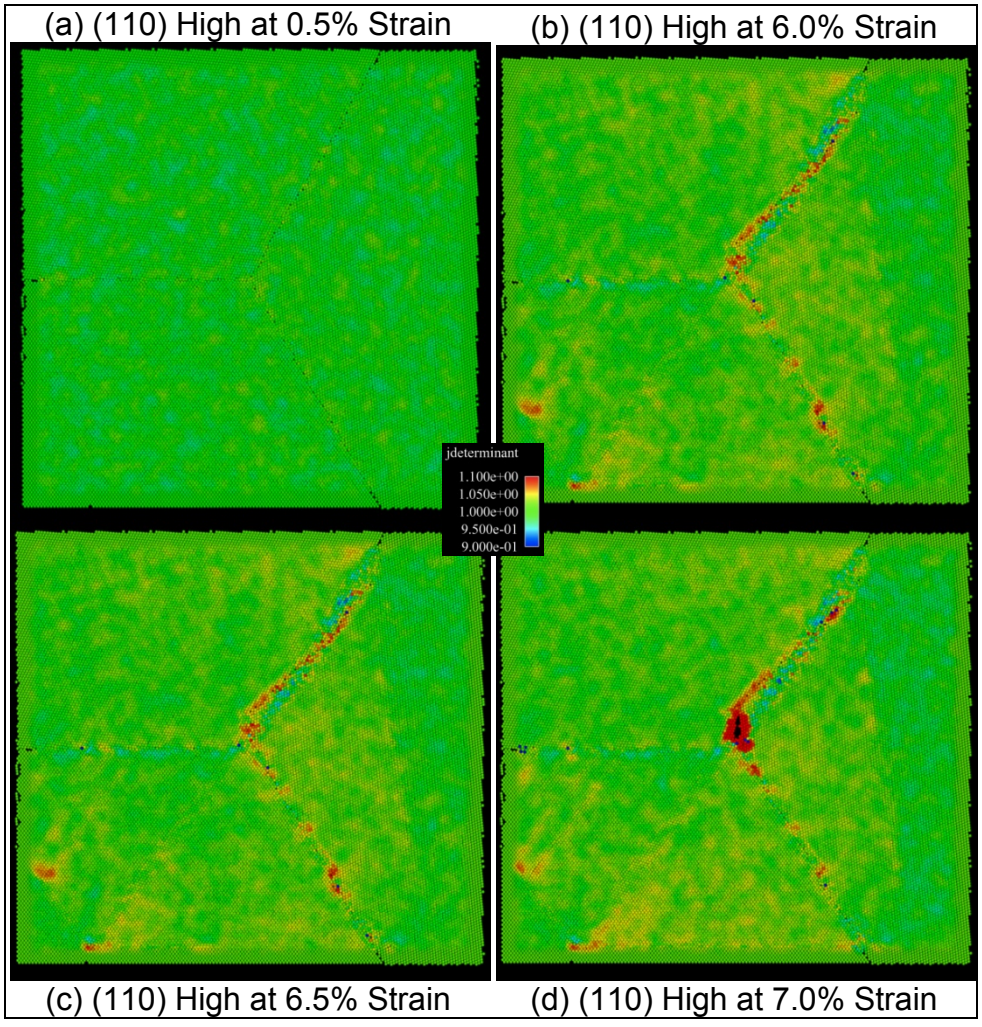

Figure 4.23 The determinant of the deformation gradient for the (110) high-angle simulation at (a) 0.5% strain, (b) 6.0% strain, (c) 6.5% strain, and (d) 7.0% strain.

The xy-component of the spin tensor is shown in Figure 4.24 for the (100) high-angle simulation. In Figure 4.24a, the grain boundaries have a clearly defined spin value while the grains have small gradients of spin dispersed throughout. In Figures 4.24b and 4.24c, the rotations at the triple junction are in opposite directions, implying those areas are pulling away from each other.

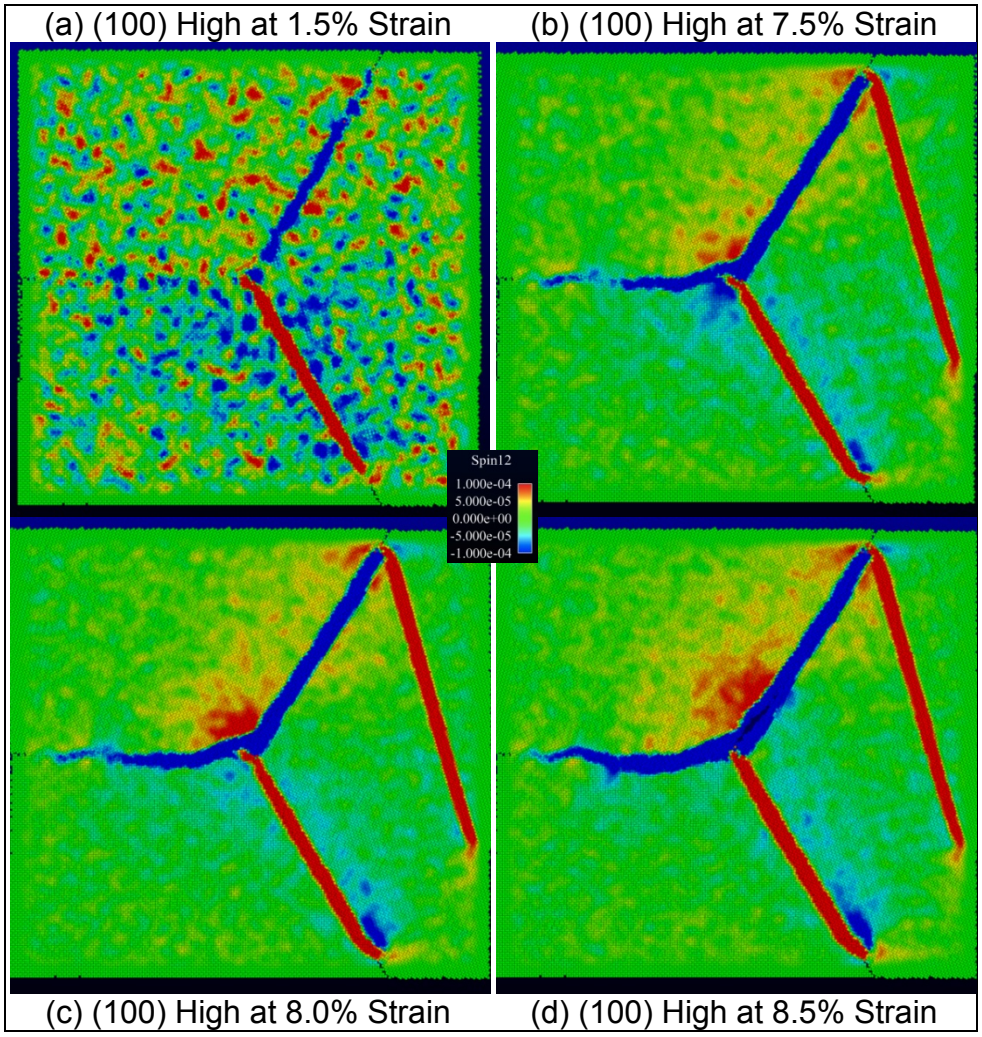

Figure 4.24 The xy-component of the spin tensor for the (100) high-angle simulation at (a) 1.5% strain, (b) 7.5% strain, (c) 8.0% strain, and (d) 8.5% strain.

The xy-component of the spin tensor for the (110) high-angle simulation illustrates rotation around the triple junction, as shown below in Figure 4.25. Figure 4.25c shows GB1 and GB2 having rotations in opposite directions and Figure 4.25d illustrates a void at the triple junction and then propagated vertically.

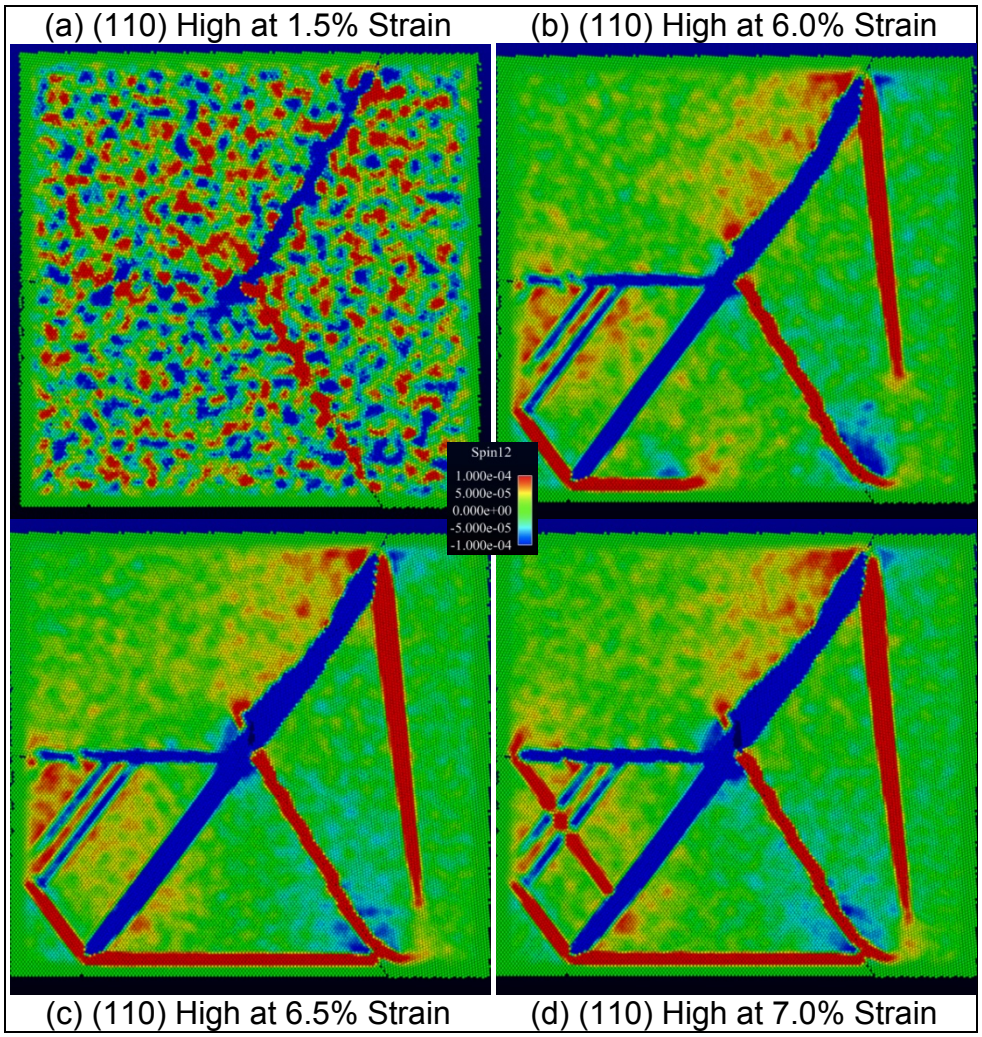

Figure 4.25 The xy-component of the spin tensor for the (110) high-angle simulation at (a) 1.5% strain, (b) 6.0% strain, (c) 6.5% strain, and (d) 7.0% strain.

On the other hand, the deviatoric stress is a widely used measure to indicate local areas of high stress. However, the deviatoric stress tensor does not indicate the triple junction as the location of highest stress. Figure 4.26 indicates the deviatoric stress plays no role in the nucleation of a void at the triple junction for the (100) high-angle simulations. The deviatoric stress is generally higher in grain 3 and on the right boundary, shown in Figure 4.26c and 4.26d.

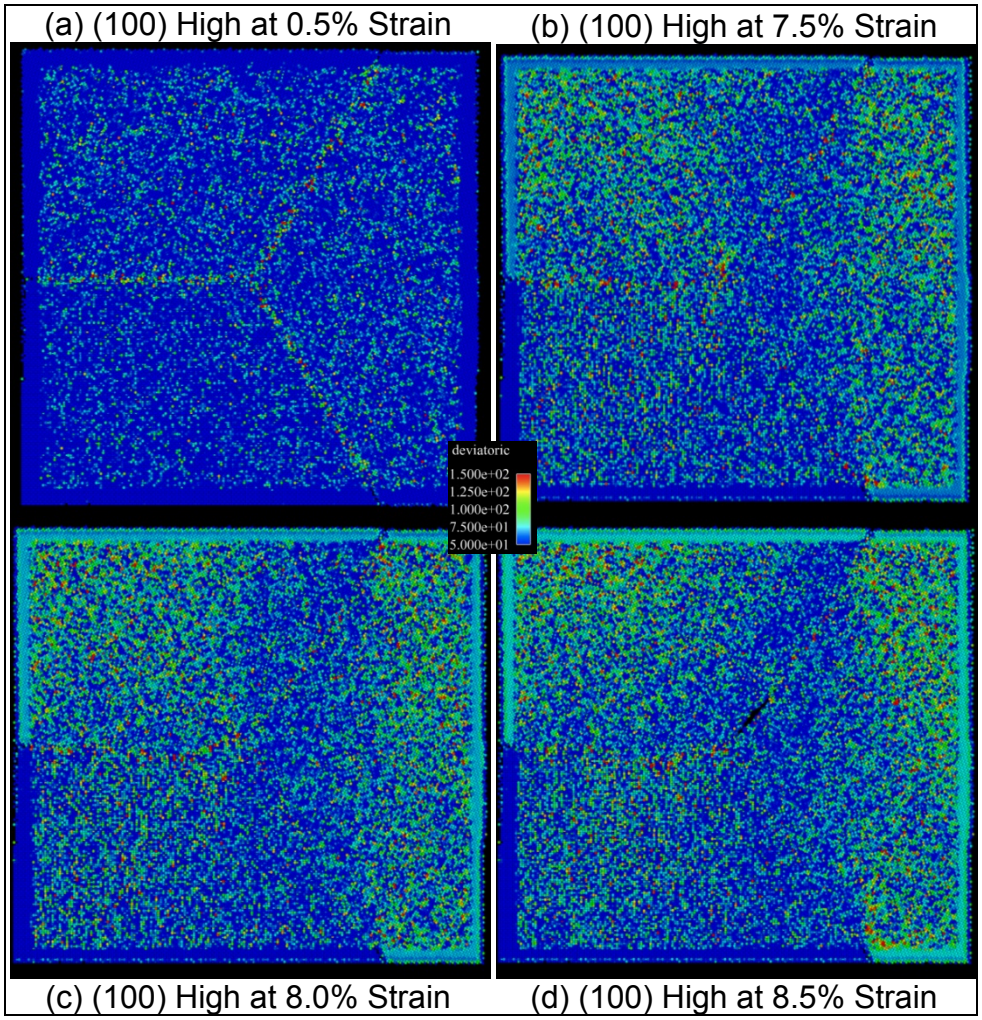

Figure 4.26 The deviatoric stress for the (100) high-angle simulation at (a) 0.5% strain, (b) 7.5% strain, (c) 8.0% strain, and (d) 8.5% strain.

Similarly, the deviatoric stress does not indicate a pattern for the (110) high-angle simulations either. In Figure 4.27c and 4.27d, the stress is generally higher on the right boundary, as shown in the (100) simulation. However, the use of a stress metric to indicate possible areas of failure in atomistic triple junctions yields little results.

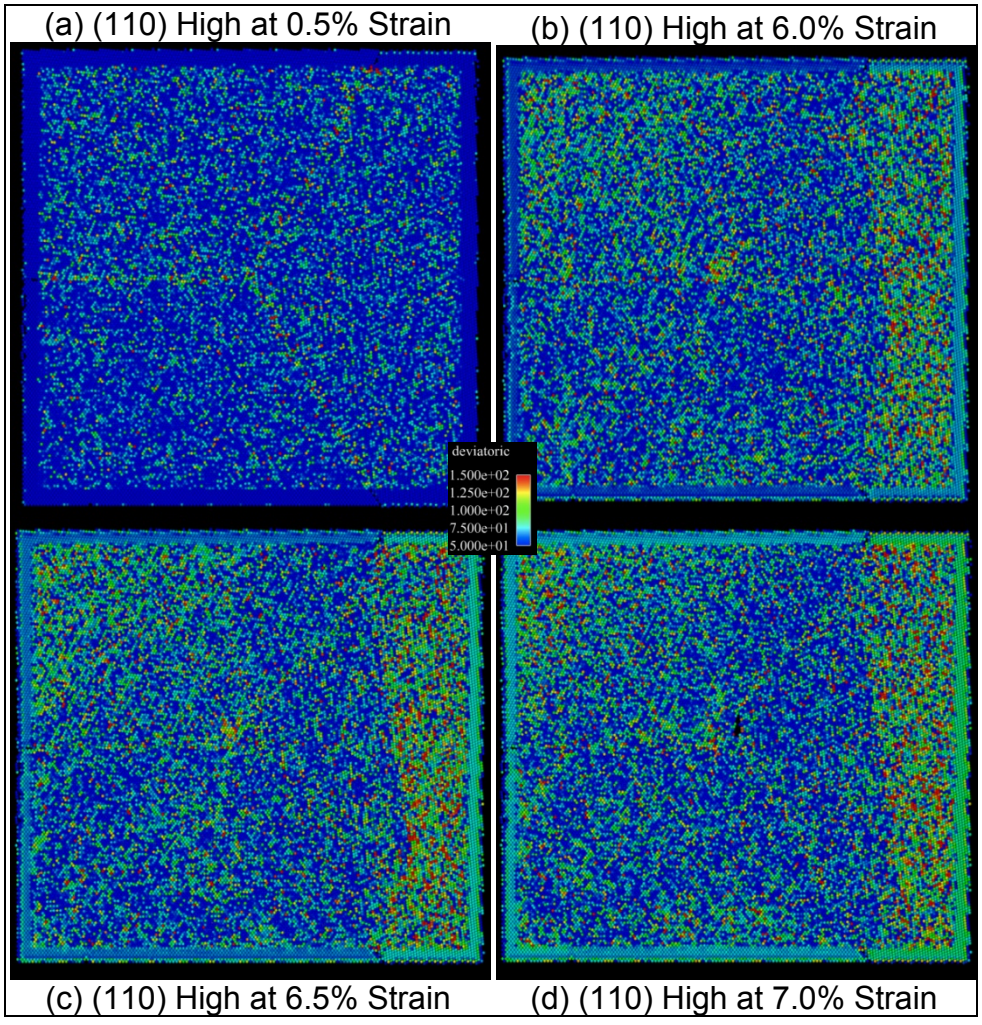

Figure 4.27 The deviatoric stress for the (110) high-angle simulation at (a) 0.5% strain, (b) 6.0% strain, (c) 6.5% strain, and (d) 7.0% strain.

To provide another method of comparison for the four triple junction simulations, the Schmid factor was calculated. The Schmid factor is a factor used to characterize the value of slip in any given crystal. The FCC lattice of aluminum has 12 slip systems and each one has a corresponding Schmid factor. Each grain is characterized by the Schmid factor of highest value. These are shown below in Table 4.4. Notice the Schmid factors for the high-angle simulations have a wider range than the Schmid factors for the low-angle simulations.

Table 4.4 The Schmid factors for each grain of the four simulations.

Orientation	Grain 1	Grain 2	Grain 3
(110) High	0.408	0.334	0.433
(110) Low	0.408	0.429	0.433
(100) High	0.423	0.433	0.349
(100) Low	0.423	0.43	0.445

4.2.3 Conclusions

The purpose of these simulations was to determine a metric that could be utilized to indicate or characterize void nucleation. The centrosymmetry parameter was little help in visualizing areas of local deformation and void nucleation and the stress-based values did a poor job of capturing local deformation and predicting void nulceation. However, the use of deformation gradient-based measures, such as the Green strain or Jacobian of the deformation gradient, in atomistic simulations was successful in visualizing local deformation and predicting void nucleation.

CHAPTER V

CRYSTAL PLASTICITY SIMULATIONS

5.1 Crystal Plasticity Overview

To learn more about the grain boundary triple junctions and for the purpose of comparison, the finite element method coupled with crystal plasticity was used to analyze the stress-state of the simulations. Multiscale modeling encompasses microscale approaches as well as mesoscale approaches. One of the mesoscale approaches used for material modeling is crystal plasticity modeling (Kocks *et al.*, 1998). Crystal plasticity models have the capability of modeling individual grains and their orientations to determine, for example, the stress state of the material (Asaro, 1983). Since crystal plasticity models input the grain orientation and slip systems, these models account for a material's anisotropy and texture evolution. The formulation and implementation of the crystal plasticity model used for this research was done by Marin (2006) and is discussed in detail in the previously mentioned report.

5.2 Simulation Overview

The crystal plasticity subroutine was coupled with Abaqus (6.9-1), a commercial finite element analysis (FEA) software package, to run the triple

junction simulations. The implicit integration scheme, commonly referred to as Abaqus/Standard, was used for these simulations (Hibbitt *et al.*, 2009). The dimensions of the three-dimensional triple junction simulation are 400 millimeters (mm) (wide) x 400 mm (high) x 100 mm (thick) and it consists of 33,040 linear hexahedral elements. In comparison, these dimensions are 1 x 10^7 larger in magnitude than the dimensions of the molecular dynamic simulations. However, the use of the crystal plasticity was to calculate the granular rotations and determine the location of increased stress and strain, which are not affected by the size of the simulation.

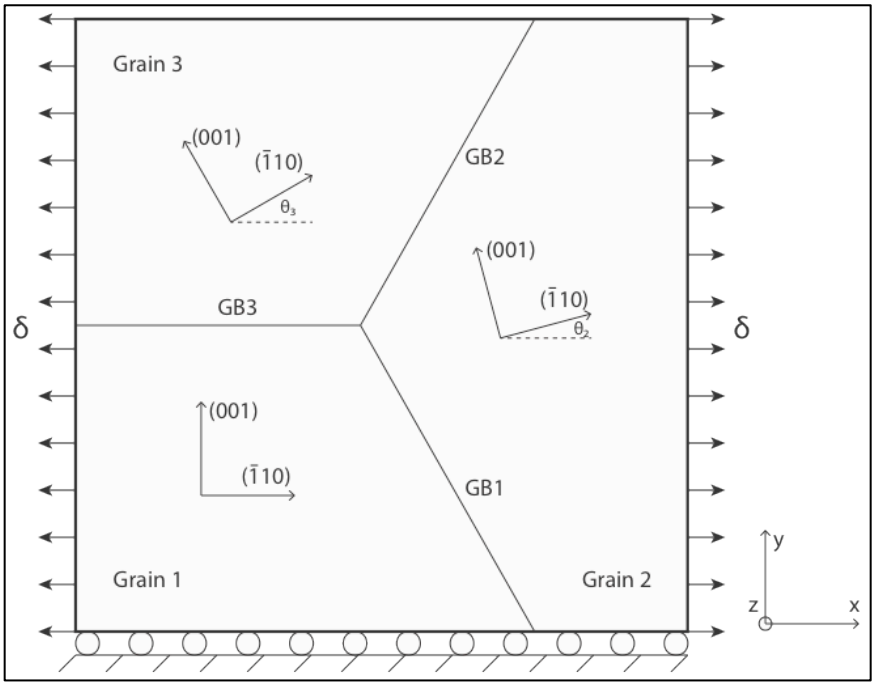

Figure 5.1 The schematic of the aluminum triple junction crystal plasticity simulations including the grain orientations and boundary conditions.

The simulations were run on a Sun Ultra 27 Workstation with an Intel® Xeon® W3570 3.20 GHz quad-core processor with the SuSE Linux Enterpise Desktop 10 operating system.

The boundary conditions applied to the system included a velocity-controlled boundary condition on the positive and negative x-face to simulate uniaxial tension. The negative y-face was constrained to movement in the x- and z- direction only, as shown in Figure 5.1. The total deformation applied to the triple junction structure was a nominal strain of 15% over a time interval of 150 seconds. The applied strain rate for the simulation is $1x10^{-1}$ sec^{-1}. The material properties for the crystal plasticity simulations of aluminum are shown in Table 5.1. The table includes the elastic parameters, specified for an anisotropic cubic lattice, and the viscoplasticity parameters. More detailed information about theses parameters and their use in the model can be found in Marin (2006).

Table 5.1 Material properties for aluminum crystal plasticity simulations.

Elasticity Parameters

C11	C12	C44
108.2 GPa	61.3 GPa	28.5 GPa

Viscoplasticity Parameters

m	$\dot{\gamma}_0$	h_0	$\kappa_{s,0}$	$\kappa_{s,S0}$	m'	$\dot{\gamma}_{S0}$
0.05	1.0 s^{-1}	204.0 MPa	205.0 MPa	290.0 MPa	0.0	5 x 10^{10} s^{-1}

The rotation angles for the grains of each of the four simulations are specified in Table 4.3. The crystal plasticity model requires the angles to be in Euler angle notation using Kock's convention. The Euler angles for the (110)

texture are shown in Table 5.2 and the Euler angles for the (100) texture are shown in Table 5.3.

Table 5.2 The Euler angles (Ψ,Θ,Φ) for (110) texture and specified rotation angle (θ).

θ	Ψ	Θ	Φ
0	0	90	45
5	5	90	45
10	10	90	45
15	15	90	45
30	30	90	45
60	60	90	45

Table 5.3 The Euler angles (Ψ,Θ,Φ) for (100) texture and specified rotation angle (θ).

θ	Ψ	Θ	Φ
0	-90	0	-180
5	-90	0	-175
10	-90	0	-170
15	-90	0	-165
30	-90	0	-150
60	-90	0	-120

5.3 Results

The main objectives of the crystal plasticity simulations were to (i) make qualitative comparisons between the crystal plasticity and MD simulations and (ii) analyze the development of the texture, which is routinely done in crystal plasticity simulations. The (100) high-angle grain boundary simulations has elevated von Mises stress along GB2, as shown in Figure 5.2.

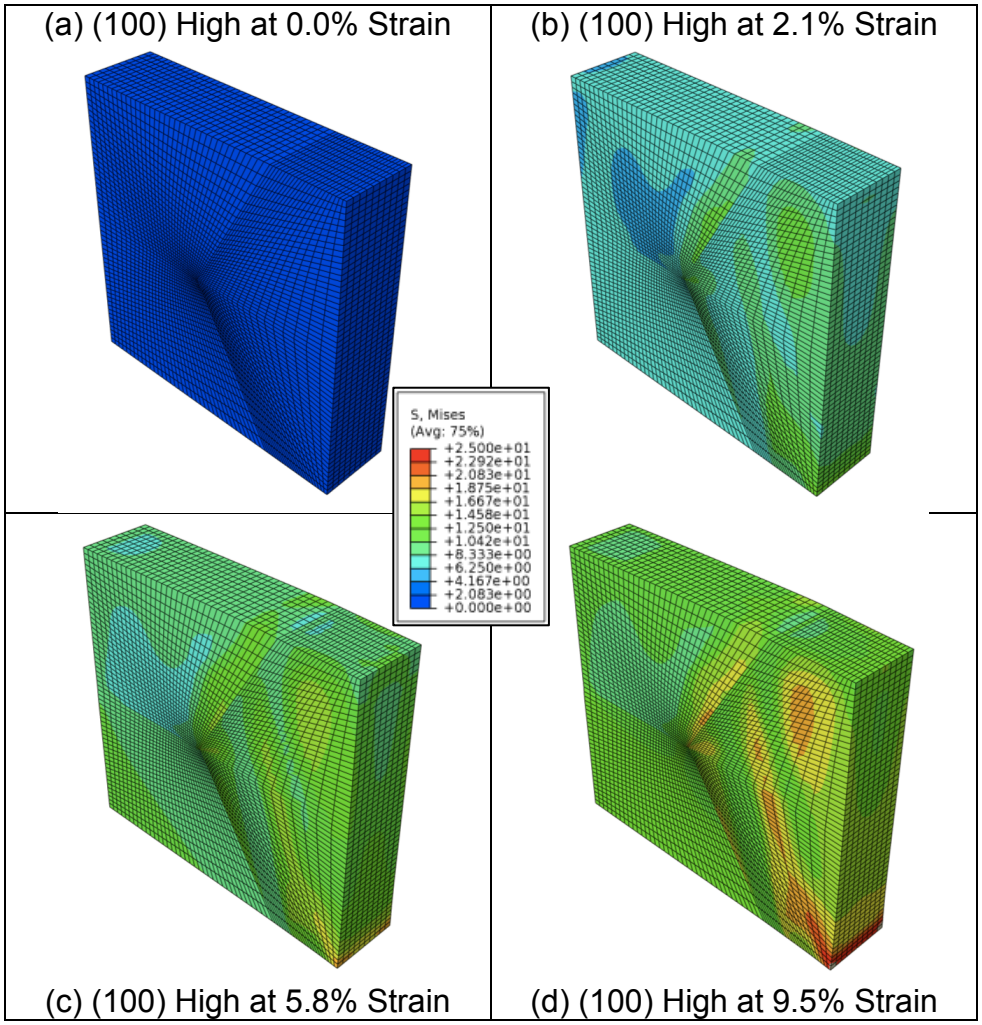

Figure 5.2 The von Mises stress (MPa) for the (100) high-angle simulation at (a) 0.0% strain, (b) 2.1% strain, (c) 5.8% strain, and (d) 9.5% strain.

Figure 5.2d is of particular interest because the molecular dynamic simulation of the (100) high-angle configuration nucleated a void at approximately 7.5% strain. If we disregard the elevation of stress around the bottom-right corner as a result of the FEA boundary conditions, the location of increased stress close to the triple junction on GB2 is the location of void

nucleation in the MD simulation. However, by looking at the plastic strain evolution pictured in Figure 5.3, we can conclude there is no such correlation between the area of increased von Mises stress and plastic strain.

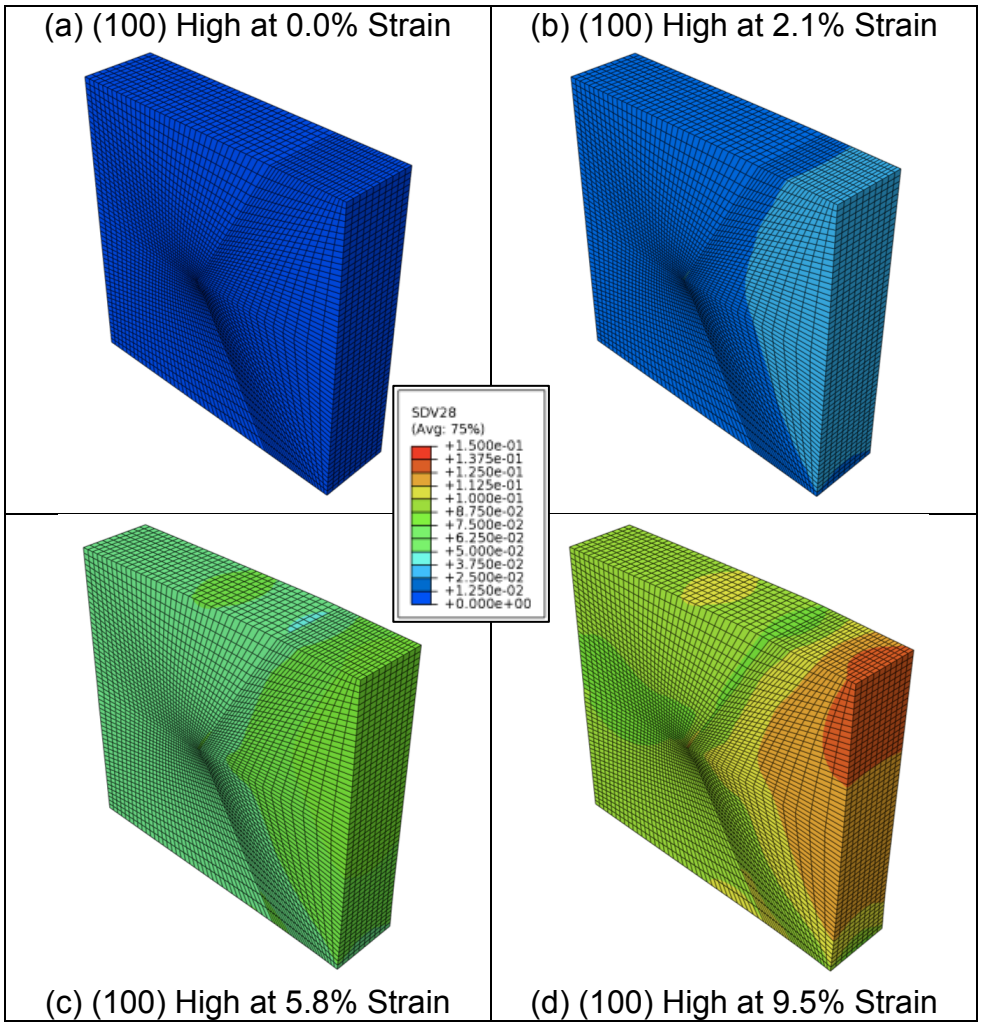

Figure 5.3 The plastic strain (mm/mm) for the (100) high-angle simulation at (a) 0.0% strain, (b) 2.1% strain, (c) 5.8% strain, and (d) 9.5% strain.

A similar analysis by Querin *et al.* (2007) concluded that an increase in hydrostatic stress, or pressure, was an important factor in the creation of voids at

the triple junction. By illustrating the hydrostatic pressure in Figure 5.4c and Figure 5.4d, we notice there is a noticeable jump at GB2 between the hydrostatic stress on either side of the grain boundary. Therefore, for this particular simulation, our results align with the results from Querin.

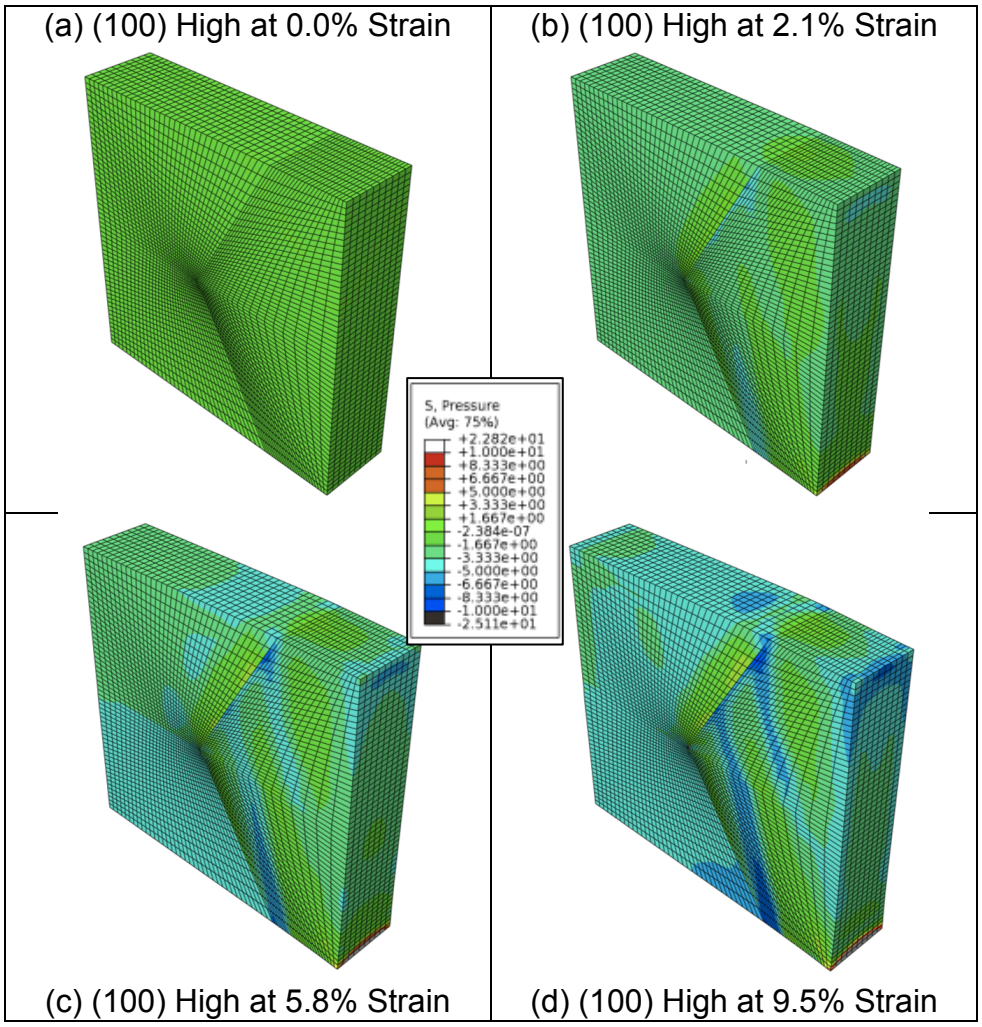

Figure 5.4 The hydrostatic stress (MPa) for the (100) high-angle simulation at (a) 0.0% strain, (b) 2.1% strain, (c) 5.8% strain, and (d) 9.5% strain.

Similarly, for the (110) high-angle grain boundary simulations, the von Mises stress is elevated around GB2 and GB3, as shown in Figures 5.5a and 5.5b. More importantly, the hydrostatic stress is nearly constant over the simulation at 1.8% strain but develops local maximums at 9.3% strain, as shown in Figures 5.5c and 5.5d.

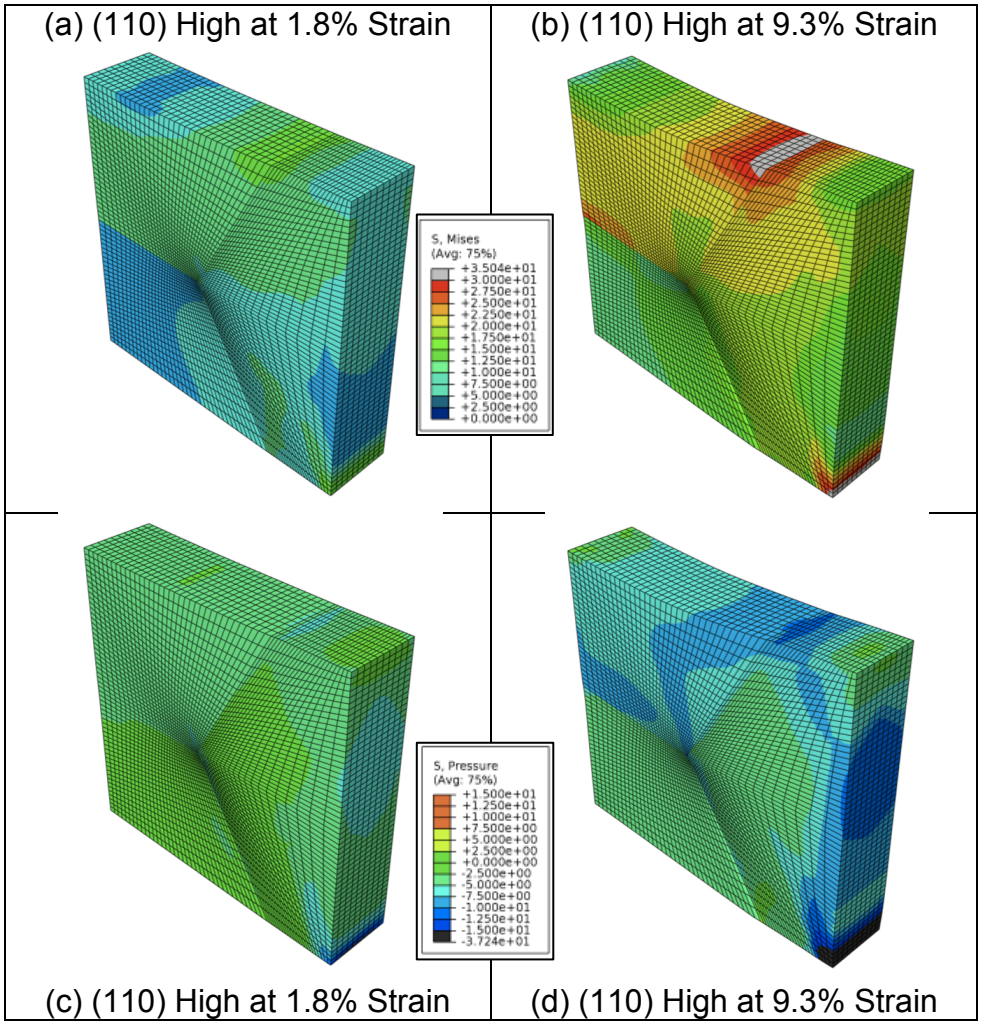

Figure 5.5 The (110) high-angle simulation with von Mises stress (MPa) shown at (a) 1.8% strain and (b) 9.3% strain as well as pressure (MPa) shown at (c) 1.8% strain and (d) 9.3% strain.

To contrast the (110) texture simulations, the (100) low-angle grain boundary simulation is shown below in Figure 5.6. The von Mises stress, Figures 5.6a and 5.6b, show little variation at 2.1% and 9.5% strain. Also, the hydrostatic pressure is constant over the simulation for the same strain levels, as shown in Figures 5.6c and 5.6d. This is intriguing because neither of the low-angle grain boundary simulations nucleated a void at a grain boundary.

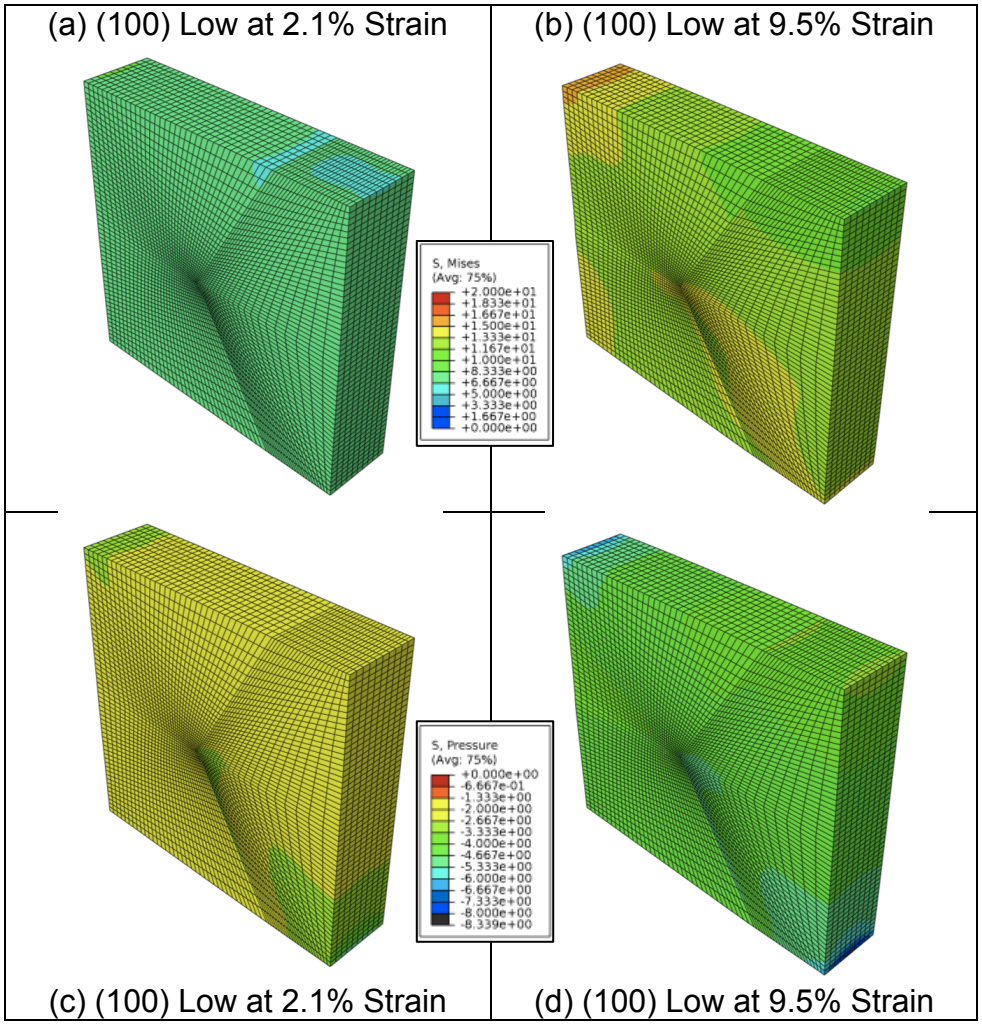

Figure 5.6 The (100) low-angle simulation with von Mises stress (MPa) shown at (a) 2.1% strain and (b) 9.5% strain as well as pressure (MPa) shown at (c) 2.1% strain and (d) 9.5% strain.

The other objective of the crystal plasticity simulations was to analyze the rotation of grains during the simulation. For the (100) high- and low- angle simulations, the rotation of grains is very large, as shown in Tables 5.4. Euler angle 2, denoted by Θ, maintained a value of zero while the other two angles changed by a wide range of magnitudes. This seems to indicate a major rotation in the grains during deformation.

Table 5.4 The Euler angles, in degrees, for the (100) high-angle and (100) low-angle simulation at 0.0% strain and 5.8% strain.

Grain	Angle	(100) High-angle GB		(100) Low-angle GB	
		0.0% Strain	5.8% Strain	0.0% Strain	5.8% Strain
1	Ψ	-90	42.63	-90	-19.84
	Θ	0	0.08	0	0.21
	Φ	-180	-14.44	-180	29.3
2	Ψ	-90	10.55	-90	-8.32
	Θ	0	0.2	0	0.16
	Φ	-150	-0.22	-175	12.95
3	Ψ	-90	-42.27	-90	-2.28
	Θ	0	0.15	0	0.21
	Φ	-120	35.69	-170	8

The (110) high- and low- angle grain boundary simulations tell a different story. The second and third Euler angles do not have bulk changes while the first Euler angle changes by a very small amount, as shown in Tables 5.5. For the (110) high-angle simulation, Euler angle 1 changes 1.38º for grain 1, 2.66º for grain 2, and 1.67º for grain 3. Similarly, for the (110) low-angle simulation, Euler angle 1 changes 0.06º for grain 1, 1.18º for grain 2, and 0.7º for grain 3.

Table 5.5 The Euler angles, in degrees, for the (110) high-angle and (110) low-angle simulation at 0.0% strain and 5.8% strain

Grain	Angle	(110) High-angle GB		(110) Low-angle GB	
		0.0% Strain	5.8% Strain	0.0% Strain	5.8% Strain
1	Ψ	0	-1.38	0	-0.06
	Θ	90	90	90	90
	Φ	45	45	45	45
2	Ψ	30	32.66	5	3.82
	Θ	90	90	90	90
	Φ	45	45	45	45
3	Ψ	60	58.33	10	9.3
	Θ	90	90	90	90
	Φ	45	45	45	45

Table 5.6 The Euler angles for the four elements located at the triple junction in each simulation at 5.8% strain.

Grain	Angle	(100) High	(100) Low	(110) High	(110) Low
1	Ψ	102.8	70.6	-1.44	0.04
	Θ	0.07	0.93	91.8	89.4
	Φ	11.2	-19.6	43.8	44.9
2 (top)	Ψ	-161.1	136.8	30	4.36
	Θ	0.03	0.88	91.4	89.7
	Φ	137.6	50.9	42.7	45.2
2 (bottom)	Ψ	132.4	107.9	29.8	4.37
	Θ	0.47	0.8	91.4	89.7
	Φ	70.64	22.2	89.6	44.9
3	Ψ	-132.2	144.9	57.9	9.25
	Θ	0.51	1.26	89.6	89.5
	Φ	-162.8	63.7	43.9	45.3

The Euler angles of the four elements located at the triple junction at 5.8% strain are shown in Table 5.6. The Euler angles show a large deviation in the initial values for the (100) high- and low-angle simulations while the deviations are small in the (110) high- and low- angle simulations.

5.4 Conclusions

The uniaxial tension simulations of triple junction simulations using a crystal plasticity model were setup very similarly to the molecular dynamics simulations. For the (100) high-angle simulation, the von Mises stress and the hydrostatic stress are elevated at the location of the nucleation of a void, while the plastic strain revealed no correlation between strain and the location of the nucleated void. For the (110) high-angle simulation, the hydrostatic stress is elevated at the triple junction but the von Mises stress is not. For the (100) low-angle grain boundary simulation, there is no concentration of von Mises or hydrostatic stress, which agrees with the MD simulation that did not nucleate a void. The stress values were the metric that provided the most information in the crystal plasticity simulations while strain values were more helpful in the molecular dynamic simulations.

CHAPTER VI

SUMMARY AND CONCLUSIONS

The overall goal of this research was to determine a method for calculating the plastic spin at the atomistic level for the purpose of multiscale-based constitutive modeling. Strides were made towards this goal with the pursuit of the two main objectives: (a) to quantify void nucleation in molecular dynamic simulations of TJ using a discrete mathematical framework and (b) to explore the possibility of calculating quantities integral to continuum-based failure models, such as the plastic spin.

A summary of the conclusions and plans for future work is described below.

- The previous research of the plastic spin was discussed, in detail, with special attention made to the configurations of the formulations of the constitutive equations for the plastic spin.
- Presently, a hypothesis for the calculation of the plastic spin has been provided, but needs additional research completed to determine the validity of the proposed plan of action.
- EAMpost was modified to use LAMMPS data and revised to include deformation metrics such as (1) the Jacobian of the deformation gradient,

(2) the velocity gradient, and (3) the spin tensor. Additional research needs to be completed to determine the importance and proper use of these metrics in regards to the atomistic level.

- Molecular dynamic simple shear simulation with and without a hole illustrated the importance of the use of local deformation measures in MD simulations. The difference in the local deformation in the simulations was clearly noticed with the use of the metrics listed above.
- The molecular dynamic triple junction simulations nucleated voids in the (100) and (110) high-angle grain orientation while voids did not nucleate in the (100) and (110) low-angle grain orientation simulations. The Green strain and Jacobian of the deformation gradient show promise as advanced local deformation measures to quantify void nucleation at the atomistic level. Future work should be done to incorporate the spin tensor analysis with the triple junction simulations once.
- Crystal plasticity simulations of the triple junction were able to illustrate the increase in von Mises and hydrostatic stress along the grain boundaries for the (100) and (110) high-angle simulation. The (100) low-angle simulation revealed no local gradients in the stress values. However, the plastic strain did not reveal as much information as hypothesized, therefore future work with the code is proposed to determine the validity of these simulations.

REFERENCES

Aravas, N. (1994). Finite-strain anisotropic plasticity and the plastic spin. *Modelling and Simulation in Materials Science and Engineering, 2*(3A), 483.

Asaro, R. J. (1983). Micromechanics of Crystals and Polycrystals. In W. H. John & Y. W. Theodore (Eds.), *Advances in Applied Mechanics* (Vol. 23, pp. 1-115): Elsevier.

Bammann, D. J. (1984). An internal variable model of viscoplasticity. *International Journal of Engineering Science, 22*(8-10), 1041-1053.

Barber, J. R. (2002). *Elasticity*. Dordrecht; Boston: Kluwer Academic Publishers.

Bilby, B. A., Gardner, L. R. T., & Stroh, A. N. (1957). *Continuous distributions of dislocations and the theory of plasticity.* In: Extrait des Actes du IX^{e} Congrès International de Mécanique Appliquée, Bruxelles, 35–44.

Buehler, M. J. (2004). *Atomistic and Continuum Studies of Deformation and Failure in Brittle Solids and Thin Film Systems.* PhD Thesis, Max-Planck-Institut für Metallforschung, Stuttgart, Germany.

Bunge, H. J., & Nielsen, I. (1997). Experimental determination of plastic spin in polycrystalline materials. *International Journal of Plasticity, 13*(5), 435-446.

Clausius, R. J. E. (1870). On a mechanical theorem applicable to heat. *Philosophical Magazine, 40*, 122-127.

Clayton, J. D., McDowell, D. L., & Bammann, D. J. (2006). Modeling dislocations and disclinations with finite micropolar elastoplasticity. *International Journal of Plasticity, 22*(2), 210-256.

Dafalias, Y. F. (1983). Corotational Rates for Kinematic Hardening at Large Plastic Deformations. *Journal of Applied Mechanics, 50*(3), 561-565.

Dafalias, Y. F. (1984). The plastic spin concept and a simple illustration of its role in finite plastic transformations. *Mechanics of Materials, 3*(3), 223-233.

Dafalias, Y. F. (1985). The plastic spin. *Transactions of the ASME. Journal of Applied Mechanics, 52*(4), 865-871.

Dafalias, Y. F., & Aifantis, E. (1990). On the microscopic origin of the plastic spin. *Acta Mechanica, 82*(1), 31-48.

Dafalias, Y. F. (1998). Plastic spin: necessity or redundancy? *International Journal of Plasticity, 14*, 909-931.

Daw, M. S., & Baskes, M. I. (1984). Embedded-atom method: Derivation and application to impurities, surfaces, and other defects in metals. *Physical Review B, 29*(12), 6443.

Daw, M. S., Foiles, S. M., & Baskes, M. I. (1993). The embedded-atom method: a review of theory and applications. *Material Science Reports, 9*(7-8), 251-310.

Dieter, G. E. (1976). *Mechanical metallurgy*. New York: McGraw-Hill.

Frenkel, D., & Smit, B. (2002). *Understanding molecular simulation : from algorithms to applications*. San Diego: Academic Press.

Friedel, J. (1952). The distribution of electrons round impurities in monovalent metals. *Philosophical Magazine, 43*, 153-189.

Gullett, P. M., Slepoy, A., Horstemeyer, M. F., Fang, H., Baskes, M. I., Wagner, G. J., et al. (2004). *Numerical Tools for Atomistic Simulations*: Sandia National Laboratories.

Gullett, P. M., Horstemeyer, M. F., Baskes, M. I., & Fang, H. (2008). A deformation gradient tensor and strain tensors for atomistic simulations. *Modelling and Simulation in Materials Science and Engineering, 16*(1), (17 pp.).

Hibbitt, K., & Sorensen (2009). *ABAQUS/Standard : user's manual*. Providence, R.I.: Hibbitt, Karlsson & Sorensen.

Holzapfel, G. A. (2000). *Nonlinear solid mechanics : a continuum approach for engineering*. Chichester; New York: Wiley.

Hoover, W. G. (1985). Canonical dynamics: Equilibrium phase-space distributions. *Physical Review A, 31*(3), 1695.

Hoover, W. G. (1986). Constant-pressure equations of motion. *Physical Review A, 34*(3), 2499.

Horstemeyer, M. F. (1995). *Physically-Motivated Modeling of Deformation-Induced Anisotropy.* PhD Thesis, Georgia Institute of Technology, Atlanta, GA.

Horstemeyer, M. F., & Gokhale, A. M. (1999). A void-crack nucleation model for ductile metals. *International Journal of Solids and Structures, 36*, 5029-5055.

Horstemeyer, M. F., Baskes, M. I., & Plimpton, S. J. (2001). Computational nanoscale plasticity simulations using embedded atom potentials. *Theoretical and Applied Fracture Mechanics, 37*(1-3), 49-98.

Horstemeyer, M. F., Lim, J., Lu, W. Y., Mosher, D. A., Baskes, M. I., Prantil, V. C., et al. (2002). Torsion/Simple Shear of Single Crystal Copper. *Journal of Engineering Materials and Technology, 124*(3), 322-328.

Jones, J. E. (1924). On the Determination of Molecular Fields. II. From the Equation of State of a Gas. *Proceedings of the Royal Society of London. Series A, 106*(738), 463-477.

Jones, J. E. (1924). On the Determination of Molecular Fields. I. From the Variation of the Viscosity of a Gas with Temperature. *Proceedings of the Royal Society of London. Series A, 106*(738), 441-462.

Kelchner, C. L., Plimpton, S. J., & Hamilton, J. C. (1998). Dislocation nucleation and defect structure during surface indentation. *Physical Review B, 58*(17), 11085.

Khan, A. S., & Huang, S. (1995). *Continuum theory of plasticity*. New York: Wiley.

Kocks, U. F. (1975). Constitutive relations for slip. In A. S. Argon (Ed.), *Constitutive Equations in Plasticity* (pp. 81-115). Cambridge, Mass.: MIT Press.

Kocks, U. F., TomÈ, C. N., & Wenk, H. R. (1998). *Texture and anisotropy : Preferred orientations in polycrystalsand their effect on materials properties*. Cambridge: Cambridge University Press.

Kratochvil, J. (1971). Finite-Strain Theory of Crystalline Elastic-Inelastic Materials. *Journal of Applied Physics, 42*(3), 1104-1108.

Kratochvil, J. (1973). On a finite strain theory of elastic-inelastic materials. *Acta Mechanica, 16*(1), 127-142.

Kröner, E. (1959). Allgemeine Kontinuumstheorie der Versetzungen und Eigenspannungen. *Archive for Rational Mechanics and Analysis, 4*(1), 273-334.

Kröner, E. (1981). *Continuum Theory of Defects.* In: Physique des defauts, New York, 215-315.

Lee, E. H. (1969). Elastic-plastic deformations at finite strains. *Journal of Applied Mechanics, ASME, 36*, 1-6.

Lippmann, H. (1995). Cosserat Plasticity and Plastic Spin. *Applied Mechanics Reviews, 48*(11), 753-762.

Liu, W. K., Karpov, E. G., Zhang, S., & Park, H. S. (2004). An introduction to computational nanomechanics and materials. *Computer Methods in Applied Mechanics and Engineering, 193*(17-20), 1529-1578.

Logan, J. D. (2006). *Applied Mathematics*. Hoboken, N.J.: John Wiley & Sons.

Mandel, J. (1971). Plasticité classique et viscoplasticité. In G. Maier, J. Salencon, W. Schneider, B. Schrefler & P. Serafini (Eds.), *Courses and Lectures held at Udine, Italy, September-October* (Vol. 97). New York: Springer.

Mandel, J. (1973). Constitutive equations and directors in plastic and viscoplastic media. *International Journal of Solids and Structures, 9*(6), 725-740.

Marin, E. B. (2006). *On the formulation of a crystal plasticity model*: Sandia National Laboratories.

Marin, E. B., Bammann, D. J., Regueiro, R. A., & Johnson, G. C. (2006). *On the formulation, parameter identification and numerical integration of the EMMI model: plasticity and isotropic damage*: Sandia National Laboratories.

Maxwell, J. C. (1870). On reciprocal figures, frames and diagrams of forces. *Transactions of the Royal Society Edinborough, 26*, 1-43.

McDowell, D. L. (2000). Modeling and experiments in plasticity. *International Journal of Solids and Structures, 37*(1-2), 293-309.

Melchionna, S., Ciccotti, G., & Holian, B. L. (1993). Hoover NPT dynamics for systems varying in shape and size. *Molecular Physics: An International Journal at the Interface Between Chemistry and Physics, 78*(3), 533 - 544.

Mendelev, M. I., Kramer, M. J., Becker, C. A., & Asta, M. (2008). Analysis of semi-empirical interatomic potentials appropriate for simulation of crystalline and liquid Al and Cu. *Philosophical Magazine, 88*, 1723-1750.

Mott, P. H., Argon, A. S., & Suter, U. W. (1992). The atomic strain tensor. *Journal of Computational Physics, 101*(1), 140-150.

Nemat-Nasser, S. (1990). Certain basic issues in finite-deformation continuum plasticity. *Meccanica, 25*(4), 223-229.

Ogden, R. W. (1984). *Non-linear elastic deformations*. Chichester; New York: E. Horwood ; Halsted Press.

Plimpton, S. (1995). Fast Parallel Algorithms for Short-Range Molecular Dynamics. *Journal of Computational Physics, 117*, 1-19.

Prantil, V. C., Jenkins, J. T., & Dawson, P. R. (1993). An analysis of texture and plastic spin for planar polycrystals. *Journal of the Mechanics and Physics of Solids, 41*(8), 1357-1382.

Querin, J. A., Schneider, J. A., & Horstemeyer, M. F. (2007). Analysis of micro void formation at grain boundary triple points in monotonically strained AA6022-T43 sheet metal. *Materials Science and Engineering: A, 463*(1-2), 101-106.

Schieck, B., & Stumpf, H. (1995). The appropriate corotational rate, exact formula for the plastic spin and constitutive model for finite elastoplasticity. *International Journal of Solids and Structures, 32*, 3643-3667.

Steinmann, P. (1996). Views on multiplicative elastoplasticity and the continuum theory of dislocations. *International Journal of Engineering Science, 34*, 1717-1735.

Stott, M. J., & Zaremba, E. (1980). Quasiatoms: An approach to atoms in nonuniform electronic systems. *Physical Review B, 22*(4), 1564.

Stumpf, H., & Badur, J. On missing links of rate-independent elasto-plasticity at finite strains. *Mechanics Research Communications, 17*(5), 353-364.

Swope, W. C., Andersen, H. C., Berens, P. H., & Wilson, K. R. (1982). A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters. *The Journal of Chemical Physics, 76*(1), 637-649.

Tadmor, E. B., Ortiz, M., & Phillips, R. (1996). Quasicontinuum analysis of defects in solids. *Philosophical Magazine A, 73*(6), 1529-1563.

Taylor, G. I. (1938). Plastic strain in metals. *Journal of the Institute of Metals, 62*(1), 307-324.

Teodosiu, C. (1989). The plastic spin: microstructural origin and computational significance. In D. R. J. Owen, E. Hinton & E. Onate (Eds.), *Computational plasticity: models, software and applications: proceedings of the Second International Conference held in Barcelona, Spain, 18th-22nd September, 1989* (Vol. 1, pp. 163). Swansea: Pineridge.

Tucker, G. J., Zimmerman, J. A., & McDowell, D. L. (2010). Shear deformation kinematics of bicrystalline grain boundaries in atomistic simulations. *Modelling and Simulation in Materials Science and Engineering, 18*(1), 015002.

Verlet, L. (1967). Computer "Experiments" on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules. *Physical Review, 159*(1), 98.

Yamakov, V., Wolf, D., Salazar, M., Phillpot, S. R., & Gleiter, H. (2001). Length-scale effects in the nucleation of extended dislocations in nanocrystalline Al by molecular-dynamics simulation. *Acta Materialia, 49*(14), 2713-2722.

Yamakov, V., Wolf, D., Phillpot, S. R., Mukherjee, A. K., & Gleiter, H. (2002). Dislocation processes in the deformation of nanocrystalline aluminium by molecular-dynamics simulation. *Nat Mater, 1*(1), 45-49.

Yamakov, V., Wolf, D., Phillpot, S. R., & Gleiter, H. (2003). Dislocation-dislocation and dislocation-twin reactions in nanocrystalline Al by molecular dynamics simulation. *Acta Materialia, 51*(14), 4135-4147.

Zhu, T., Li, J., J. Van Vliet, K., Ogata, S., Yip, S., & Suresh, S. (2004). Predictive modeling of nanoindentation-induced homogeneous dislocation nucleation in copper. *Journal of the Mechanics and Physics of Solids, 52*(3), 691-724.

Zimmerman, J. A., Kelchner, C. L., Klein, P. A., Hamilton, J. C., & Foiles, S. M. (2001). Surface Step Effects on Nanoindentation. *Physical Review Letters, 87*(16), 165507.

Zimmerman, J. A., Jones, R. E., Klein, P. A., Bammann, D. J., Webb III, E. B., & Hoyt, J. J. (2002). Continuum Definitions for Stress in Atomistic simulation. (SAND2002-8608), Size: 79 p.

Zimmerman, J. A., Bammann, D. J., & Gao, H. (2009). Deformation gradients for continuum mechanical analysis of atomistic simulations. *International Journal of Solids and Structures, 46*, 238-253.

APPENDIX A

SIMPLE SHEAR REPLICATION PYTHON SCRIPT

```
#####################################################################
#python 2.6
#
#  smpl_shr_sim.py
#
#  Created by Matthew Priddy on 2/12/09.
#
# This is an attempt to write python to perform a simple
# shear movement of a block of atoms (represented by points in
# python).  The purpose of this is to create an example that can be
# used by EAMpost to accurately calculate the deformation gradient.
#
# Note: To output numbers with decimal places, input numbers with
#         decimal places.  One strange thing about Python.
#
import linecache
import random
import math

f=atom_file = open('atom.initial', 'a')              # write initial data
f2=solution_file = open('dump.eampost', 'a')     # write final data

# input values
r =             2.5                 # radius of atom (lattice units)
x_natoms =   50                  # number of atoms in x-direction
y_natoms =   25                  # number of atoms in y-direction
z_natoms =   4                   # number of atoms in z-direction
v =             0.0002              # velocity of shear (distance/timestep)
timestep =    5                   # number of timesteps in simulation
t_inc =         10000               # the length of the timestep

# The outer box bounds of the simulation (unit length)
xlo = -(2.0 * r * x_natoms / 2.0)
xhi = (2.0 * r * x_natoms / 2.0)
ylo = -(2.0 * r * y_natoms / 2.0)
yhi = (2.0 * r * y_natoms / 2.0)
zlo = -(2.0 * r * z_natoms / 2.0)
zhi = (2.0 * r * z_natoms / 2.0)

# initial variables
d = 1                   # the atom number counter
i = 0.0                 # the count operator for the x-coordinate
j = 0.0                 # the count operator for the y-coordinate
k = 0.0                 # the count operator for the z-coordinate
xcoord = 0.0            # the x-coordinate calculated next
```

```
ycoord = 0.0          # the y-coordinate calculated next
zcoord = 0.0          # the z-coordinate calculated next

# write the number of atoms before creating them all.  The values are (# + 1)
because it starts at 0, not 1.  i.e. if a = 39, the number of atoms would be 40 in x-
direction because the interval is (0,39)

natoms = (x_natoms) * (y_natoms) * (z_natoms)

# determine coordinates for each atom and write it to atom.initial
while k <= z_natoms - 1:
        zcoord = (zlo + r) + (k * (2.0*r))

        j = 0
        while j <= y_natoms - 1:
                ycoord = (ylo + r) + (j * (2.0*r))

                i = 0
                while i <= x_natoms - 1:
                        xcoord = (xlo + r) + (i * (2.0*r))
                        f.write(str(d) + ", " + str(xcoord) + ", " + \
                                        str(ycoord) + ", " + str(zcoord) + "\n")
                        d = d + 1
                        i = i + 1
                j = j + 1
        k = k + 1
f.close()

# use atom.initial for the initial data so we can start moving.

t = 0                           # the time starts at t = 0
i = 1                           # variables used throughout
j = 0
k = 0

# LAMMPS dump output has 9 lines of various information
# before the atom data which is also written below.
while t <= timestep:

# these are the new values for the updated timesteps
        xlo = xlo + (v * t * t_inc)
        xhi = xhi + (v * t * t_inc)
        t_timestep = t * t_inc

        f2.write("ITEM: TIMESTEP\n" + str(t_timestep) + "\n" + \
```

```
        "ITEM: NUMBER OF ATOMS\n" + str(natoms) + "\n" + \
        "ITEM: BOX BOUNDS\n" + str(xlo) + " " + str(xhi) + "\n" + \
        str(ylo) + " " + str(yhi) + "\n" + str(zlo) + " " + \
        str(zhi) + "\n" + "ITEM: ATOMS FROM PYTHON PROGRAM\n")

        i = 1
        while i <= natoms:
                atom_data = linecache.getline('atom.initial', i,
module_globals=None)
                if atom_data == "":
                        break
                atom_data = eval(atom_data)
                atom_data = list(atom_data)
                atom_coord = atom_data
                atom_coord_num = atom_coord[0]
                atom_coord_x = atom_coord[1]
                atom_coord_x = float(atom_coord_x)
                atom_coord_y = atom_coord[2]
                atom_coord_y = float(atom_coord_y)
                atom_coord_z = atom_coord[3]
                atom_coord_z = float(atom_coord_z)

# We want to change the x-coordinate of the atom based on the
# y-coordinate of the atom.  In simple shear, as the y-coordinate
# increases, the atoms are moving faster in the positive
# x-direction.
# the velocity equation is v_atom = v * [(y - ylo) / (yhi - ylo)]
# simply multiply the velocity times time and add it to the x-coordinate
                ylo_r = ylo + r
                yhi_r = yhi - r
                v_atom = 0.0
                v_atom = v * ((atom_coord_y - ylo_r) / (yhi_r - ylo_r))
                atom_coord_x = (atom_coord_x + (v_atom * t * t_inc))

# Round the values to 4 decimal places (LAMMPS outputs 6 sig. figures,
# so this is the closest I can get to 6 significant figures.

                atom_coord_x = round(atom_coord_x, 4)
                atom_coord[1] = atom_coord_x
                f2.write(str(atom_coord[0]) + "  " + str(atom_coord[1]) + \
                str(atom_coord[2]) + "  " + str(atom_coord[3]) + "\n")
                i = i + 1
        t = t + 1
f2.close()
#####################################################################
```

APPENDIX B

LAMMPS SIMPLE SHEAR RELAXATION INPUT FILE

```
#######################################################################
# LAMMPS – version 15 January 2010
# 3d metal relaxation simulation  -  relax_shear.in
# Al Simple Shear Simulation, Relaxation process

units                   metal
boundary                s  s  p
atom_style              atomic
region          box block -120 120 -50 50 -30 30 units box
create_box              1 box
lattice         fcc 4.05 orient x 1 0 0 orient y 0 1 1 orient z 0 -1 1
create_atoms            1 box
mass                    1 26.9815
region          inner block -80 80 -40 40 -30 30 units box
group                   inner region inner
group                   outer subtract all inner
delete_atoms            group outer

# Uncomment this section for simulation with hole
#region                 hole cylinder z 0 0 3 INF INF
#group                  hole region hole
#group                  final subtract inner hole
#delete_atoms           group hole

pair_style              eam/fs
pair_coeff              * * Al_mm.eam.fs Al
lattice         fcc     1.0
log                     pyr_full.log
restart         10000 pyr_1.rst pyr_2.rst
neighbor                2.0 bin
neigh_modifydelay 10
fix                     1 all npt temp 0.01 0.01 0.5 z 0 0 25 drag 50
compute                 1 all centro/atom fcc
compute                 my_temp all temp
thermo                  10
thermo_style custom step temp pe pxx pyy pzz pxy xlo xhi ylo yhi zlo zhi
thermo_modify           lost warn norm yes
dump                    1 all custom 10000 dump.eampost id x y z c_1
timestep                0.005
run                     30000
#######################################################################
```

APPENDIX C

LAMMPS SIMPLE SHEAR INPUT FILE

```
#########################################################################
# LAMMPS – version 15 January 2010
# 3d metal simple shear simulation  -  uni_shear.in
# Al Simple Shear Simulation, shear process

units                   metal
boundary                s  s  p
atom_style              atomic
read_restart  ../relax_hole_3/pyr_1.rst
mass                    1 26.9815
pair_style              eam/fs
pair_coeff              * * Al_mm.eam.fs Al
lattice         fcc 1.0
log                     pyr_full.log
restart         10000 pyr_1.rst pyr_2.rst
neighbor                2.0 bin
neigh_modifydelay 10
region          top block INF INF 30 INF INF INF units box
group                   top region top
region          bottom block INF INF INF -30 INF INF units box
group                   bottom region bottom
group                   boundary union bottom top
group                   free subtract all boundary
group                   most subtract all top
fix      1 free npt temp 0.01 0.01 0.5 z 0 0 25 couple none drag 50
fix                     4 bottom move linear 0.00 0.00 0.00 units box
fix                     5 top move linear 0.01 0.00 0.00 units box
compute                 1 all centro/atom fcc
compute                 2 all stress/atom
compute                 my_temp all temp
thermo                  10
thermo_modify       lost warn norm yes
thermo_style custom step temp pe pxx pyy pzz pxy xlo xhi ylo yhi zlo zhi
dump                    1 all  custom 10000 dump.eampost  id type x y z c_1 c_2[1]
c_2[2] c_2[3] c_2[4] c_2[5] c_2[6]
dump                    2 free  custom 10000 dump.free  id type x y z c_1 c_2[1]
c_2[2] c_2[3] c_2[4] c_2[5] c_2[6]
timestep                0.005
run                     320000
#########################################################################
```

APPENDIX D

PREWARP INPUT FILE

```
#####################################################################
# PreWarp Input File for Al_15_30 aka Al (110) high-angle grain boundary
# simulation.

# set output filename root
filename ./Al_H_00.run3

# set size of atomic system
box -300  300  -300  300  -11.456  11.456  1.0

# set periodicity
periodicity 0 1 0
overlap 1.1
#expand 1.07

# set number of grains
grain AlH                                       # Grain 3 (top left)
site -8.0 10.0 0.0 1.0
orient x -.707 .707 0 y 0 0 1

grain AlH                                       # Grain 2 (right side)
site 12.5 -1.0 0.0 1.0
orient x .966 -.966 .259 y -.259 .259 .966

grain AlH                                       # Grain 1 (bottom left)
site -7.0 -9.75 0.0 1.0
orient x .866 -.866 .5 y -.5 .5 .866

# define lattice
lattice user AlH
  0.00000000  2.02500000  2.02500000
  2.02500000  0.00000000  2.02500000
  2.02500000  2.02500000  0.00000000
2                                               # number of atoms
Al 0.0 0.0 0.0   1.00
H  0.5 0.5 0.5   0.00

build

# set output files
write data
#write ensight
#write r3d
write warp
#####################################################################
```

APPENDIX E

PREWARP ORIENTATION ANGLES

Table E.1 Orientation angles for (110) texture for x- and y-components of the grains.

θ	x-components			y-components		
0	0.707	-0.707	0.000	0.000	0.000	1.000
5	0.704	-0.704	0.087	-0.062	0.062	0.996
10	0.696	-0.696	0.174	-0.123	0.123	0.985
15	0.683	-0.683	0.259	-0.183	0.183	0.966
30	0.612	-0.612	0.500	-0.354	0.354	0.866
60	0.354	-0.354	0.866	-0.612	0.612	0.500

Table E.2 Orientation angles for (100) texture for x- and y-components of the grains.

θ	x-components			y-components		
0	1.000	0.000	0.000	0.000	1.000	0.000
5	0.996	0.087	0.000	-0.087	0.996	0.000
10	0.985	0.174	0.000	-0.174	0.985	0.000
15	0.966	0.259	0.000	-0.259	0.966	0.000
30	0.866	0.500	0.000	-0.500	0.866	0.000
60	0.500	0.866	0.000	-0.866	0.500	0.000

APPENDIX F

LAMMPS AL TJ RELAXATION INPUT FILE

```
#######################################################################
# LAMMPS – version 9 January 2009
# 3d metal relaxation simulation  -  relax_iter.in
# Al Triple Junction, iteration relaxation

units                   metal
boundary                p  p  p
atom_style              atomic
read_data               atom_in_00.atoms
mass                    1 26.9815
pair_style              eam/fs
pair_coeff              * * Al_mm.eam.fs Al
lattice         fcc 1.0
log                     pyr_full.log
restart         1000 pyr_1.rst pyr_2.rst
neighbor                2.0 bin
neigh_modifydelay 10
region          left block INF -185.0 INF INF INF INF units box
group                   left region left
group                   swap1 subtract all left
region          right block 185.0 INF INF INF INF INF units box
group                   right region right
group                   swap2 subtract swap1 right
region          top block INF INF 185.0 INF INF INF units box
group                   top region top
group                   swap3 subtract swap2 top
region          bottom block INF INF INF -185.0 INF INF units box
group                   bottom region bottom
group                   free subtract swap3 bottom
velocity                all create 1.0 1231
fix       1 free npt 1.0 300.0 0.5 aniso NULL NULL 0 0 0 0 25 drag 50
fix                     2 left planeforce 1.0 0.0 0.0
fix                     3 right planeforce 1.0 0.0 0.0
fix                     4 top planeforce 1.0 0.0 0.0
fix                     5 bottom planeforce 1.0 0.0 0.0
compute                 1 all centro/atom
compute                 my_temp all temp
thermo                  10
thermo_modify           lost warn norm yes
thermo_style custom step temp pe pxx pyy pzz press xlo xhi ylo yhi zlo zhi
dump                    1 all custom 50000 dump.relax tag type x y z c_1
timestep                0.005
run                     50000

# now, all the pressure to minimize in the y-direction
```

```
fix       1 free npt 300.0 300.0 0.5 aniso 0 0 NULL NULL 0 0 25 drag 50
fix                 2 left planeforce 0.0 1.0 0.0
fix                 3 right planeforce 0.0 1.0 0.0
fix                 4 top planeforce 0.0 1.0 0.0
fix                 5 bottom planeforce 0.0 1.0 0.0
thermo              10
thermo_modify       lost warn norm yes
thermo_style        custom step temp pe pxx pyy pzz press xlo xhi ylo yhi zlo zhi
dump                2 all custom 10000 dump.relax2 tag type x y z c_1
run                 10000
# now, all the pressure to minimize in the x-direction
fix                 1 free npt 300.0 300.0 0.5 aniso NULL NULL 0 0 0 0 25 drag
50
fix                 2 left planeforce 1.0 0.0 0.0
fix                 3 right planeforce 1.0 0.0 0.0
fix                 4 top planeforce 1.0 0.0 0.0
fix                 5 bottom planeforce 1.0 0.0 0.0
thermo              10
thermo_modify       lost warn norm yes
thermo_style        custom step temp pe pxx pyy pzz press xlo xhi ylo yhi zlo zhi
dump                3 all custom 10000 dump.relax3 tag type x y z c_1
run                 10000
# now, all the pressure to minimize in the y-direction
fix       1 free npt 300.0 300.0 0.5 aniso 0 0 NULL NULL 0 0 25 drag 50
fix                 2 left planeforce 0.0 1.0 0.0
fix                 3 right planeforce 0.0 1.0 0.0
fix                 4 top planeforce 0.0 1.0 0.0
fix                 5 bottom planeforce 0.0 1.0 0.0
thermo              10
thermo_modify       lost warn norm yes
thermo_style        custom step temp pe pxx pyy pzz press xlo xhi ylo yhi zlo zhi
dump                4 all custom 10000 dump.relax4 tag type x y z c_1
run                 10000
# now, all the pressure to minimize in the x-direction
fix       1 free npt 300.0 300.0 0.5 aniso NULL NULL 0 0 0 0 25 drag 50
fix                 2 left planeforce 1.0 0.0 0.0
fix                 3 right planeforce 1.0 0.0 0.0
fix                 4 top planeforce 1.0 0.0 0.0
fix                 5 bottom planeforce 1.0 0.0 0.0
thermo              10
thermo_modify       lost warn norm yes
thermo_style        custom step temp pe pxx pyy pzz press xlo xhi ylo yhi zlo zhi
dump                5 all custom 10000 dump.relax5 tag type x y z c_1
run                 10000
#####################################################################
```

APPENDIX G

LAMMPS AL TJ UNIAXIAL TENSION INPUT FILE

```
#####################################################################
# LAMMPS – version 9 January 2009
# 3d metal uniaxial tension test  -  uni_iter.in
# Al Triple Junction, uniaxial tension

units                   metal
boundary                p  p  p
atom_style              atomic
read_restart  ../pyr_2high_110.rst
reset_timestep          0
mass                    1 26.9815
pair_style              eam/fs
pair_coeff              * * ./Al_mm.eam.fs Al
lattice         fcc 1.0
log                     pyr_full.log
restart 10000 pyr_h100_*.rst
neighbor                2.0 bin
neigh_modify delay 10
fix       1 free npt 300.0 300.0 0.5 aniso NULL NULL 0 0 0 0 25 drag 50
fix                     2 left planeforce 1.0 0.0 0.0
fix                     3 right planeforce 1.0 0.0 0.0
fix                     4 top planeforce 1.0 0.0 0.0
fix                     5 bottom planeforce 1.0 0.0 0.0
fix                     6 all deform 1 x delta -20 20 units box remap x
compute                 1 all centro/atom
compute                 2 all stress/atom
compute                 my_temp all temp
thermo                  10
thermo_modify           lost warn norm yes
thermo_style custom step temp pe pxx pyy pzz press xlo xhi ylo yhi zlo zhi
dump                    1 all  custom 10000 dump.relax tag type x y z c_1 c_2[1]
c_2[2] c_2[3] c_2[4] c_2[5] c_2[6]
dump                    2 all  custom 10000 dump.eampost tag x y z
timestep                0.005
run                     200000
#####################################################################
```

APPENDIX H

CRYSTAL PLASTICITY SIMULATION ABAQUS INPUT FILE

```
*Heading
 Aluminum (100) High-Angle TJ - 3D Model Uniaxial Deformation (Abaqus 6.9-1)
** in mm, N, tonne(10^3 kg), s, MPa (N/mm^2), mJ, tonne/mm^3
** Job name: 3D_uniaxial_100_high       Model name: Al_100_high
**********************************************************************************************
Node,    INPUT=tj.nd
*Include, INPUT=node_sets.es
*Element, INPUT=tj.el type=C3D8R
*Include, INPUT=el_sets.es
*Solid Section, elset=grain1, controls=EC-1, material="Grain 1",
*Solid Section, elset=grain2, controls=EC-1, material="Grain 2",
*Solid Section, elset=grain3, controls=EC-1, material="Grain 3",
*Section Controls, name=EC-1, hourglass=ENHANCED       1., 1., 1.
*Amplitude, name=Amp-1            0.,          0.,        150.,          1.
*Material, name="Grain 1"
*Depvar        70,
*User Material, constants=5
10080.,   1., -90.,   0.,-180.
*Material, name="Grain 2"
*Depvar        70,
*User Material, constants=5
12880.,   1., -90.,   0.,-150.
*Material, name="Grain 3"
*Depvar        70,
*User Material, constants=5
10080.,   1., -90.,   0.,-120.
**********************************************************************************************
Step, name=Step-1, nlgeom=YES, inc=500
uniaxial tension, rate ~ 2 mm/sec
*Static          0.01, 150., 2e-06, 1.5
*Boundary
Bottom, 2, 2
*Boundary, amplitude=Amp-1
Left, 1, 1, -30.
*Boundary, amplitude=Amp-1
Right, 1, 1, 30.
*Output, field
*Node Output
CF, RF, U
*Element Output, directions=YES
LE, S, SDV
*Contact Output
CDISP, CSTRESS
*End Step
**********************************************************************************************
```