2018-04-01

Stress Modulated Grain Boundary Mobility

Derek Michael Lontine
Brigham Young University

Stress Modulated Grain Boundary Mobility

Derek Michael Lontine

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Oliver K. Johnson, Chair
Eric R. Homer
David T. Fullwood

Department of Mechanical Engineering

Brigham Young University

ABSTRACT

Stress Modulated Grain Boundary Mobility

Derek Michael Lontine
Department of Mechanical Engineering, BYU
Master of Science

This thesis consists of a thermodynamically based kinetic model that more accurately predicts grain boundary mobility (GBM) over a large range of thermodynamic states including changes in temperature, pressure and shear stress. The form of the model was validated against calculated GBM values for Al bicrystals via molecular dynamics (MD) simulations. A total of 98,786 simulations were performed (164 different GBs, each with a minimum of 250 different thermodynamic states, and 2 different driving forces). Methodology for the computation of the GBM via MD simulations is provided. The model parameters are directly linked to extensive thermodynamic quantities and suggest potential mechanisms for GBM under combined thermal and triaxial loads. This thesis also discusses the influence of GB character on the thermodynamic mobility parameters. The resulting insights about GB character and thermodynamic state on GBM suggest an opportunity to achieve designed microstructures by controlling thermodynamic state during microstructure evolution.

Keywords: material science, grain boundary engineering, grain boundary mobility, molecular dynamics, high pressure, ulta-high pressure, grain growth, shear coupling

ACKNOWLEDGMENTS

There are too many people to acknowledge in my growth as an individual. I sincerely feel grateful for the support that has been offered to me from every direction. When I began pursuing a graduate degree I felt overwhelmed (and still do to some extent). I've been helped and encouraged at every turn by my neighbors, friends, professors, colleagues & family.

Particular thanks is owed to my advisor and now friend, Dr. Oliver Johnson. You have been consistently supportive and trusting of me. You've pushed me to stretch myself, and pushed back far too many lunches to talk with me.

I owe a great deal of thanks to US Synthetic Corp for supporting me and my desires to pursue higher education. The company and all of my friends who work there are so supportive of me and my desire to grow. Thank you, Ken Bertagnolli and Kevin Graham, you instilled confidence in my capability and despite your better judgment went along with the experiment.

I would like to thank my children, Anna and Paul. Thank you for demanding that I play with you when I thought that I "needed to work" on this research and other coursework. Looking back it is hard to imagine that Anna was just 2, and Paul was just born the first semester I started this journey. Now, I have a 5 year old and a 3 year old. I've missed far too many of the precious moments. You two have provided me with joy beyond measure. I hope that this thesis will stand as a permanent example to you of your potential. I want the absolute best for you. I love you.

Most of all, I would like to acknowledge my wife, Sarah. Words just don't say enough. Thank you. Thank you for your patience. Thank you for tolerating my frustrations, my nerd-out moments, and my stress-out moments. Thank you for understanding my desire to do this. Thank you for calming me when I've been stressed. Thank you for sticking with me. Thank you for absolutely everything. I don't think we knew going back to school would be this difficult, but we made it. And we're done. I look forward to giving you back your husband.

TABLE OF CONTENTS

LIST OF FIGURES . . . . . **vi**

LIST OF TABLES . . . . . **vii**

NOMENCLATURE . . . . . **viii**

Chapter 1 Introduction **1**
1.1 Background 2
1.2 Molecular Dynamics 3
1.2.1 Interatomic Potentials 4
1.2.2 Thermodynamic Ensembles & Connections to Statistical Mechanics 4
1.2.3 LAMMPS 6
1.3 Bicrystal Simulation Cell 7

Chapter 2 Methods for MD Mobility Calculations **9**
2.1 Thermal & Mechanical Equilibration 9
2.2 Mobility Simulation 11
2.3 Errors in MD Driving Force Method 12
2.4 Determining the position of the GB 13
2.5 Calculation of GB Velocity 14
2.6 Calculation of GB Mobility 15
2.7 Design of Experiments 17

Chapter 3 Theory **19**
3.1 Thermodynamics: External Stress Work Contributions 19
3.2 Thermodynamics: Free Energy 20
3.3 Arrhenius Behavior: Kinetics 21

Chapter 4 Model Performance **24**
4.1 Suitability of Kinetic Model 24
4.2 Influence of Known Factors of Error 29
4.3 Changes in Kinetic Model Performance by Suppression of Intensive/Extensive Conjugate Pairs 30
4.4 Evaluation of the Kinetic Model Using Validation Data 32
4.5 Convergence of Model Parameters 33

Chapter 5 Model Coefficients **36**
5.1 Trends in GBM Kinetics - Pre-exponential Factor M_0 36
5.2 Trends in GBM Kinetics - Activation Energy U^* 38
5.3 Trends in GBM Kinetics - Activation Volume V^* 41
5.4 Trends in GBM Kinetics - Activation Volume Deviator Tensor Λ^*_{ij} 44

Chapter 6 Conclusion **50**
6.1 Advantages & Disadvantages of Predictive Model 50
6.2 Summary 51
6.3 Future Work 52
6.4 Major Findings 52
6.5 Funding 53

REFERENCES **54**

Appendix A Corrigendum to Mobility Due to Synthetic Driving Force Errors **59**

Appendix B Corrigendum to Mobility Due to GB Velocity Calculations **63**
B.1 MATLAB Script: analyzeGBvelocityRef.m 68

Appendix C HPC Optimization & Workflow **84**
C.1 Dataset 84
C.2 Usage Statistics 85

Appendix D Example LAMMPS Input Files & Corresponding Shell Scripts **86**
D.1 LAMMPS Input Scripts 86
D.1.1 Thermalizing Calculations 86
D.1.2 Mobility Calculations 89
D.2 Shell Scripts 91
D.2.1 Thermalizing Job 91
D.2.2 Mobility Job 92

Appendix E GB position tracking algorithm **93**
E.1 OVITO Scripts 93
E.2 MATLAB Script: analyzeMobility.m 96

Appendix F Validation Model Tools **108**
F.1 Primary Validation Script (Including Plots) 108
F.2 Model Builder Script 116

LIST OF FIGURES

1.1 Mishin EAM potentials using first-principles calculations 5
1.2 Illustration of the minimum energy configuration for STGB ⟨100⟩ 8

2.1 Illustration of grain boundary motion in MD simulations 11
2.2 Driving Force Error in STGB ⟨110⟩ GBs 13
2.3 Illustration of CKSD GB finder 15
2.4 Sample output of GB tracking algorithm 16
2.5 Illustration of grain boundary stress state 18

4.1 Goodness of fit of all data 25
4.2 Characteristic fits using thermodynamic potential model in randomly selected GBCs . 26
4.3 Appropriateness of thermodynamic model in ATGB and STGB systems 27
4.4 Pearson's correlation coefficient in multiple GBCs 28
4.5 Model validation data fitting 33
4.6 The effect of sample size on parameter calculations 35

5.1 Model parameter M_0 versues the GB tilt or inclination angle for the respective ATGB or STGB noted. 37
5.2 Activation energy versues the GB tilt or inclination angle for the respective ATGB or STGB noted. 39
5.3 Correlation between M_0 and U^* 40
5.4 Activation volume versues the GB tilt or inclination angle for the respective ATGB or STGB noted. 42
5.5 Correlation between V^* and U^* 43
5.6 Activation volume deviator tensor components Λ^*_{12} versues the GB tilt or inclination angle for the respective ATGB or STGB noted. 46
5.7 Activation volume deviator tensor components Λ^*_{23} versues the GB tilt or inclination angle for the respective ATGB or STGB noted. 47
5.8 Comparison of published shear coupling factors to Λ_{12} in STGB ⟨100⟩ 49

A.1 Difference in Driving Force for Run05013 60
A.2 Difference in Driving Force for Run06526 61
A.3 Difference in Driving Force for Run14311 61
A.4 Difference in Driving Force for Run08099 62

B.1 Difference in GB Velocity for Run06526 65
B.2 Difference in GB Velocity for Run44350 66
B.3 Difference in GB Velocity for Run14311 66
B.4 Difference in GB Velocity for Run16001 67
B.5 Difference in GB Velocity for Run16501 67

C.1 Parallel efficiency of random walk LAMMPS 85

LIST OF TABLES

1.1 Thermodynamic ensembles 6

4.1 Effect of reduced model parameters on kinetic model quality 31

A.1 Tabulated errors of differing GBs under differing thermoelastic states 62

B.1 Details regarding analysis of dual GB velocity data 64

NOMENCLATURE

A	Helmholtz free energy
δ_{ij}	Kronecker delta
ε_{vol}	Volumetric strain
$\tilde{\varepsilon}_{ij}$	Strain deviator tensor
f	Driving force
G	Gibbs free energy
G^*	Activation free energy
G_a	Free energy of activated state
G_0	Free energy of ground state
k_b	Boltzmann constant
M	Grain boundary mobility
$\hat{n}_i$	Plane normal
p	Pressure
R^2	Pearson's correlation coefficient
S	Entropy
S^*	Activation entropy
σ_{ij}	Cauchy stress tensor
$\tilde{\sigma}_{ij}$	Cauchy stress deviator tensor
T	Temperature
t_i	Traction force
U	Internal energy
U^*	Activation (internal) energy
v	Grain boundary velocity
V'	Instantaneous volume
V	Volume
V^*	Activation isostatic volume
Λ_{ij}	Volume deviator tensor
Λ^*_{ij}	Activation volume deviator tensor

Subscripts, superscripts, and other indicators

$[\,]_i$	indicates [ ] is a vector
$[\,]_{ij}$	indicates [ ] is a tensor
$[\,]_a$	indicates [ ] is an activated state property
$[\,]_0$	indicates [ ] is a ground state property
$[\,]_{iso}$	indicates [ ] is an isostatic scalar property
$\tilde{[\,]}$	indicates [ ] is a deviatoric property
$\hat{[\,]}_i$	indicates [ ] is a unit vector
$[\,]'$	indicates [ ] is an instantaneous (spatial reference frame) property

CHAPTER 1. INTRODUCTION

Grain boundaries (GBs) strongly influence many material properties including mechanical strength [1,2], diffusivity and work-hardening [3]. The impact of GBs is amplified with decreasing grain size [4–6] as GBs represent larger volume fractions of the material.

If the character and connectivity of GBs in polycrystalline materials can be controlled, significant improvements in material properties can be achieved [7–11]. Traditional grain boundary engineering (GBE) methods manipulate the GB character distribution (GBCD) and connectivity through cyclic thermomechanical processing [7]. Separate studies have demonstrated that motion of GBs can be enhanced [12–26] or suppressed [27–31] with the addition of various forms of mechanical stress. The effect of temperature, stress or combinations of the two is likely to vary with crystallographic character of the GB [12–31].

These discoveries suggest an opportunity to employ temperature and stress to selectively accelerate or suppress the motion of different subpopulations of GBs to reach target microstructures. Intelligent selection of processing routes requires a model to predict the influence of thermoelastic state on GB mobility. Such a model would (1) enable process design, (2) enhance current microstructure evolution models/simulation techniques [32–36] to include the influence of stress, and (3) expand our understanding of the mechanisms of GB motion generally.

The present work proposes a simple thermodynamically inspired model and validates it against GB mobility calculations obtained from Molecular Dynamics (MD) simulations. The model presented provides connections to extensive thermodynamic quantities that may additionally reveal mechanisms for GB migration. Observations suggest that GB expansion and deformation may be determining factors in the mobility of GBs. The proposed model assists in understanding GB motion phenomena observed with changes in stress state [15, 16, 27, 30, 31, 37].

1.1 Background

The velocity (v) of GBs is generally observed to be proportional to the driving pressure (P) for motion [38]:

$$v = MP \tag{1.1}$$

where the proportionality constant, M, is the mobility of the GB, and is a key factor in controlling the kinetic process of GB motion.

It has been observed both experimentally [39] and through simulation [39, 40] that the kinetics of GBM generally follow a thermally-activated Arrhenius relationship with temperature. Some MD simulations of certain GBs demonstrate a "thermally damped" behavior where GBM decreases as temperature increases, or "athermal" behavior in which the mobility is apparently unaffected by temperature [41].

The role of mechanical stresses have also been separately interrogated. Experiments [12–31, 37, 42–44] and simulations [45–50] demonstrate that GBM typically increases with shear stress and is suppressed by hydrostatic compressive stress (pressure). Although these factors (temperature, pressure, shear stress) have been separately investigated, studies interrogating the combination of these states have not previously been conducted.

The driving pressure is traditionally regarded as the product of the interface energy γ and the GB curvature κ. Thus the GB velocity at a given point along an interface having GB curvature can be written as

$$v = M\gamma\kappa \tag{1.2}$$

These equations perform well when describing the behavior in cases of isotropic grain growth. Many models exist that simulate GB motion in these isotropic states [32–36].

The advancement of these models to include GBM anisotropy or "anisotropic grain growth" is far more complicated. Anisotropic grain growth models include the effect of grain boundary character (GBC) into the parameters that affect mobility[1]. In these cases, the GB mobility M and

[1]The GBC is a generic term for the description of the crystallography of the GB using the 5-dimensional factor space that defines a GB.

GB energy γ are functions of the GB character, $\psi = [\omega, \theta, \phi, \alpha, \beta]$. Thus Eq 1.2 becomes

$$v = M(\psi)\gamma(\psi)\kappa \tag{1.3}$$

This velocity equation still lacks critical information regarding the influence of imposed boundary conditions. It does not consider changes in mobility due to changes in the thermoelastic state (temperature, pressure and shear stress). Thus a model that predicts changes in mobility under applied thermoelastic states (T, σ_{ij}) is desired to inform grain growth models to include these factors. Such a model would be included in a curvature driven GB evolution model as

$$v = M(\psi, T, \sigma_{ij})\gamma(\psi T, \sigma_{ij})\kappa \tag{1.4}$$

Here we see that the GB energy (γ) may also be influenced by thermoelastic states. This may occur, however, in this study the driving pressure for GB motion was controlled independent of GB curvature, so this becomes of little consequence in this particular study. However, continuum-level grain growth models must also include GB energy.

Microstructural evolution models that consider a wide array of thermoelastic states would require a velocity equation as represented in Eq. 1.4. However, a generalized model predicting the mobility as a function of GBC and thermoelastic states is needed to create a microstructure evolution model. This thesis seeks to provide such a model for GB mobility.

1.2 Molecular Dynamics

This research employs a modern computational technique called molecular dynamics (MD) to study the effect of stress and temperature on grain boundary mobility. Molecular dynamics is an atomistic materials modeling approach where individual atoms are simulated using an interatomic potential. The Newtonian physics for each atom are calculated over many time steps. From these calculations atom positions, velocities, and forces are all calculated and the simulated atoms are permitted to evolve. Molecular dynamics includes both deterministic and stochastic numerical methods and both are applied in this research.

In the present work, we investigate trends in GBM under a large number of combined thermal and triaxial (combined pressure and shear stress) stress states. The resulting 98,786 MD simulations were performed using LAMMPS (Large-scale Atomic/Molecular Massively Parallel Simulator) [51]. Total computation time to complete these simulations was 316,621,308 CPU-minutes (approx. 600 CPU-years).

The intent of the present work was not to deeply interrogate atomic motion mechanisms in each simulation, but rather, to observe trends in the mobility with changes in the applied conditions of the system for GBs of different character. This catalog of mobility simulations is among the largest in scope of those conducted to date studying GB mobility.

1.2.1 Interatomic Potentials

Interatomic potentials are the framework for calculating the forces on atoms in MD simulations. An interatomic potential enables MD software to determine the atom's potential energy given the position relative to neighbors of the atom. Interatomic potentials may be developed by experimental methods, by utilizing quantum mechanics (first principles) modeling, or by any combination of the two. Fundamentally, interatomic potentials define the forces acting on atoms due to their proximity to one another.

A class of interatomic potentials is the embedded atom method (EAM) potential [53]. One such EAM potential was developed by Mishin et al [52] for aluminum. This a potential is schematically represented in Fig 1.1. MD simulations performed in this research used the Aluminum EAM interatomic potential from Mishin et al in all simulations.

1.2.2 Thermodynamic Ensembles & Connections to Statistical Mechanics

The thermodynamic ensemble of MD simulations determines the primary boundary conditions imposed upon the system. Thermodynamic ensembles control intensive or extensive properties of the simulation cell. However, intensive and extensive variables that are conjugate to one another cannot be controlled at the same time. These ensembles have a basis in statistical mechanics and thermodynamics. There are many forms of thermodynamic ensembles and these are detailed in table 1.1.

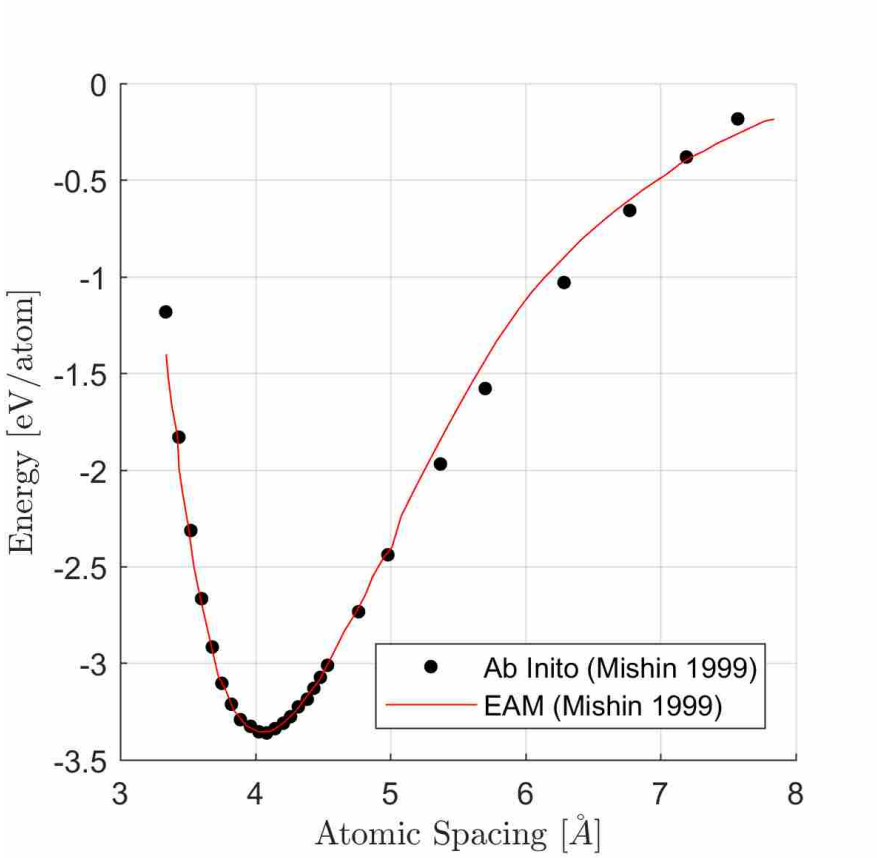


Figure 1.1: Interatomic potential (EAM) utilized in the present work compared to datapoints obtained from quantum mechanics materials modeling (Ab Inito). All data obtained from Mishin et. al. [52] and is for Aluminum.

Each thermodynamic ensemble has a partition function (Ψ) that corresponds to that ensemble. A partition function is a statistical description of the micro-state. Each partition function is related to a characteristic state function (Q) via

$$\ln(\Psi) = \alpha Q \tag{1.5}$$

where α is a coefficient that may be positive or negative (usually the Boltzmann constant, k_b). This relationship corresponds to the Arrhenius relationship that relates many kinetic processes to the thermodynamic driving forces for transformation.

Many MD simulations (including ones that study GB mobility) are performed in the canonical NVT ensemble (where number of atoms, volume and temperature are controlled). The

Table 1.1: Detail of thermodynamic ensembles available in most MD simulation packages. Boxes indicate that the ensemble uses the intensive or extensive parameter checked. N is number of atoms held constant, μ is the chemical potential, V is volume, P is pressure, E is energy, T is temperature, H is enthalpy. Conjugate pairs (N & μ), (V & P) are grouped together. Also, the intensive/extensive parameters (E, T & H) are also grouped to demonstrate that control of these factors typically does not coexist.

Ensemble	Name	N	μ	V	P	E	T	H	Characteristic State Function
Microcanonical	NVE	✓		✓		✓			TS
Canonical	NVT	✓		✓			✓		A - Helmholtz Free Energy
Grand canonical	μVT		✓	✓			✓		Φ - Grand potential
Isoenthalpic-isobaric	NPH	✓			✓			✓	
Isothermal-isobaric	NPT	✓			✓		✓		G - Gibbs Free Energy

associated characteristic state function to the canonical (NVT) ensemble is the Helmholtz free energy(A). In a similar manner, those MD simulations performed in the isothermal-isobaric (NPT) ensemble have an associated partition function being the Gibbs free energy(G). It is appropriate to analyze processes by using the characteristic state function that corresponds to the thermodynamic ensemble.

By utilizing this information, we see that if the effect of mechanical stresses (P) and temperature (T) are to be modulated, one must use the NPT thermodynamic ensemble. Additionally, kinetic processes that result from MD simulations performed in the NPT ensemble are characterized by the Gibbs free energy.

1.2.3 LAMMPS

LAMMPS (Large-scale Atomic/Molecular Massively Parallel Simulator) was the molecular dynamics software package used in this research [51]. The software is developed and maintained by Sandia National Laboratories and is widely used and accepted in the materials science community. LAMMPS is licensed under the GNU public license making this software free for all who desire to employ it.

LAMMPS is predominately written in C++ and employs many high-performance computing (HPC) techniques to accelerate computational work and take advantage of multi-core processing. LAMMPS is also capable of running graphical processing unit (GPU) hardware to further accelerate computations. No new features or custom code was added to the generic

LAMMPS package (version Aug 2017) that is publicly available. However, the MISC and MANYBODY packages within LAMMPS must be included when compiling LAMMPS for the methods used in this research to function. The Fulton Supercomputing Lab (FSL) has many builds of LAMMPS that meet this criterion and these builds were employed in this research.

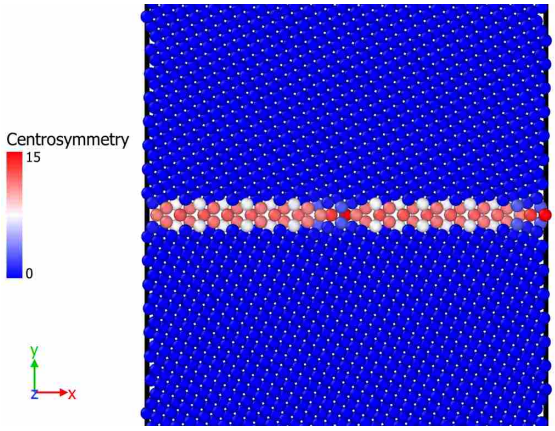
1.3 Bicrystal Simulation Cell

The present work studied 164 aluminum bicrystals having unique grain boundary character (GBC). Simulation cells contained between 28,060 and 113,230 atoms and were periodic in all three dimensions. As a consequence of the periodicity, each configuration contained 2 identical GBs. The minimum energy configurations for each GBC were obtained from the work of Tschopp et. al. [54]. The embedded-atom method (EAM) potential used both in the work of Tschopp et. al. and the present work was obtained from Mishin et. al. [52]. In all cases, the simulation cell was oriented such that the GB normals were parallel to the y-axis of the laboratory coordinate system. Depending on the GBC, the tilt or inclination axis, respectively, for each GB was oriented parallel to the z-axis.

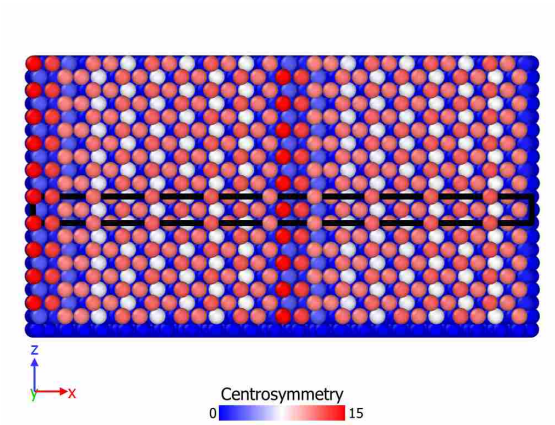
An example of atomic positions of the minimum energy configuration for one GBC are shown in figure 1.2. The figure shows a boxed region where the initial Tschopp simulation cell was defined.

The simulation cells as provided by Tschopp et. al. [54] are far too small for accurate simulation of GBM. Most simulation cells were under 2000 atoms. The simulation cells were replicated and expanded to increase the simulation cell size. Larger simulation cells are preferred; however, computation time required for simulations scales with the number of atoms simulated (see Appendix C). This research targeted a simulation cell that was at minimum 10 unit cells in both the X and Z dimensions. Many simulation cells exceeded this minimum size.

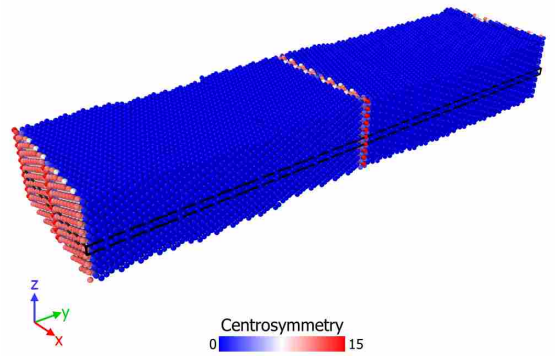
This dataset contains, 68 symmetric tilt grain boundaries (STGBs) and 96 asymmetric tilt grain boundaries (ATGBs). Of the STGBs there are 29 $\langle 100 \rangle$ STGBs, 32 $\langle 110 \rangle$ STGBs, and 7 $\langle 111 \rangle$ STGBs. Many of the GBs in this dataset are characterized by low coincident site lattice (CSL) numbers including 26 Σ3's, 16 Σ5's , 27 Σ9's, 27 Σ11's, and 10 Σ13a's.



(a)



(b)



(c)

Figure 1.2: Schematic representation of the minimum energy configuration of STGB $\langle 100 \rangle$ as visualized in Ovito [55] (a)showing the XY plane with tilt axis oriented in the Z axis (b) showing the XZ plane through the center of the GB (c) showing an isometric view of the entire simulation cell. Color indicates the value of the centrosymmetry parameter.

CHAPTER 2. METHODS FOR MD MOBILITY CALCULATIONS

MD was employed to study the effect of thermoelastic states on GBM. By doing so, a wide rage of GBCs, stress states and temperatures may be simulated rapidly. Experimental methods to study GBM could be used, however, these methods require (a) synthesis of GB bicrystals that vary in GBC (b) a capability to reach the thermoelastic states and (c) a significant amount of time to run each sample.

Each MD simulation seeks to obtain the GBM through a procedure that is detailed below. This process was performed for each thermoelastic state and GBC. The method detailed here is generalized and may also be employed to study GBM without application of mechanical loads. It also simulates laboratory thermodynamic states where experimental studies on GBM have been performed.

2.1 Thermal & Mechanical Equilibration

The thermodynamic ensemble used in this work was NPT (in which the number of molecules, pressure and temperature are controlled). This ensemble was used so that we might modulate the parameters of stress and temperature as in the laboratory environment. The NPT ensemble permits not just hydrostatic pressure, but changes to all normal stresses (σ_{11}, σ_{22}, σ_{33}) independently. Additionally, all simulations in the present work were performed under triclinic boundary conditions. Triclinic simulation cells permit shear stress to be applied to the simulation cell, thus expanding the available stresses to a full (6-element) stress tensor. Under triclinic boundary conditions, the simulation cell is permitted to form a parallelepiped that can be deformed to various extents under the action of both normal and shear stresses. The triclinic constraint further opens the degrees of freedom and conforms to the laboratory thermodynamic state.

The NPT thermodynamic ensemble was previously studied by Janssens et al [56] in their pioneering work using MD to study GB mobility. In their work, the NPT ensemble produced nearly identical mobility calculation results to the NVE ensemble (number of atoms, system volume and energy being controlled). However, Janssens et al [56] did not study the effect of stress on GB mobility.

We describe the stress state as combinations of the scalar hydrostatic pressure p and stress deviator tensor[1] components $\tilde{\sigma}_{ij}$ such that the complete stress state is the tensor sum

$$\sigma_{ij} = -p\delta_{ij} + \tilde{\sigma}_{ij} \tag{2.1}$$

where δ_{ij} is the Kronecker delta and σ_{ij} is the Cauchy stress. This description separates the isotropic and deviatoric components of stress, providing a method to encode the pressure and shear stresses separately. The definition of pressure[2] being the negative mean normal stress or

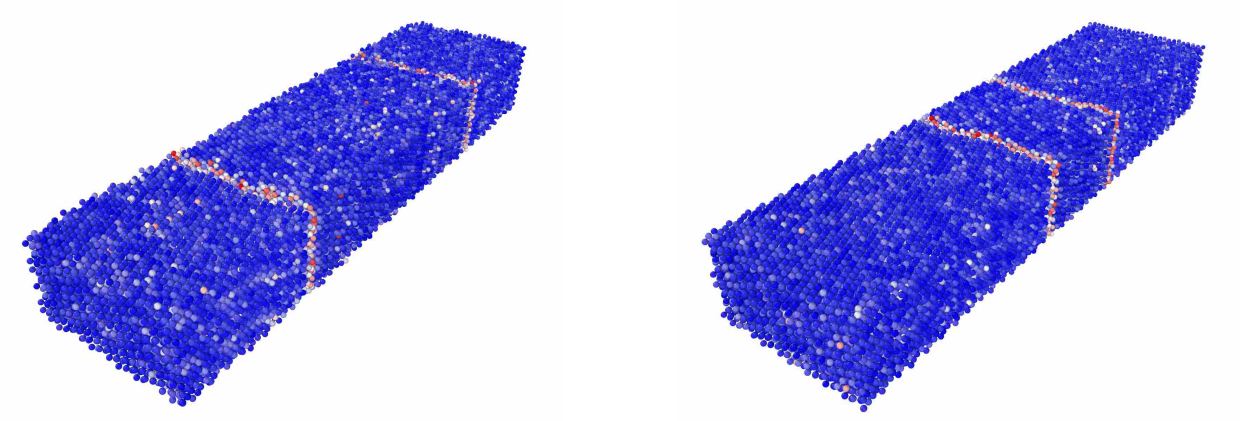
$$p = -\frac{1}{3}\sigma_{kk} = -\frac{\sigma_{11} + \sigma_{22} + \sigma_{33}}{3} \tag{2.2}$$

Therefore, if all three normal stresses ($i = j$) are equal, they are each equivalent to $-p$.

Each GB simulation was equilibrated for at least 10ps in the NVE ensemble to activate atoms from their reference state of 0°K to the analysis temperature, T. The initial assignment of the atomic velocities was introduced stochastically through a randomly generated seed integer. The system ensemble is subsequently changed to triclinic NPT. Finally, triaxial stress loads are applied. The temperature is set and the system is equilibrated for 40ps.

[1]Tensor mechanics are heavily utilized in this work. Important to note is the notation associated with tensor mechanics. Attempts are made to clearly identify scalars, vectors and tensors throughout this work. The primary method to distinguish these in notation between these is the use of subscripts. No subscript on a variable implies that it is a scalar, a single subscript ($[\ ]_i$) implies a vector, and two subscripts ($[\ ]_{ij}$) imply a rank-2 tensor and so on. In each subscripted case, the Einstein notation is implied ($y_i = \sum_{i=1}^{3} y_i$ and $A_{ij} = \sum_{i=1}^{3}\sum_{j=1}^{3} A_{ij}$). Direct notation could also be used, however, this is avoided for the sake of clarity

[2]Because sign conventions for compression vs. tension differ throughout the literature, we make explicit metion of the convention employed in this work. Stress states in LAMMPS are defined with compression being positive. The convention used in this work defines compressive stresses (σ_{ij}) as negative and compressive pressure (p) positive. We attempt to carefully clarify these sign conventions where needed.



(a) (b)

Figure 2.1: Schematic representation of a Σ11 29.5° bicrystal experiment as visualized in Ovito [55] (a) before synthetic driving pressure was applied and (b) 400ps after driving pressure was applied. Color indicates differing values of the centrosymmetry parameter. GBs are indicated by a high centrosymmetry. The crystal orientation with the assigned increase in potential energy for the synthetic driving pressure decreases in size (central grain). The low-energy grain grows as time progresses and grain boundary motion is observed.

2.2 Mobility Simulation

After the equilibration step, a mobility simulation was conducted. The objective of this simulation was to observe the translation of the GBs over time. To provide the driving pressure for GB motion to occur in the absence of GB curvature, a synthetic driving pressure was applied. The system was then permitted to evolve for up to 800ps. No changes to the boundary conditions were made during this mobility simulation.

The synthetic driving pressure was described and investigated in detail by Jannsens et. al. [56]. This method utilizes a crystal orientation order parameter to define differences between two grains in the bicrystal. One grain has an additional potential energy applied to atoms matching a specified crystal orientation order parameter. The difference in potential energy density between grains acts as a synthetic driving pressure (P). The difference in potential energy results in the high-energy orientation being unfavored and the low-energy orientation being favored. Thus, the grain having lower energy grows at the expense of the high energy grain. This results in a reduction of total system energy through reducing the number of atoms associated with the high energy grain. Snapshots of one of these simulations are shown in figure 2.1.

2.3 Errors in MD Driving Force Method

The synthetically applied energy method has some noted issues as discussed by Olmsted et. al. [57]. The primary issue was not initially resolved in this research and is related to the amount of synthetically applied free energy to the energetically preferred grain. In some cases, the orientation order parameter that defines the high energy grain is very close to the orientation order parameter for what should be the low energy grain. Improper assessment of the cutoff limits of the orientation order parameter in the LAMMPS function call results in some energy being assigned to the grain being energetically preferred (where no additional energy should have been assigned). Current work is being conducted to further resolve these differences and errors and corrected values will be presented in a future publication. This error and corresponding data are discussed at length in Appendix A.

The data from Appendix A is roughly summarized in Fig 2.2 for the STGB $\langle 110 \rangle$ GBs. This data shows that for low angle symmetric tilt grain boundaries a high amount of error for mobility calculations is anticipated due to low or zero driving force difference between the two grains. This trend maintains across all low angle STGBs due to the low difference in orientation order parameter between the two grains in the bicrystal simulation cell.

Figure 2.2 shows regions where low angle GBs likely contaminate results produced by the synthetic driving force method. We will adopt the convention of noting these regions (highlighted in dashed red boxes). This convention will be used throughout the remainder of the text to indicate regions where a mis-assignment of potential energy has likely occurred due to the proximity of the orientation order parameter in the two grains. These regions indicate areas where there exists low confidence in the data produced because of the driving force misassignment error. These indicated regions are based on estimates given the data provided in appendix A.

The ATGBs do not incur a similar feature of near-proximity orientation order parameter. Thus, the ATGBs can be considered approximately accurate. Minor effects (even within ATGBs) of the misassignment of MD driving force can be observed as some atoms achieve an orientation order parameter that is within the cutoff region due to thermal fluxuations. This error is typically less than 7%. We can thus regard most ATGB results as being reasonably accurate.

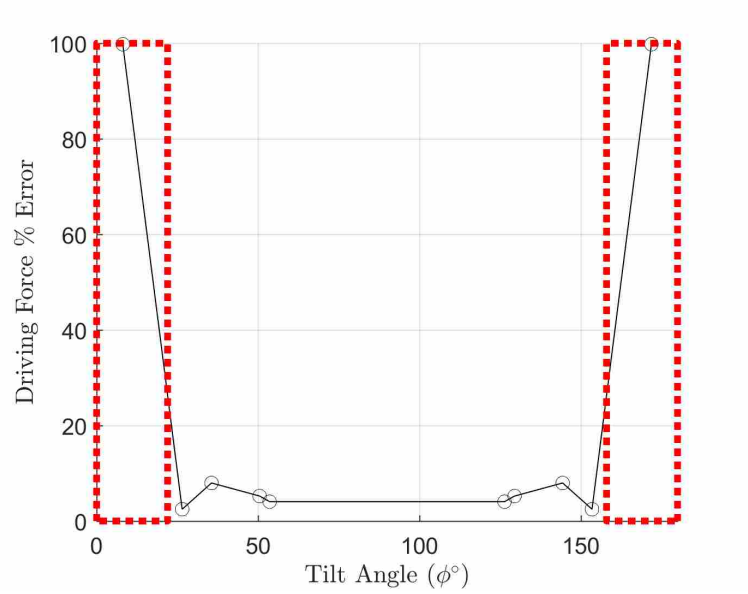


Figure 2.2: The error in the applied mobility against actual mobility vs changes in the tilt angle in STGB $\langle 110 \rangle$. This data shows that for low angle tilt GBs a significant error between the applied driving force and the actual driving force in the simulation is non-negligible. Dashed lines indicate regions where simulated data likely do not produce acceptably accurate results.

2.4 Determining the position of the GB

The GB position at each time step was computed after the mobility experiment. Due to the computationally intensive nature of LAMMPS, it was found that there were significant computational advantages in not performing post-processing steps while the main calculations were being performed. Even utilizing LAMMPS to compute the centrosymmetry parameter came at a net computational expense.

Instead, OVITO, an open-source MD visualization & post-processing tool was utilized to compute the positions of the two grain boundaries throughout the mobility experiment [55]. The grain boundary positions were determined by computing the centrosymmetry parameter for each atom in the simulation cell. The centrosymmetry parameter is a scalar value that indicates the local disorder. A high centrosymmetry indicates higher disorder, and a grain boundary can be clearly indicated by a high density of high centrosymmetry localized around a specific region.

An algorithm was developed that tracks the density of atoms weighted by the centrosymmetry parameter with respect to displacement in the x,y or z directions. In the case of the simulation cells used in this research, all grain boundaries were parallel to the x-z plane, so

changes in the centrosymmetry parameters with respect to y were used to determine the GB positions. A kernel density estimation or kernel smoothing density (KSD) function with a smoothing parameter and weighted according to the centrosymmetry parameter was used to obtain an effective "centrosymmetry density" throughout the simulation cell [58]. This KDE function $f(y)$ is defined as

$$f(y) = \frac{1}{nh}\sum_{i=1}^{n} w_i K\left(\frac{y-y_i}{h}\right) \tag{2.3}$$

where n is the number of points, h is a positive-non-zero smoothing parameter, w_i are the weights and K is the normal distribution function. The centrosymmetry parameter for each atom was used as the weighting parameter w_i.

Because the simulation cells were fully periodic, a circular form of the KSD function (CKSD) was obtained via a publicly avialable script by Vlad Atanasiu [59]. The peaks of the density estimation function were taken as the positions of each of the grain boundaries. Figure 2.3 shows a snapshot of one timestep using this CKSD function. The distance between grains (indicated in red arrows in Fig. 2.3) is taken as the minimum of the two distances between grains (there are two because of the periodicity).

2.5 Calculation of GB Velocity

The GB velocity was computed by tracking positions of the two GBs over a maximum time frame of 800ps. Due to the periodicity, GB motion occurs in both GBs of the simulation cell in equal magnitude. The GB velocity was then extracted as a rate of distance between GBs[3] over time. Simple curve fitting of the distance between GBs provides twice the GB velocity (as both GBs are moving). Thus the velocity of the GB (v) is defined as

$$v = \frac{1}{2}\frac{dy}{dt} \tag{2.4}$$

A key feature of the simulations performed with periodicity in all 3 dimensions is a lack of initial transients in the GB velocity. In simulations of dual-periodic systems, there is an early

[3]In order for GB velocity to be accurately represented using this method, it is assumed both GBs are displacing at equal rates. It is possible, however, that GB velocity is different between the two GBs. In that case this method simply reports the average of the two rates. The effect of different GB velocities was studied in Appendix B.

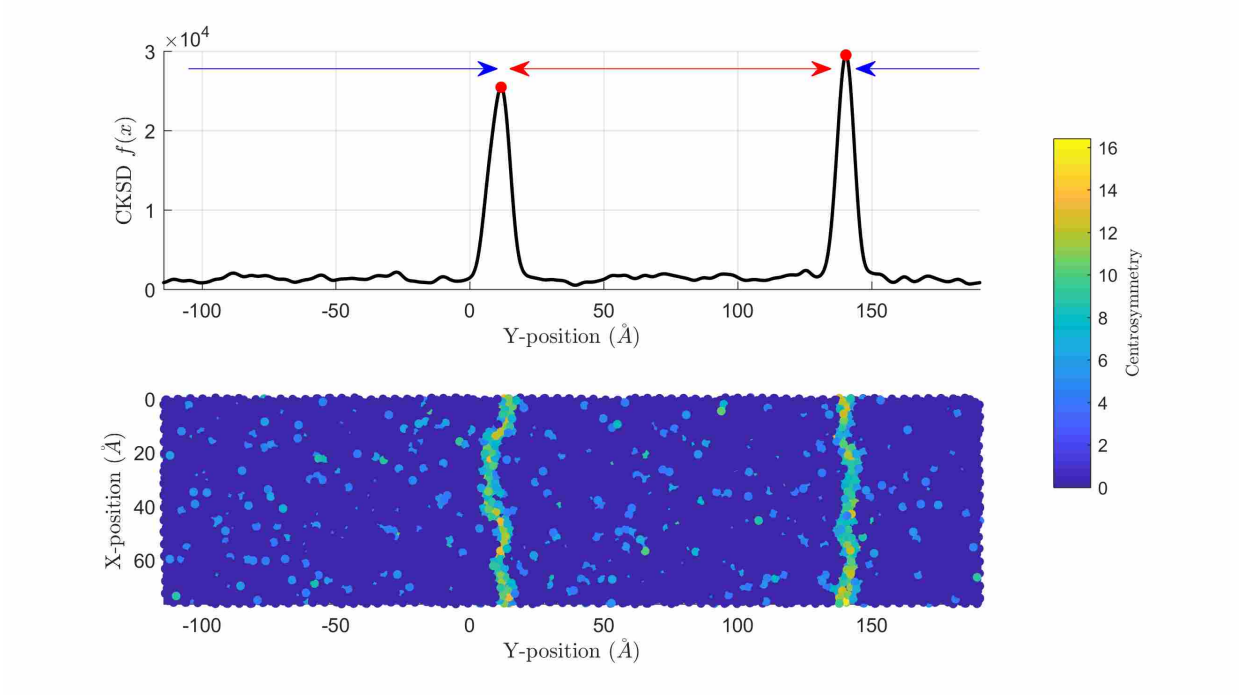


Figure 2.3: Sample output of GB position algorithm determining the position of GBs in the simulation cell. The two arrows indicate differing directions that the distance between GBs could be measured. Color of atoms in the lower part of the figure are values of the centrosymmetry parameter.

transient stage where grain boundary mobility is not linear for approximately 100ps. A transient portion to this degree is not observed in mobility experiments conducted with fully-periodic simulation cells.

2.6 Calculation of GB Mobility

The GBM, M is calculated through use of equation 1.1 after determining the GB velocity v. The applied synthetic driving pressure P was specified. Thus the conversion from GB velocity (v) to GBM (M) is defined as

$$M = \frac{v}{P} \tag{2.5}$$

It is through these conversions and calculations of mobility that we can take any desired synthetic driving pressure magnitude from the Jannsens' driving pressure method and normalize it into a calculated mobility. In simulations with a higher synthetic driving pressure the GB velocity is

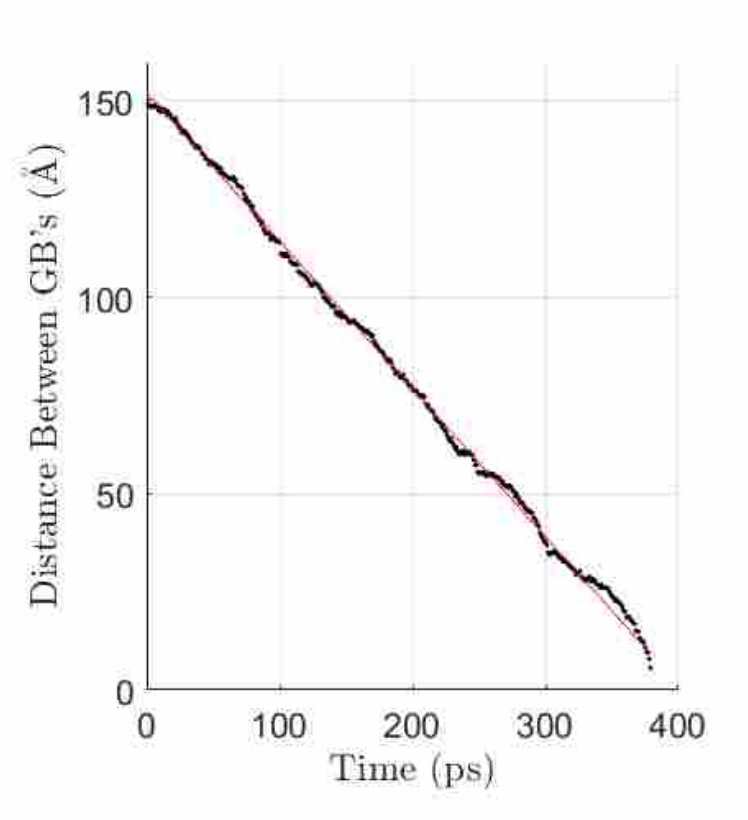


Figure 2.4: Sample output of GB tracking algorithm showing the convergence of two grains due to an applied synthetic driving pressure. The slope of the fitted line is taken as double an individual GB velocity.

proportionately higher than the lower synthetic driving pressure. In this work no differences in trends were observed with different magnitudes of synthetic driving pressure.

The synthetic driving pressure is applied as an energy per atom (eV/atom). Volume of the unit cell (and number of atoms per unit cell) is used to convert units between the synthetic driving pressure (eV per atom) and pressure (GPa) for use as the driving pressure as in equation 1.1.[4] We may explicitly detail the calculation of the mobility including these conversions as

$$M = \frac{v}{fna^3} \tag{2.6}$$

[4]Conversion from eV/atom to 1/P takes place in a similar fashion to a units conversion like [atom/eV][unit cell/atom][volume of unit cell/unit cell] which is a unit of pressure and may be directly converted via standard units conversion.

where f is the synthetically applied addition of energy[5] per atom (eV/atom), n is the number of atoms per unit cell, and a is the edge length of the FCC unit cell ($\AA^3$). By utilizing this equation we have a measure of pressure in the denominator ($eV/\AA^3$). Simple units conversion can convert this to more standard units of pressure (Pa). This calculation is implied when discussing the synthetic driving pressure from this point forward and is regarded as a minor detail. It is reported here for exhaustive clarity.

2.7 Design of Experiments

By utilizing triclinic simulation cells in the NPT ensemble, we can make use of the full stress tensor. Accordingly, the factor space in this work could be 8-dimensional (temperature, synthetic driving pressure and 6 independent stress terms) each dimension being continuous. However, due to computational constraints the factor space was reduced to 5-dimensional (temperature, pressure, two shear stresses and synthetic driving pressure). The fifth factor (synthetic driving pressure) was treated as a categorical factor with two levels to evaluate the influence of synthetic driving pressure.

To completely interrogate the factor-space, a full-factorial design of experiments (DOE) is desired. However, to further reduce the computational demands, a latin hypercube DOE [60] (sometimes referred to as a sphere-packing DOE) was utilized to interrogate the factor space. Latin hypercube DOEs are capable of restricting the number of experimental samples while still exploring the continuous factor space in a manner similar to the behavior of full-factorial DOEs, thus reducing the number of required simulations.

Extremely high compressive hydrostatic pressures (up to 5GPa) were used within the pressure response space. These pressures are reasonably obtained in modern large volume ultra-high pressure presses typically designed for the processing of ultra-hard polycrystalline compacts [61]. High compressive hydrostatic pressure was previously studied up to 1 GPa by Molodov et. al. [27] and up to 2.8 GPa by Hahn et. al. [28]. Tensile hydrostatic stresses (negative pressures) were also included in the DOE, but of much smaller magnitude (maximum of 0.5 GPa hydrostatic tensile stress).

[5] See information regarding problems with the synthetically applied energy as noted previously in Section 2.2 and also detailed in Appendix A.

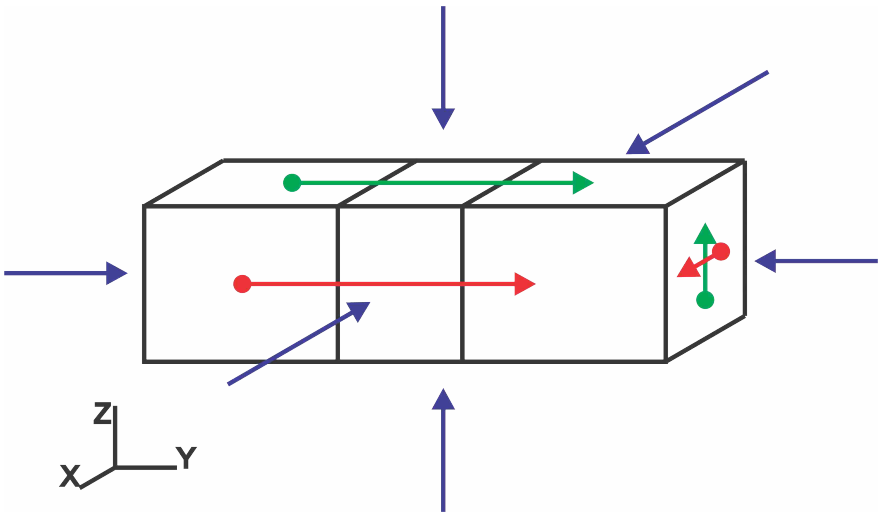


Figure 2.5: Schematic representation of bicrystal stress loading scheme. Color indicates differing stress conditions applied. The hydrostatic pressure (p) is colored blue, σ_{12} is colored red, and σ_{23} is colored green.

The two shear stresses selected for the present work are σ_{12} and σ_{23}. These shear stresses are purely deviatoric being independent of the hydrostatic pressure. As mentioned previously, all GBs were parallel to the x-z plane (with the GB normal parallel to the y-axis), thus σ_{12} and σ_{23} act on the simulation cell as shown in Figure 2.5. Shear stresses act on the GB plane in the same way as all other planes parallel to the x-z plane due to the uniformity of the stress field in the simulation cell. These shear stresses induce a traction force ($t_i = \sigma_{ij} \cdot \hat{n}_j$) that acts on the GB plane in a direction that is not purely parallel to the GB plane normal. This effectively produces a shearing stress analogous to other MD studies on the effect of shearing of the GB [19, 48–50, 62]. The extent of shear stress was limited to a maximum of 1.25 times the polycrystalline yield at elevated temperature (130 MPa [63]).

CHAPTER 3. THEORY

3.1 Thermodynamics: External Stress Work Contributions

In order to discuss the kinetics, we must first establish the thermodynamics with the associated application of an external stress. A differential change in internal energy U is defined as

$$\partial U = T\partial S - \partial W \tag{3.1}$$

There are two main factors, the first being the the entropic term (TS) where T is the temperature and S is the entropy, and the second being an applied mechanical work term (W). In this case, the applied mechanical work uses the convention of work performed on the system.

The work term can be split into body forces (W_b) and surface forces (W_s)

$$\partial W = \partial W_b + \partial W_s \tag{3.2}$$

which can be expanded to

$$\partial W = \underbrace{\int_\Omega b_i \partial u_i dV'}_{\partial W_b} + \underbrace{\int_{\partial\Omega} t_i \partial u_i d\Gamma}_{\partial W_s} \tag{3.3}$$

where b_i is a body force, ∂u_i is the change in displacement vector field, t_i is the surface traction force vector field, $\partial\Omega$ is surface that bounds the body of Ω, V' is the instantaneous volume (spatial reference frame) of the body, and $d\Gamma$ is a differential area for integration about the surface [64].

Upon application of (1) the balance of linear momentum (Cauchy's first law), (2) the divergence theorem and (3) utilizing the symmetry of stress (Cauchy's second law), the above equation describing the internal energy due to external work acting on the system is simplified to

$$\partial W = \int_\Omega \sigma_{ij} \partial \varepsilon_{ij} dV' \tag{3.4}$$

If the system has no stress/strain gradient (as is true in atomically small systems like those analyzed in this thesis), we may then appropriately reduce this further to

$$\partial W = \sigma_{ij} \partial \varepsilon_{ij} V' \tag{3.5}$$

Where σ_{ij} is the Cauchy stress, and ε_{ij} is the Hencky[1] (true) strain. Thus we can write the applied mechanical work as

$$W = \sigma_{ij} \varepsilon_{ij} V' \tag{3.6}$$

and by expanding these terms into hydrostatic and deviator components we have

$$W = p \varepsilon_{vol} V' - \tilde{\sigma}_{ij} \tilde{\varepsilon}_{ij} V' \tag{3.7}$$

where p is the pressure, ε_{vol} is the volumetric strain ($\ln(V'/V_0)$), V' is the instantaneous volume, $\tilde{\sigma}_{ij}$ are components of the stress deviator tensor, and $\tilde{\varepsilon}_{ij}$ are components of the strain deviator tensor.

3.2 Thermodynamics: Free Energy

The combined influence of temperature, hydrostatic stress and deviatoric stress is proposed to be modeled through a Gibbs type thermodynamic potential energy (G) that includes the effect of non-hydrostatic stress work as in Eq. 3.7. Various definitions of this potential energy exist throughout the literature using a wide array of definitions for the thermoelastic work [38, 64]. The form discussed in Eq. 3.7 is a simple form that matches the stress measure used in controlling the system, the Cauchy stress. By employing a suitable Legendre transformation [65], we can define the Gibbs free energy as

$$G = U - TS + p \varepsilon_{vol} V' - \tilde{\sigma}_{ij} \tilde{\varepsilon}_{ij} V' \tag{3.8}$$

where all terms are as have been defined previously.

From Eq. 3.8 we combine $\varepsilon_{vol} V'$ into a single term that we name (for clarity) the isostatic volume (V). This is the volume change due to the contributions of a hydrostatic stress. We similarly

[1]The thermoelastic work in the purest sense is defined using D_{ij} the stretching tensor. This tensor is equivalent to the Hencky (logarithmic) strain if the principal stretch directions are stationary in the deformation process. This constraint is required so that path-dependency may be consistently accounted for. The deformation processes we impose is consistent with these constraints.

combine $\hat{\varepsilon}_{ij}V'$ terms and refer to this combined term as a volume deviator tensor[2] (Λ_{ij}). Thus, equation 3.8 becomes

$$G = U - TS + pV - \tilde{\sigma}_{ij}\Lambda_{ij} \tag{3.9}$$

A similar description has also recently been used by Ball, Alexander & Schuh [66] in their work modulating stress for kinetic processes.

The incremental change in energy of this form is

$$dG = -SdT + Vdp - \Lambda_{ij}d\tilde{\sigma}_{ij} \tag{3.10}$$

From these relations, we are left with the natural variables of temperature, pressure and the stress deviator tensor (T, p, $\tilde{\sigma}_{ij}$). This form is desirable as it is the energetically appropriate model for stress-driven processes and matches the thermodynamic ensemble used in the molecular dynamics simulations of this work.

3.3 Arrhenius Behavior: Kinetics

We postulate that mobility exhibits an Arrhenius-type temperature dependence with an activation free energy G^*, defined by $G^* = G_a - G_0$ where G_0 is the free energy of the ground state and G_a is the energy of a postulated activated state through which the GB must pass to continue in the evolution of the kinetic process. Expressing the mobility in this fashion and making use of Eq. 3.9, we have

$$M = \alpha \exp\left(\frac{-G^*}{kT}\right) \tag{3.11a}$$

$$= \alpha \exp\left(\frac{S^*}{k}\right) \exp\left(\frac{-U^*}{kT}\right) \exp\left(\frac{-pV^*}{kT}\right) \exp\left(\frac{\tilde{\sigma}_{ij}\Lambda^*_{ij}}{kT}\right) \tag{3.11b}$$

[2]Although deviatoric processes are (strictly speaking) volume conserving, we make use of this form and call the deviatoric extensive variable a "volume" for convenience. The units of the combined term are m^3. Thus, labeling this extensive variable as a volume is appropriate.

The non-temperature-dependent term contains the activation entropy S^*, and can be incorporated into the pre-exponential giving

$$M = M_0 \exp\left(\frac{-U^*}{kT}\right) \exp\left(\frac{-pV^*}{kT}\right) \exp\left(\frac{\tilde{\sigma}_{ij}\Lambda^*_{ij}}{kT}\right) \tag{3.12}$$

If desired, the remaining exponential terms could be combined to define an activation enthalpy, which would correspond to the traditional activation energy for the process. However, leaving the terms expanded allows us to define the activation parameters

$$U^* = -kT^2 \frac{\partial \ln(M)}{\partial T} \tag{3.13a}$$

$$V^* = -kT \frac{\partial \ln(M)}{\partial p} \tag{3.13b}$$

$$\Lambda^*_{ij} = kT \frac{\partial \ln(M)}{\partial \tilde{\sigma}_{ij}} \tag{3.13c}$$

Where U^* is the activation energy, V^* is the activation volume conjugate to isostatic stress, and Λ^*_{ij} is an activation volume deviator tensor component conjugate to the corresponding i, j stress deviator tensor component.

These activation parameters can be interpreted as sensitivity measures of mobility to the conjugate intensive state variables. As intensive state variables are modulated, the energetic barrier to the kinetic process either increases or decreases. For example, as the magnitude V^* increases, the effect of pressure becomes more dominant. A negative V^* suggests that pressure enhances the kinetic process. These trends are made explicit by the form of Eq. 3.13b.

These activation parameters may also provide greater insight into mechanisms of GB motion. V^* may be instructive in determining the volume required for a processes such as vacancy diffusion. Λ^*_{ij} may be used in quantifying the shear deformation of the GB unit cell or deformation of the crystal unit cell required for GB motion.

The present work is limited in scope to modulating specific portions of the stress deviator tensor. All other stress deviator components are held at zero. The mobility in the present work, therefore, is modeled as

$$M(T, p, \tilde{\sigma}_{12}, \tilde{\sigma}_{23}) = M_0 \exp\left(\frac{-U^*}{kT}\right) \exp\left(\frac{-pV^*}{kT}\right) \exp\left(\frac{\tilde{\sigma}_{12}\Lambda^*_{12}}{kT}\right) \exp\left(\frac{\tilde{\sigma}_{23}\Lambda^*_{23}}{kT}\right) \tag{3.14}$$

Validation of the proposed model was accomplished by evaluating Pearson's Correlation Coefficient (R^2) for multi-dimensional linear regression of Eq. 3.14 against MD mobility calculations that modulate intensive variables (T, p, $\tilde{\sigma}_{12}$ and $\tilde{\sigma}_{23}$). The best fit parameters resulting from the regression analysis provide estimates for the conjugate extensive quantities (ΔU, V^*, Λ^*_{12} and Λ^*_{23}).

CHAPTER 4. MODEL PERFORMANCE

4.1 Suitability of Kinetic Model

From this study we observed that mobility generally does change in a superimposed manner with combined thermal, pressure and shear loads. This suggests that if a specific mobility is desired, mobility may be tuned by application of mechanical loads. Once activation parameters were determined through linear regression, we can calculate a predicted behavior for GB mobility. The correlation between the model-based predicted mobility and the calculated mobility for all 98,786 simulations is shown in Fig. 4.1. We can observe the overall model behavior by the Pearson's Correlation Coefficient (R^2) of these data. As shown in figure 4.1 The model prediction proposed in Eq. 3.14 fits well ($R^2 = 0.936$ $RMSE = 45.665$) to calculated (simulated) mobility in the case of analyzing all data collectively.

The fidelity of this model demonstrates the ability to use a characteristic state function, like was developed in Eq. 3.12, to predict the effect of the state variables on GBM. It is important to note that deviatoric stresses in four additional tensor components were controlled, but not modulated. The quality of this prediction suggests that modulation of these states could reveal additional mechanisms for GBM.

Four grain boundary characters (GBCs) were randomly selected and their predicted vs simulated mobility trends are shown in Fig. 4.2 a-d. Within each selected GBC, we may see that the model generally predicts well with varying degrees of quality. In the $\Sigma 11$ 5.05° sample, we can see a high density of low simulated mobility. It is clear from this data, that this high density of low mobility influences the overall performance of the predictive model for that GBC. The other data are much better suited to the model where intermediate levels of mobility are reported.

When overlaying several of these characteristic fits for each GBC within their respective ATGB or STGB class, we may see general trends on the suitability of the model within each class.

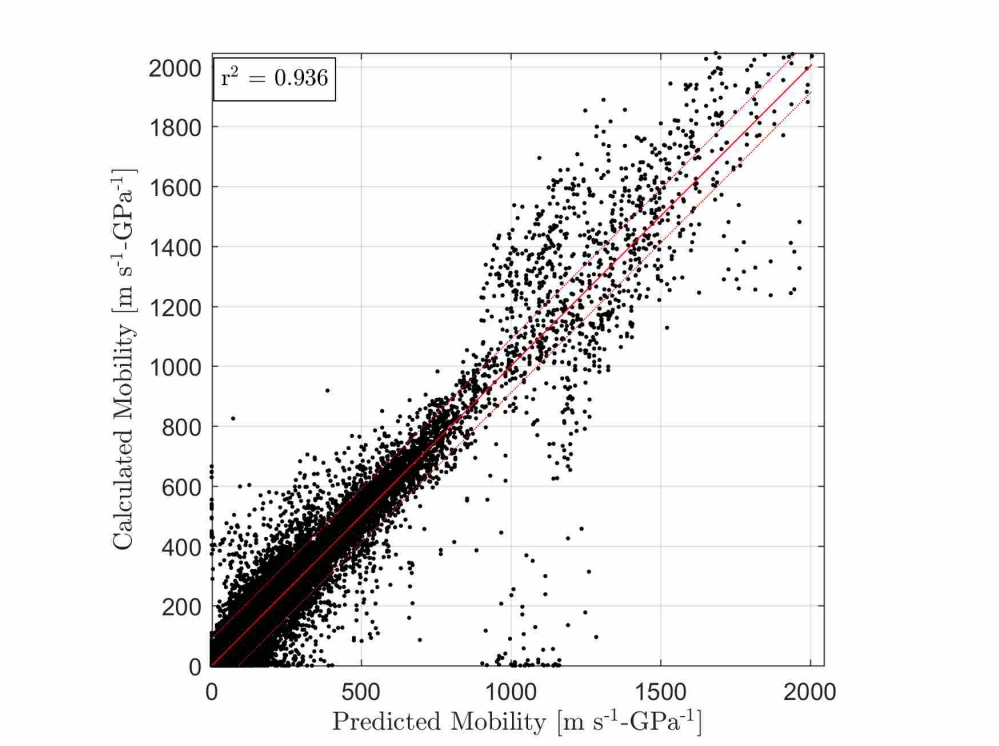


Figure 4.1: Modeled mobility vs calculated (simulated) mobility for all 98,786 simulations performed (all thermoelastic states, all GBCs all driving pressures). All GBCs were modeled separately and overlayed on this plot. Best fit line and 90% confidence intervals are shown in red and red-dashed lines respectively.

Figure 4.3 shows the combined GBM prediction vs simulated data using this grouping method. Within each class, we observe agreement between predicted and simulated mobility.

It is apparent that some GBs (Σ13a and STGB ⟨100⟩) appear far more predictable than others. To further interrogate this effect, we investigate the role that incliniation or tilt angle (ATGB or STGB respectively) has on the predictability of the model. Figure 4.4 shows model R^2 with respect to the inclination or tilt angle for each GB class. It is in Fig. 4.4 that we can observe some low angle GBs (STGB ⟨100⟩) to have very poor quality fits. Also of particular note is the Σ9 GB within the STGB ⟨111⟩ class being of particular poor quality.

There are many GBCs that do not fit well at all. Many of these that do not fit well have minimal mobility regardless of thermoelastic state. Again, the synthetic driving force method's error may be a determining cause for these errors. Alternatively, the extent of the DOE thermoelastic states may not have been large enough for some of these GBs to provide meaningful information to the fitting routine for extractio of model parameters.

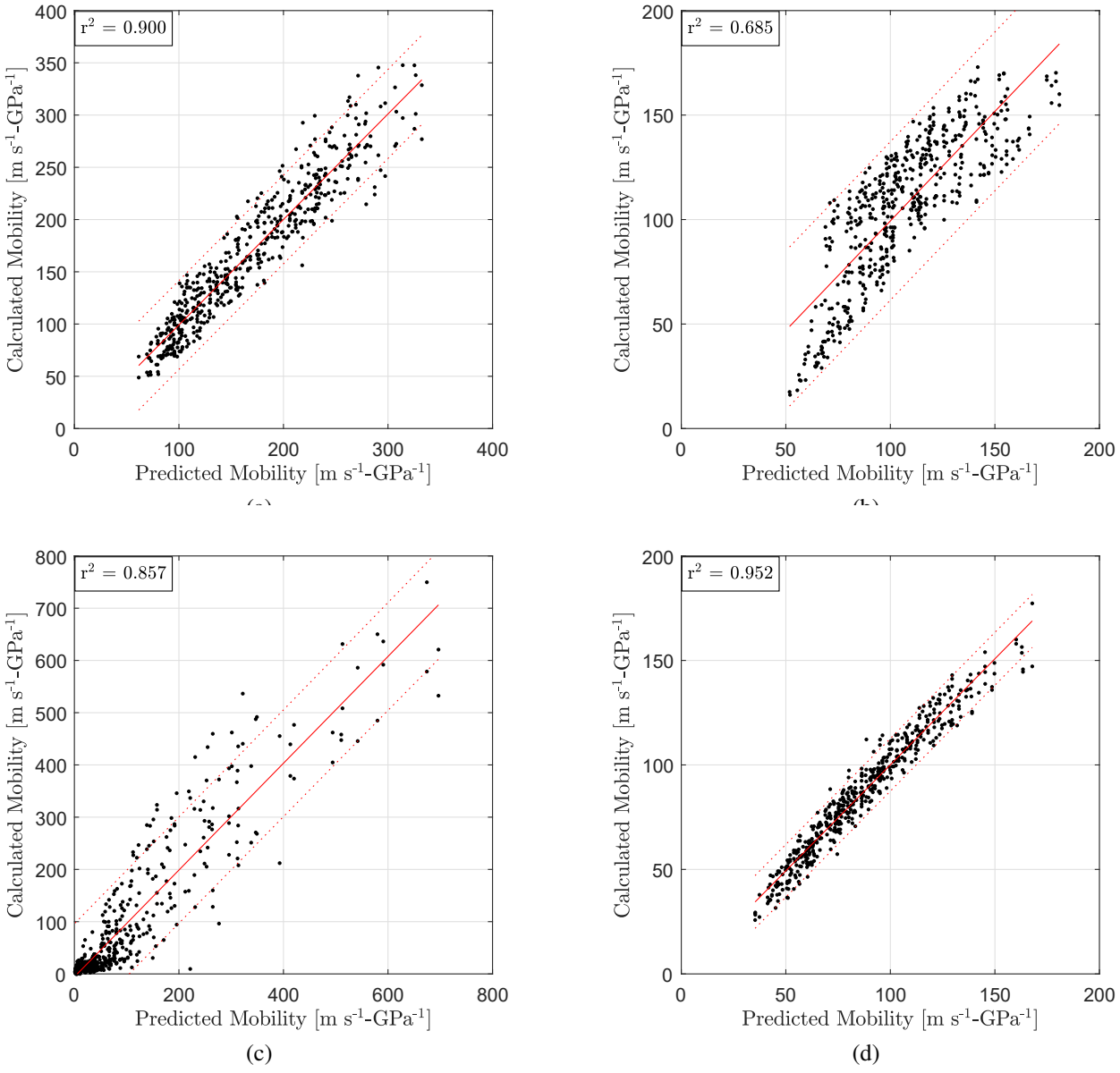
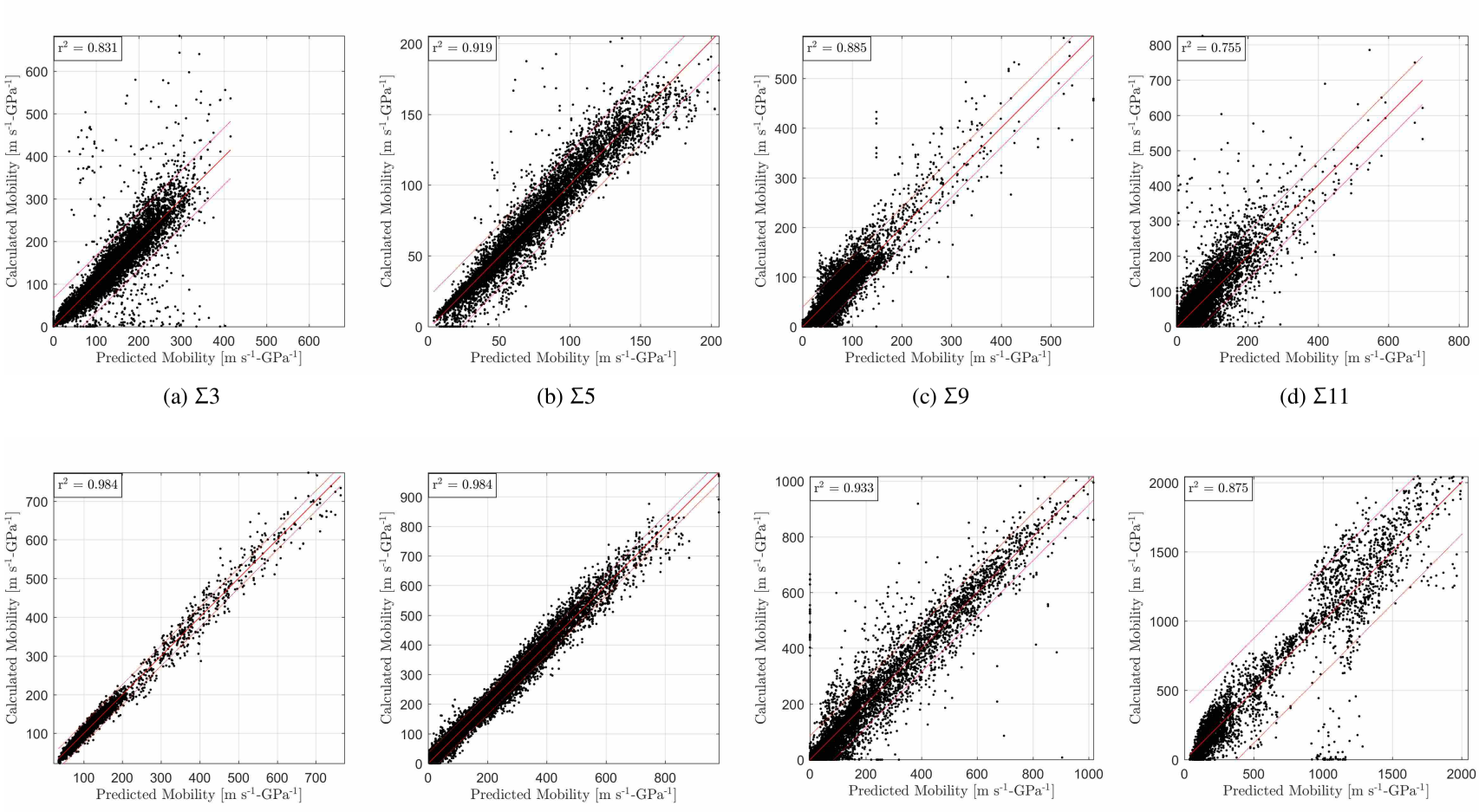


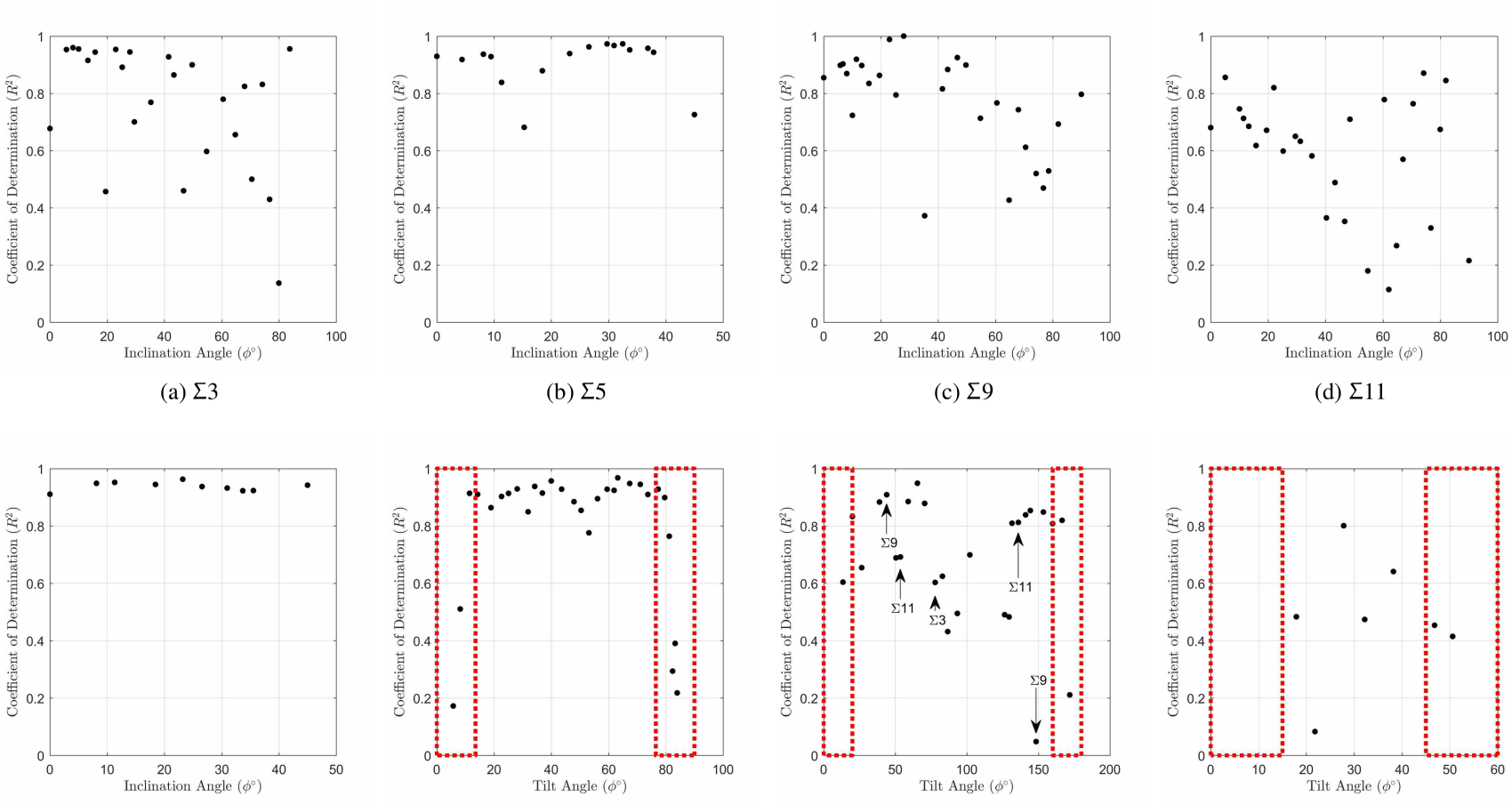
Figure 4.2: Four randomly selected examples of the thermoelastic model fitting on individual GBCs. The GBCs are (a) Σ3 49.68° (b) Σ5 15.26° (c) Σ11 5.05° and (d) Σ13a 11.31°. These demonstrate the effectiveness of fitting GB mobility utilizing the thermodynamic model for a single grain boundary structure. Red lines are linear correlations of the predicted vs simulated mobility with 95% confidence intervals shown in dashed red.



(a) Σ3 (b) Σ5 (c) Σ9 (d) Σ11

(e) Σ13a (f) STGB ⟨100⟩ (g) STGB ⟨110⟩ (h) STGB⟨111⟩

Figure 4.3: The model predicted mobility vs the calculated (simulated) mobility (via MD) combined across all simulations performed on all planes within noted GB class. Red lines show linear fits of all data displayed with dashed red lines as 95% confidence intervals of the linear correlation. The slopes of all lines are nearly 1.



(a) Σ3 (b) Σ5 (c) Σ9 (d) Σ11

(e) Σ13a (f) STGB ⟨100⟩ (g) STGB ⟨110⟩ (h) STGB ⟨111⟩

Figure 4.4: Pearson's correlation coefficient (R^2) as a function of the misorientation angle for the respective ATGB or STGB noted. Red regions indicate regions where the synthetic driving force order parameter likely has a misassignment such that both grains receive a synthetic free energy.

In all cases there are two related factors that influence the error in calculations of model parameters. The first being a required induced stochasticity in the NVE equilibration step. The random nature of the initial velocity assignment to each atom will always induce some variation from one calculation to the next. The second source being the scale to which the calculations were made. As simulation cell size increases it is anticipated that the effects of stochasticity are reduced. Computationally, this is impractical, so little improvement upon the overall predictability can be made when studying such a large dataset. The degree to which these factors actually influence the overall fit is unknown.

4.2 Influence of Known Factors of Error

There are two known influences of error that affect the accuracy of these results. First is the influence of misassigned synthetic driving force discussed in section 2.3 and further detailed in appdendix A. The second influence is that of the GB velocity calculation that averages the two GB velocities rather than computing GB velocities individually (first mentioned in section 2.5 and detailed in appendix B).

In Fig. 4.4 we make note of the convention employed in section 2.3. The red-boxed sections indicate areas where the synthetic driving force method does not accurately distinguish between the two different grain boundary orientation order parameters. In Fig. 4.4f, we see coincidence between those GBCs that have poor fitting to the model and those that exhibit the error in the synthetic driving force method. The error of the synthetic driving force method is likely the primary reason for the lack of fit to the thermodynamic model for STGB $\langle 100 \rangle$ GBs.

There are 13 GBCs that lie within regions that are highly susceptible to misassignment of synthetic driving force as shown in Fig. 4.4. These GBCs can be expected to have errors in actual mobility calculations in excess of 20% error. These 13 GBCs make up 7.4% of the full dataset. Thus due to this known source of error 92.6% of the dataset can be trusted to within an estimated 10% accuracy.

The second source of error (the GB velocity calculation), is noted in section 2.5 and discussed in detail in appendix B. This error does not appear to influence the STGBs to any degree, however, some evidence exists of a directional dependence in mobility in some but certainly not all ATGBs. For example, one $\Sigma 11$ 22.0° ATGB demonstrated a difference in GB

velocity under one thermodynamic state, and all other analyzed ATGBs in appendix B did not. In figure 4.4, we can see that the kinetic model for the $\Sigma 11$ 22.0° ATGB does not perform well ($R^2 = 0.6$), whereas the $\Sigma 9$ 90° ATGB (one case interrogated in appendix B) performs much better ($R^2 = 0.8$). These data suggest an interesting GB motion difference in some ATGBs that is further discussed in appendix B.

The results of the sampling of an alternative method for measuring the GB velocity (discussed in appendix B) shows that less than 5% of all ATGB simulations exhibited this behavior of different GB velocities based on direction. Even with those GBs that exhibited differing GB velocities, the method employed in the body of this work averages the two GB velocities. Any error due to the GB velocity calculations is considered a minor effect that may yield interesting results in some cases.

These two sources of error and corresponding data highlight that the model R^2 is a key feature in determining accuracy of the thermodynamic model to GB motion phenomena. As various forms of irregular behavior are exhibited (i.e. differing GB velocity in $\Sigma 11$ 22.0° ATGB), the model performance suffers. It is clear that these forms of error are definitively indicated in the model R^2 as demonstrated in Fig. 4.4.

By analyzing the model R^2 we may estimate the combined influence of these two known errors (synthetic driving force error and GB velocity error) along with other stochasticity to determine how reliable the data is. Of all the GBCs, 77% have an R^2 above 0.75. Combining data from appendix A and B, we may estimate the errors of these data as roughly $\pm 10\%$. However, having a low R^2 for one GBC does not necessarily mean that these two known errors are in effect.

Provided this data, it is estimated that reasonable confidence ($\pm 10\%$) can be applied to approximately 87% of the data. The remaining 13% of the data may have errors anywhere from 10-100%. The best indicator of confidence in the data is the model R^2 as shown in Fig 4.4.

4.3 Changes in Kinetic Model Performance by Suppression of Intensive/Extensive Conjugate Pairs

The viability of alternate thermodynamic models was also interrogated as predictors of MD simulated mobility. As the number of terms included in the thermodynamic model provided in Eq 3.14 is reduced we see a reduction in overall performance of the predictive model. Table 4.1 shows

Table 4.1: R^2 and root-mean-squared-error (RMSE) for predicted mobility vs calculated (simulated) mobility across all GBCs analyzed in this study. Each line shows a corresponding inclusion or exclusion of activation parameters utilized in determining the predicted mobility. A checkmark indicates inclusion of the activation parameter in the predictive model. Through inclusion of more activation parameters, the corresponding coefficient of determination is improved. Rows are sorted by improved R^2

U^*	V^*_{iso}	Λ^*_{12}	Λ^*_{23}	R^2	RMSE
				0.828	75.0028
			✓	0.83	74.3858
✓				0.853	69.2631
	✓			0.856	68.5945
✓			✓	0.856	68.5664
	✓		✓	0.859	67.8548
		✓		0.875	63.7529
		✓	✓	0.879	62.9038
✓	✓			0.88	62.8703
✓	✓		✓	0.883	61.9635
✓		✓		0.904	56.1327
	✓	✓		0.904	55.7919
✓		✓	✓	0.907	55.0031
	✓	✓	✓	0.908	54.7630
✓	✓	✓		0.933	47.0062
✓	✓	✓	✓	0.936	45.6658

the combined predicted vs simulated mobility across all GBCs analyzed for various complexity of predictive models. As the model progresses from the most reduced to the most complete, we observe an increase in the overall predictability of the model.

From these data we observe improvements in R^2 and RMSE with connections to specific terms of the applied model. The dominant effects appear to be temperature and pressure with Λ^*_{12} following closely behind. Interestingly, only minor improvements in predictability are observed as the Λ^*_{23} term is added to the model. Upon first inspection of the data, it appears that the Λ^*_{23} factor induces very little influence upon the model. However, some improvements in predictability are made with inclusion of Λ^*_{23}.

This again suggests that all model parameters that are required to accurately describe the GBM kinetics. Thermodynamic predictability of stress modulated GBM is reduced if terms are removed or missing from the kinetic model.

4.4 Evaluation of the Kinetic Model Using Validation Data

To further evaluate the kinetic model, one randomly selected GBC ($\Sigma 3$ 49.68°) was selected to perform a validation study on the model. Model validation is a process using one subpopulation of the full dataset (training set) to train a predictive model and determine if the model can predict another subpopulation of the dataset (validation set). The validation set is a subpopulation that was not used in training of the model. A single training set and validation set (TSVS) provides a single estimate of the predictability of the model.

By investigating multiple TSVSs within a the single GBC ($\Sigma 3$ 49.68°) we can determine the influence of the size of the training set on the variance in prediction due to subpopulation sample size. The $\Sigma 3$ 49.68° GB dataset had 518 calculations of GBM under 256 different thermoelastic states (two driving pressures). Using this concept of model validation, a total of 25,424 TSVSs were created that exist within the dataset of 518 calculations. The training sets had various population sizes, however, the validation set sample size was held at 100 samples(approx. 20% of the full dataset). A much larger population of TSVSs exist ($\sum_{k=20}^{418} \binom{518}{k}$), however, all possible combinations could not be investigated. The output of each TSVS study is a scalar estimate of the error (root-mean-square error)

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (\hat{M}_i^v - M_i^v)^2}{n}} \tag{4.1}$$

where n is the number of samples in the validation set, M_i^v is the i-th simulated value from the MD simulations of the validation set and $\hat{M}_i^v$ is the i-th predicted value of the mobility using the training set model for the validation set. [1]

The RMSE of validation data vs model data in all 25,424 TSVSs tested in this study are plotted in figure 4.5. As can be observed from the figure, average sampled *RMSE* decreases rapidly with just a few training samples. This suggests that in future work, a small number of samples may be required to make predictions that fit reasonably well. Using approximately 100 samples typically achieves a *RMSE* that is only marginally different than the full model *RMSE* having all 518 samples. From one perspective, this data suggests that this work may have oversampled the factor space (i.e. performed more simulations than were strictly necessary).

[1] The code utilized for this process is provided in appendix F.

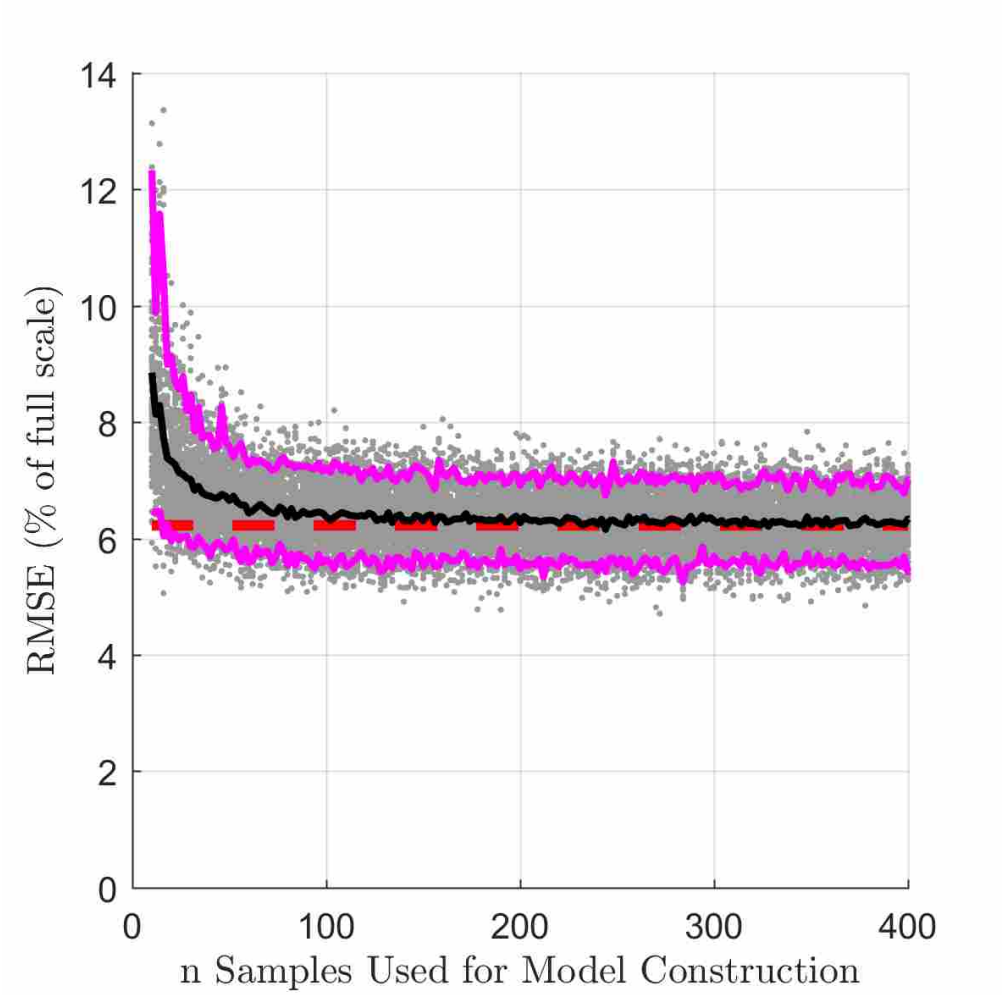


Figure 4.5: Calculations showing RMSE of validation set evaluated against the training model of training set sample size noted in the abscissa. 100 RMSE values for each sample size were obtained. Magenta lines show bounds of 95% of RMSE values for validation sets for a given sample size. Black indicates the average *RMSE* for all subpopulation sets having the same number of training data points. The red dashed line indicates the model *RMSE* if all data is used in model construction. All data is for one characteristic GBC (Σ3 49.68°).

4.5 Convergence of Model Parameters

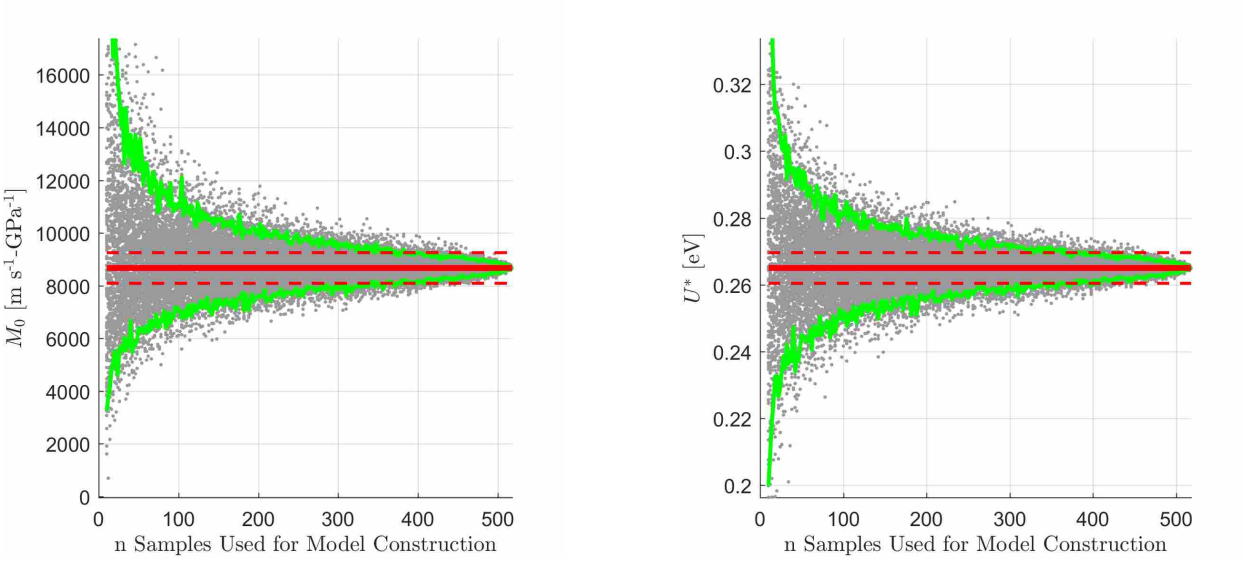
Model fidelity is improved by increasing sample size of MD simulations for a given GBC. Figure 4.6 shows the convergence of multiple subpopulations of data for an individual GBC (Σ3 49.68°). As population size increases predictability of the modeled parameters increases.

Although the model fits the data well with low numbers of training samples (as shown in Fig. 4.5), predictability of the model parameters requires many more samples until convergence is

met. A high number of samples is required to acceptably converge on the model parameters that are used throughout the remainder of this work.

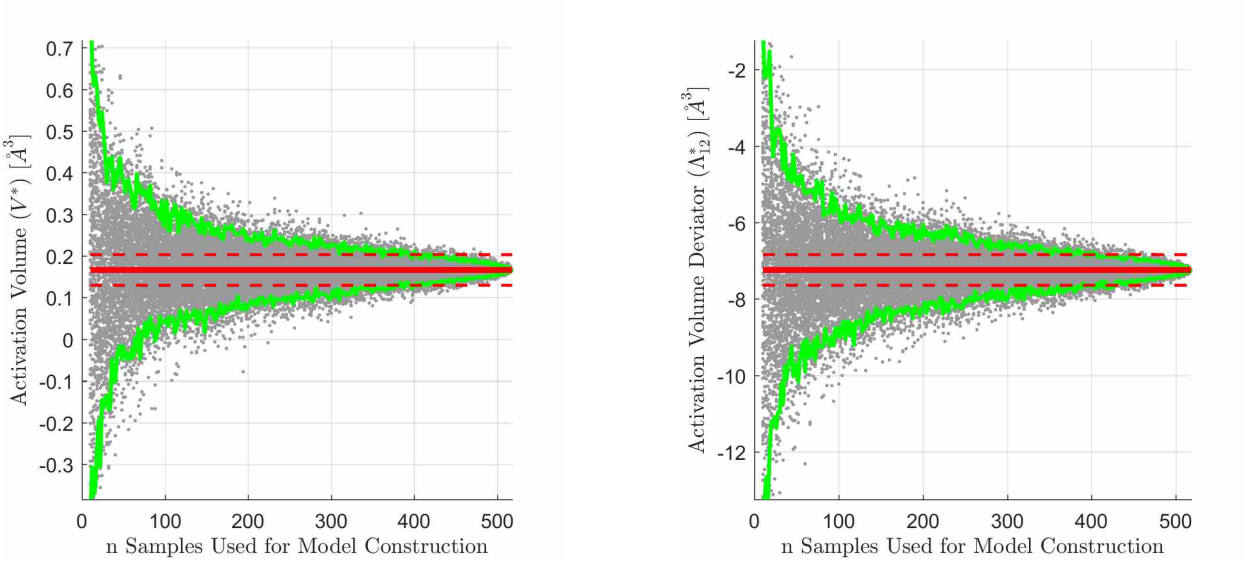
These data demonstrate a counterpoint to the oversampling implication above. In these data, a large number of samples is required to accurately converge on a solution for the model parameters. More than 400 samples are required to achieve model parameter results that would behave in a manner that is consistent with performance of 518 samples (within 95% confidence intervals of the model parameters using the full dataset of 518 samples). One possible reason for this might be the applied stochasticity of MD simulations. Other reasons might include MD simulation size, design space extent or other parameters for MD simulations not being considered in this model.

This being said, convergence upon a specified parameter is relatively smooth. Further studies of convergence could study the effect of a greater number of sampling points (requring more MD simulations). However, within the factor space studied, the MD calculation dataset is fairly exhaustive.



(a)

(b)



(c)

(d)

Figure 4.6: Calculations of the model parameters estimates from subpopulations of the full dataset for (a) M_0, (b) U^* (c) V^* (d) Λ_{12}^* (Λ_{23}^* is not shown). As data is added to the subpopulations, model parameter estimates are improved. 95% of the parameter estimates of a given subpopulation size exist within the bounds of the green lines. Error bars for each parameter estimate are not shown. The final or "target" estimate (solid red) with 95% confidence intervals (dashed red) are also shown.

CHAPTER 5. MODEL COEFFICIENTS

This chapter discusses the model using all mobility calculation data points for all GBCs and the corresponding model coefficients. The intent is to interrogate trends in model parameters within the GBC space.

5.1 Trends in GBM Kinetics - Pre-exponential Factor M_0

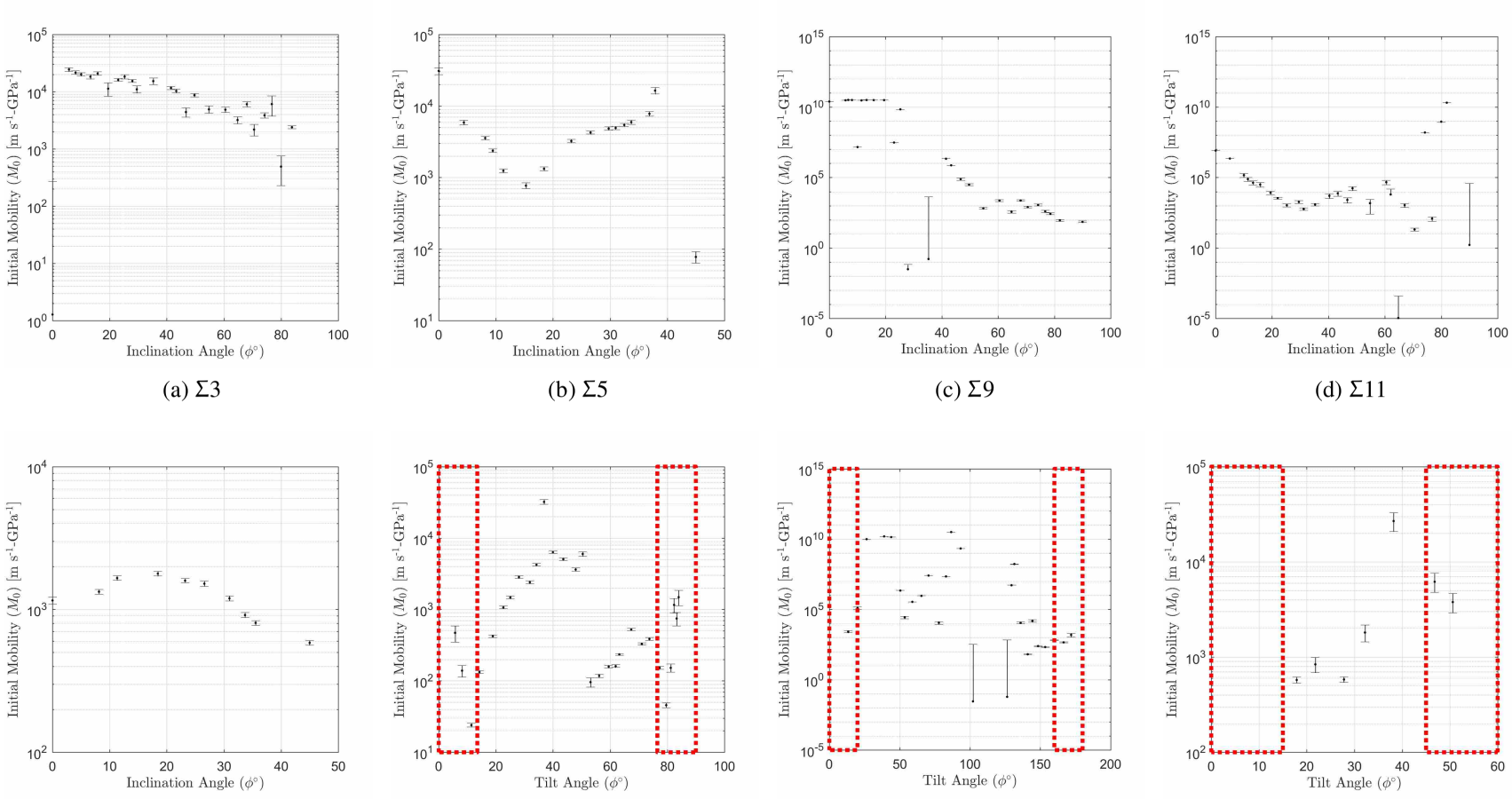
The first and most fundamental model parameter is the exponential prefactor M_0. In the context of a fitting parameter, M_0 indicates the base mobility observed in all MD simulations performed for that GB character. This M_0 is the mobility that would be observed if the remaining exponential terms resulted in zero exponents. The case of zero exponents would be observed if (in the limit) temperature was very large, pressure was zero, and all stress deviator terms are also zero. M_0 does not indicate sensitivity to any intensive parameter, but it does provide a sense for what the mobility could be in a highly activated state.

M_0 is related to the activation entropy of the kinetic model via

$$S^* = k_b \left(\ln(M_0) + \ln(\alpha)\right) \tag{5.1}$$

as a consequence of this relationship, we consistently plot M_0 on a log scale to indicate changes in activation entropy. A high S^* (and consequently high M_0) indicates a large number of configuration states available for the kinetic process to occur.

Figure 5.1 shows M_0 across all GBCs analyzed in this work. What can be observed from this data is that there are, in some cases, smooth trends of M_0 with respect to changes in inclination angle or tilt angle for ATGB and STGBs respectively. Some GBs have extremely low M_0 and these GBCs correspond to GBCs that do not fit the kinetic model well.



(a) Σ3 (b) Σ5 (c) Σ9 (d) Σ11

(e) Σ13a (f) STGB ⟨100⟩ (g) STGB ⟨110⟩ (h) STGB ⟨111⟩

Figure 5.1: Model parameter M_0 versues the GB tilt or inclination angle for the respective ATGB or STGB noted.

The smooth trends observed in 5.1 provide evidence of structural influence on mobility. Additionally, the data suggest that activation entropy is similarly influenced by GBC. Interestingly, there exist some near-zero M_0 measurements. This implies a very low activation entropy.

5.2 Trends in GBM Kinetics - Activation Energy U^*

The activation energy, U^* is the model parameter that indicates sensitivity to temperature. A high U^* expresses strong thermally activated behavior- that mobility increases as temperature increases. A low U^* illustrates a weak thermally activated behavior. A negative U^* indicates thermally damped behavior- that mobility decreases as temperature increases.

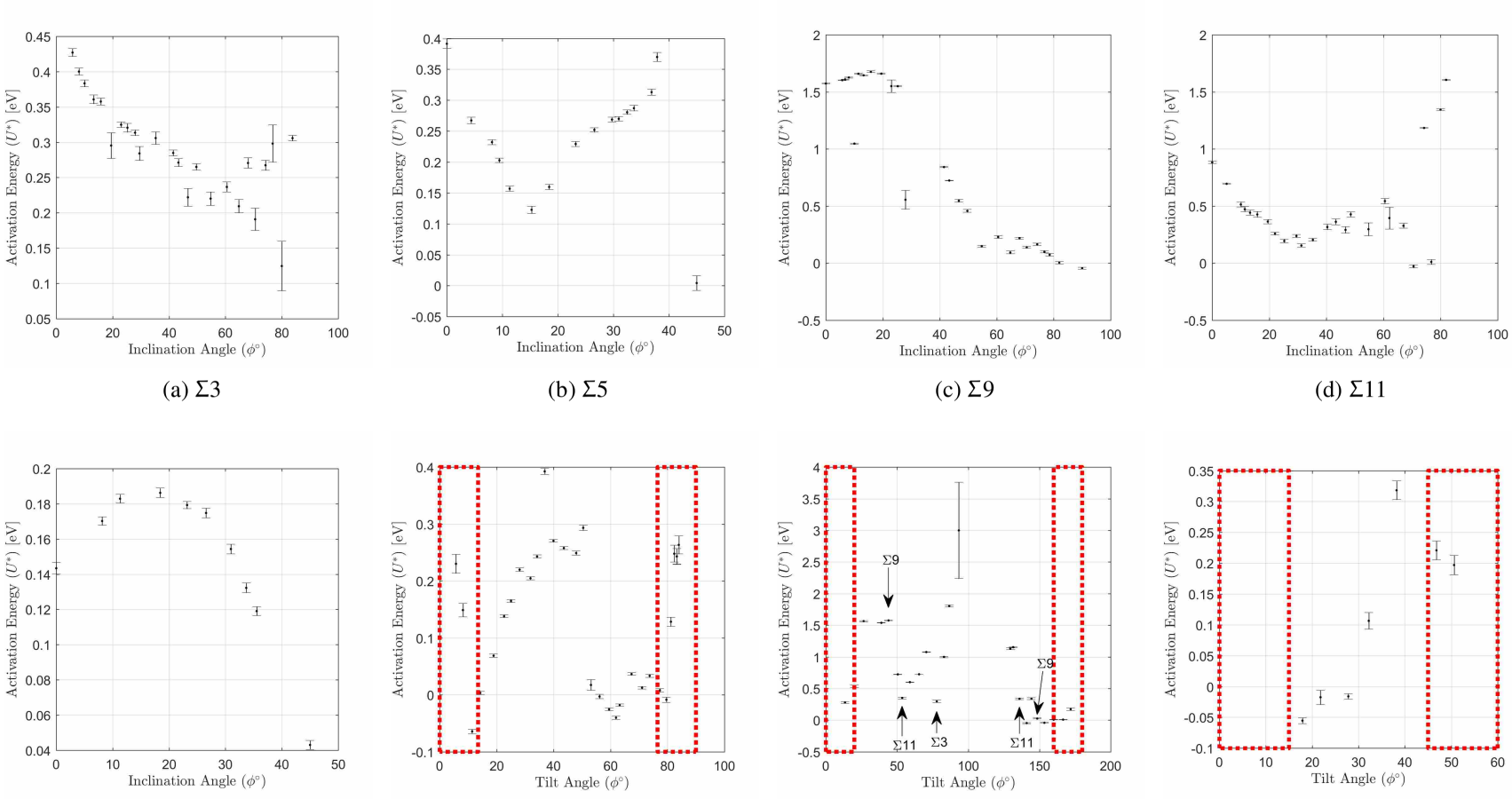
The thermodynamic underpinnings suggest that the activation energy is the energy required for a kinetic process to occur. Thus a high activation energy implies a high energy barrier that must be overcome for GB motion to occur. Similarly, low activation energies imply a small barrier that must be overcome.

The activation energies U^* as a function of GBC is provided in Fig 5.2. Most GBCs exhibit weak to strong thermally activated behavior (postive U^*). Again we observe smooth trends with changes in GB character. This is particularly exhibited in the Σ3, Σ9 and Σ13a ATGBs. However, the STGBs exhibit almost no smooth trend and typically have relatively low activation energy values.

The data exhibit some evidence of thermally damped behavior as reported by Homer et. al. [67], however, the GBC where thermally damped behavior is exhibited does disagree with GBC reported by Homer. In total, there are 20 GBCs that exhibit thermally damped behavior (i.e. negative U^*).

It is interesting to note that many of the GBCs that exhibit thermally damped behavior are poorly suited to the Arrhenius model. Of the 20 GBCs that exhibit thermally damped behavior, only 9 have a R^2 greater than 0.6. This behavior may be poorly suited for the kinetic model proposed in Eq. 3.12.

However, all of the data suggesting thermally damped behavior are not strongly anti-thermal. The GBCs that are characterized by the model as thermally damped have a low activation energy. Consequently we may regard these GBCs as nearly athermal (having no temperature response). The end result being that these data suggest that thermally activated



(a) Σ3 (b) Σ5 (c) Σ9 (d) Σ11

(e) Σ13a (f) STGB ⟨100⟩ (g) STGB ⟨110⟩ (h) STGB ⟨111⟩

Figure 5.2: Activation energy versues the GB tilt or inclination angle for the respective ATGB or STGB noted.

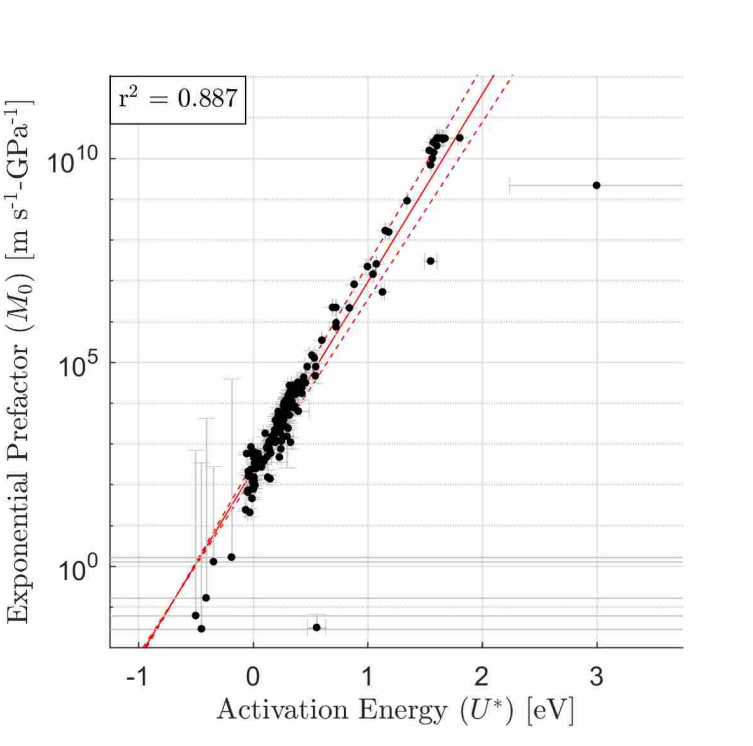


Figure 5.3: Correlation between the exponential prefactor M_0 and the activation energy U^* for all GBCs analyzed. R^2 and prediction bounds are displayed on the fit between the activation parameters.

behavior is far better suited for the kinetic model. Athermal and antithermal behavior may need to be modeled in some other way.

Trends observed in Fig. 5.2 are similar to those observed in Fig. 5.1. Figure 5.3 provides a scatterplot of M_0 and U^*. We observe a strong linear relationship between these two parameters. The linear correlation nearly crosses the origin.

The fact that M_0 has thermodynamic underpinnings in the entropy of the system, might justify a linear correlation between the activation energy and the activation entropy. This type of dependence may be mechanistically instructive. As the available number of configuration states (entropy) for mobility increases, the energy barriers to kinetic behavior decrease.

5.3 Trends in GBM Kinetics - Activation Volume V^*

The activation volume V^* associated with changes in hydrostatic stress (pressure) reveals sensitivity in mobility to applied pressure. A high, positive V^* suggests strong pressure-dampening behavior; whereas a low V^* suggests little to no effect of hydrostatic pressure on GBM. A negative V^* implies pressure activated GBM.

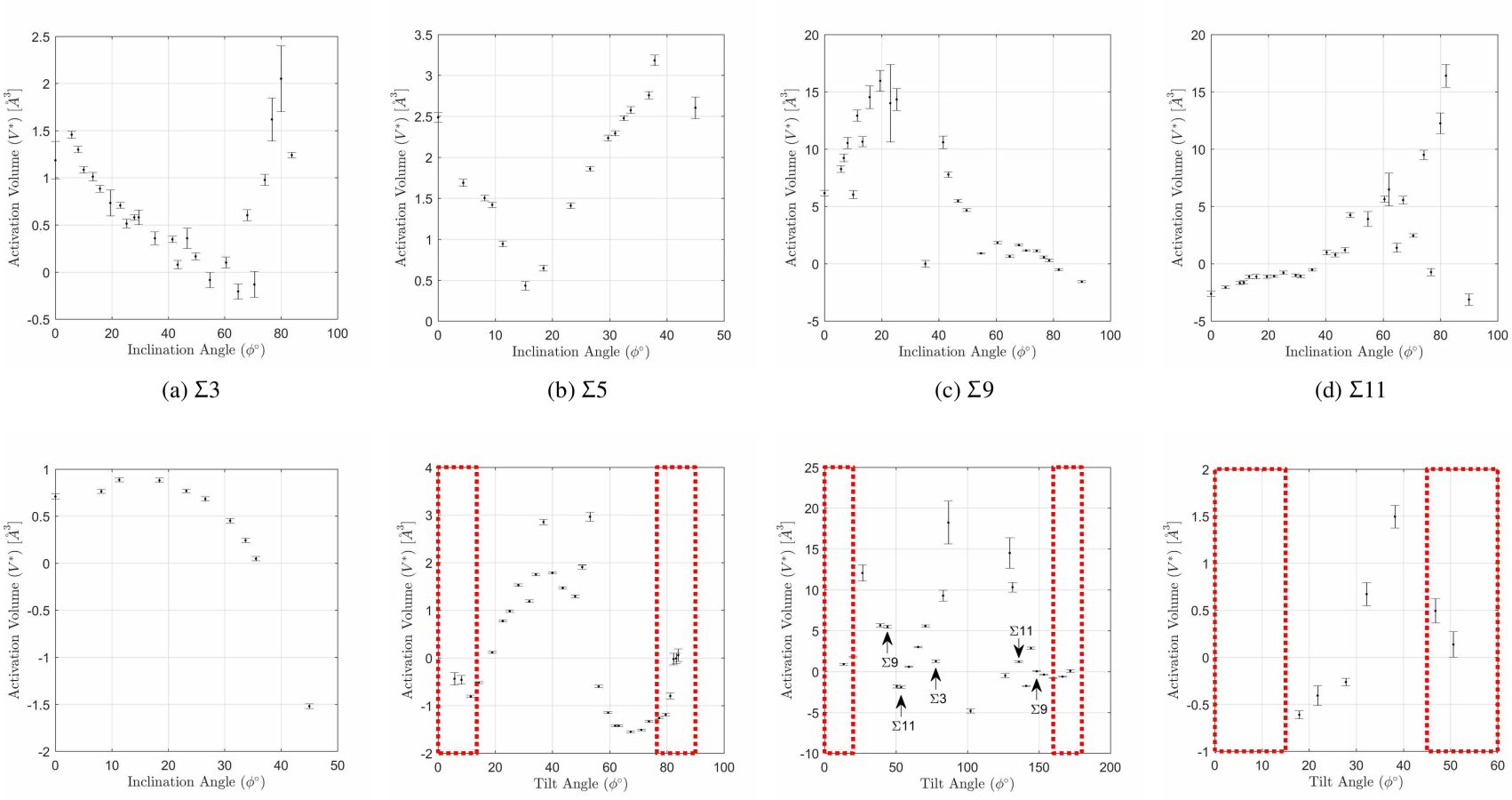
Analysis of the thermodynamics and corresponding kinetics for GBM suggest that activation volume is the required change in volume for GB motion to occur. A positive activation volume having meaning that volume must increase (temporarily in the activated state) in order for the process to evolve. Molodov et. al. have suggested that activation volume for GB mobility is nearly the same as the activation volume for bulk self-diffusion [27]. A negative activation volume suggests that the volume change required is compressive and densifying and that densifying mechanisms assist in the evolution of the kinetic process.

The activation volumes V^* associated with hydrostatic pressure is provided in Fig. 5.4. Surprisingly, there are again some clear, smooth trends in activation volume with changes in inclination angle and tilt angle. This suggests that there may be structure-property relationships that could assist in determining the activation volume for other GBs not studied here.

We report that there exists a sensitivity of GB mobility to hydrostatic pressure in most GBs tested. In most, but interestingly not all cases, hydrostatic compressive pressure suppresses GBM. These exceptions appear to be pressure activated (i.e. negative V^*). There is some evidence in the literature [41] of nonintuitive GB motion behavior (thermally damped). Here we observe a different kind of nonintuitive GB motion behavior: pressure activated behavior. Those GBCs that exhibit a negative activation volume V^* (pressure activated) tend to be fairly low in magnitude (having low sensitivity). These pressure activated GBCs need further investigation to determine root causes for the activation.

In figure 5.4 we note some special GBs within the STGB $\langle 100 \rangle$ and $\langle 110 \rangle$ classes. Two $\Sigma 5$ GBs noted within the STGB $\langle 100 \rangle$ grouping are surpisingly inconsistent with the remainder of the dataset, however, there appears to be a more coherent trend within the $\Sigma 5$ dataset. There may be trends in activation volume throughout the GB fundamental zone [68].

Careful analysis of Fig. 5.4 will reveal that several of the trends are similar to those observed in Figs. 5.1 and 5.2. Surprisingly, we see very similar trends between the activation



(a) Σ3 (b) Σ5 (c) Σ9 (d) Σ11

(e) Σ13a (f) STGB ⟨100⟩ (g) STGB ⟨110⟩ (h) STGB ⟨111⟩

Figure 5.4: Activation volume versues the GB tilt or inclination angle for the respective ATGB or STGB noted.

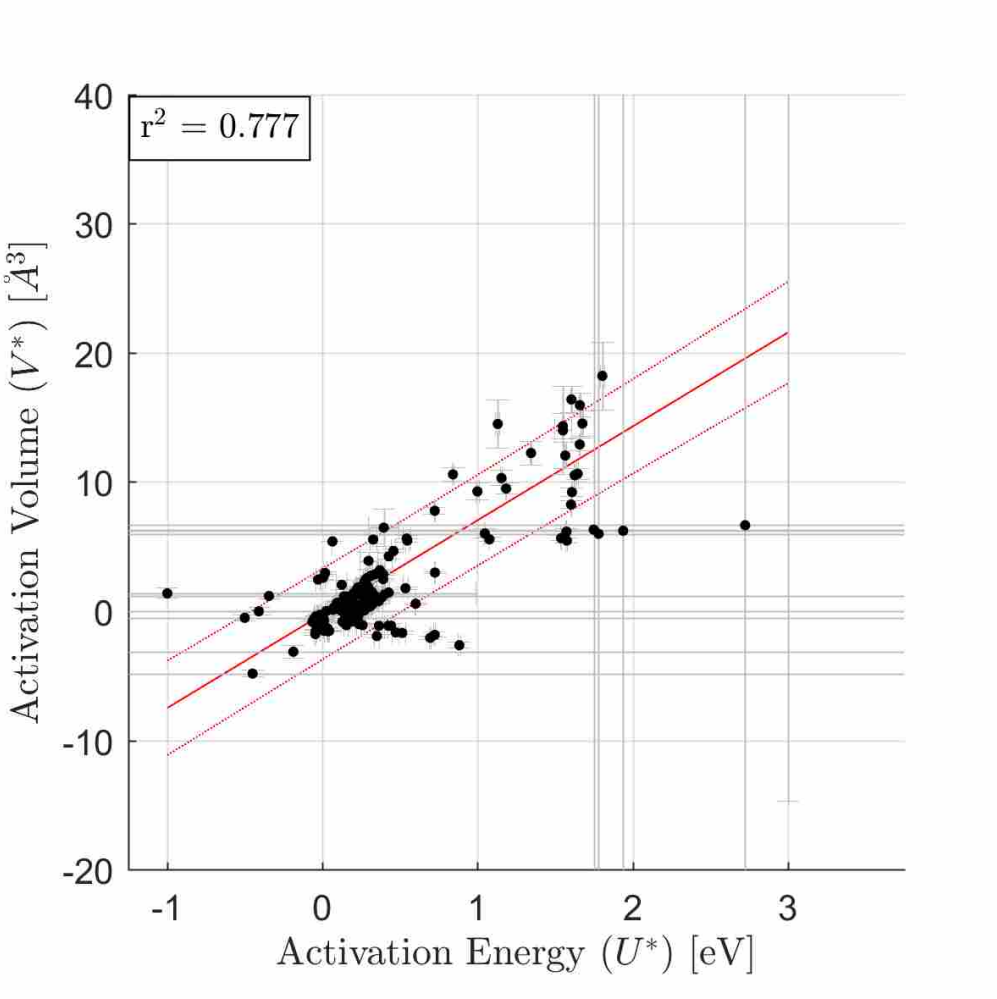


Figure 5.5: Correlation between the activation energy U^* and the activation volume V^*. Coefficient of determination and prediction bounds are displayed.

energy and the activation volume. Figure 5.5 shows the correlation between the activation energy U^* and the activation volume V^*.

This finding suggest that in some cases the behavior of those GBCs that exhibit thermally damped behavior (negative U^*) may also be activated by a densifying mechanism. This connection suggest that study of thermally damped behavior could also employ application of hydrostatic compressive stress to study thermally damped GBM. Some datapoints in Fig. 5.5 disagree with this trend. Whatever the case may be, this relationship is an interesting observation that requires further study.

The link between these model parameters suggest that mechanisms for GBM for thermally activated or damped behavior are similar to (or are the same as) those mechanisms that correspond to changes in activation volume. There exists a thermodynamic reasoning for the correlation observed in figure 5.5. As the energy required for the kinetic process increases, a volume change due to thermal expansion is naturally expected. This volume change enhances the capability for bulk-self-diffusion. Applied mechanical pressure suppresses this volume strain, and likewise reduces the mobility. The factors of temperature and pressure are both isotropic measures. It is therefore a natural consequence that these two factors are not orthogonal to each other and that the effects of temperature and hydrostatic pressure are definitively linked. The effect of increased compressive pressure directly reverses the thermodynamic effect of increased temperature.

This is intuitive when one considers that temperature is merely a measure of atomic kinetic energy. Pressure, on the other hand is a measure of atomic potential energy. As kinetic energy increases, potential energy decreases.

A key finding from this data suggests that pressure does influence GB mobility (M). Prior to this work, experimental data existed to suggest that overall GB motion was influenced, however, experimental studies rely on curvature driven GB motion to make measurements of the mobility. No data previously existed to inform the community whether pressure damped GB motion was damped due to pressure-induced changes in (a) GB curvature, (b) GB energy or (c) GB mobility or combinations of the three. While it does not rule out additional effects, this data clearly suggests that mobility does exhibit an Arrhenius relationship with changes in pressure.

5.4 Trends in GBM Kinetics - Activation Volume Deviator Tensor Λ^*_{ij}

The activation deviator volume components Λ^*_{ij} associated with changes in the stress deviator tensor express sensitivity in mobility to applied shear stress $\tilde{\sigma}_{ij}$. Considering that GBs do not have cubic symmetry, great care must be taken to express deviator tensor terms appropriately [68].

Sign indicates direction of sensitivity. A negative Λ^*_{ij} corresponds to suppression of GBM when $\tilde{\sigma}_{ij}$ is acting in the positive direction and activation of GBM when $\tilde{\sigma}_{ij}$ is negative. This effect is due to the anisotropy of the GB. In the case of a negative activation shear strain, a negative stress

may have a mobility activating effect. There is no predictive mechanism in this model for shear dampening effects, however, data produced from the MD simulations supports this behavior. A high valued (positive or negative) Λ^*_{ij} corresponds to high activation of mobility due to applied $\tilde{\sigma}_{ij}$ stress in one direction or another.

The direct connection to the mechanisms for GBM are less intuitive for Λ^*_{ij}. One way to consider the meaning of Λ^*_{ij} is the associated volume change due to application of shear stress. As shear stress is applied, a dilation of the disordered region of the GB may occur. This dilation may permit greater self-diffusion through the GB. This disagrees with the traditional continuum level perspective stress and strain deviator tensors being purely volume conserving, however, we seek to find connections to mechanisms. A dilation due to shear strain may enhance the capability for self-diffusion and is consistent with the units of the activation parameter.

An alternative way to interpret Λ^*_{ij} is using the form derived earlier in Eq. 3.8 where $\Lambda_{ij} = \tilde{\varepsilon}_{ij} V'$. In this form, the volume is not necessarily dilating, but there is a shear strain acting in the system[1]. This shear strain $\tilde{\varepsilon}_{ij}$ may indicate a lateral displacement for a unit displacement in the direction of GB motion. This lateral displacement rate is known throughout the literature as shear coupling factor and typically is denoted as

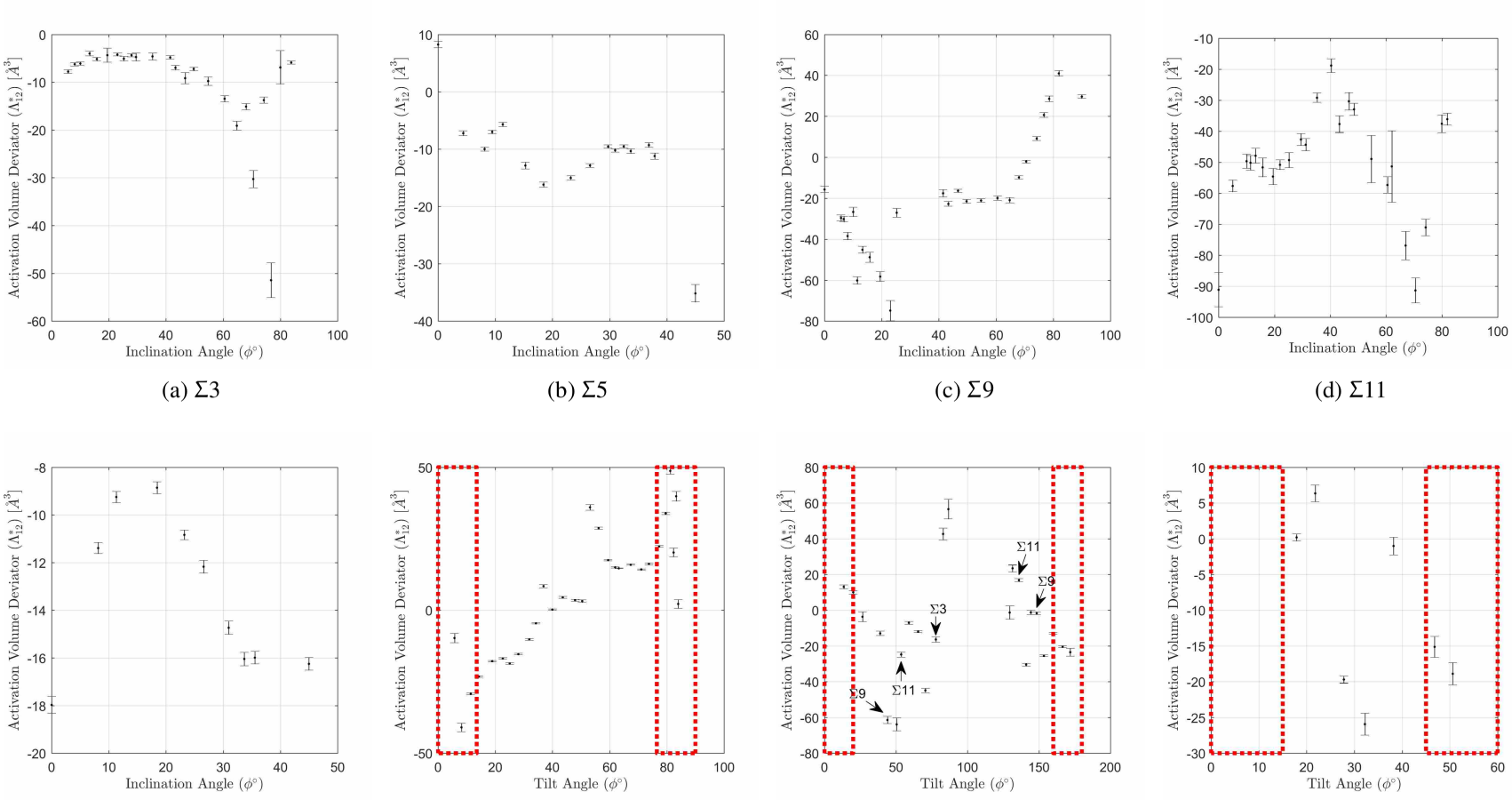
$$\beta = \frac{v_{\parallel}}{v_n} \tag{5.2}$$

where $v_{\parallel}$ is the velocity of the GB parallel to the GB normal velocity v_n [48].

The activation volume deviator components Λ^*_{12} and Λ^*_{23} are provided in figures 5.6 and 5.7 respectively. The activation volume deviator components (Λ^*_{12} and Λ^*_{23}) demonstrate some sensitivity to shear stress.

Trends with respect to the inclination or tilt angle exist for Λ^*_{12}; however, little meaningful data can be extracted from trends with Λ^*_{23}. Interestingly, the axis of the GB tilt or inclination angle is parallel the z-axis in all simulations studied. Thus strains associated with the $\tilde{\sigma}_{23}$ shear stress act parallel with the GB inclination/tilt axis. In most cases, the low values of Λ^*_{23} in comparison to those of Λ^*_{12} demonstrate relatively little sensitivity to stress in the corresponding $\tilde{\sigma}_{23}$ direction. This suggests that mechanisms for GBM do not rely on coordinated sliding motions between the

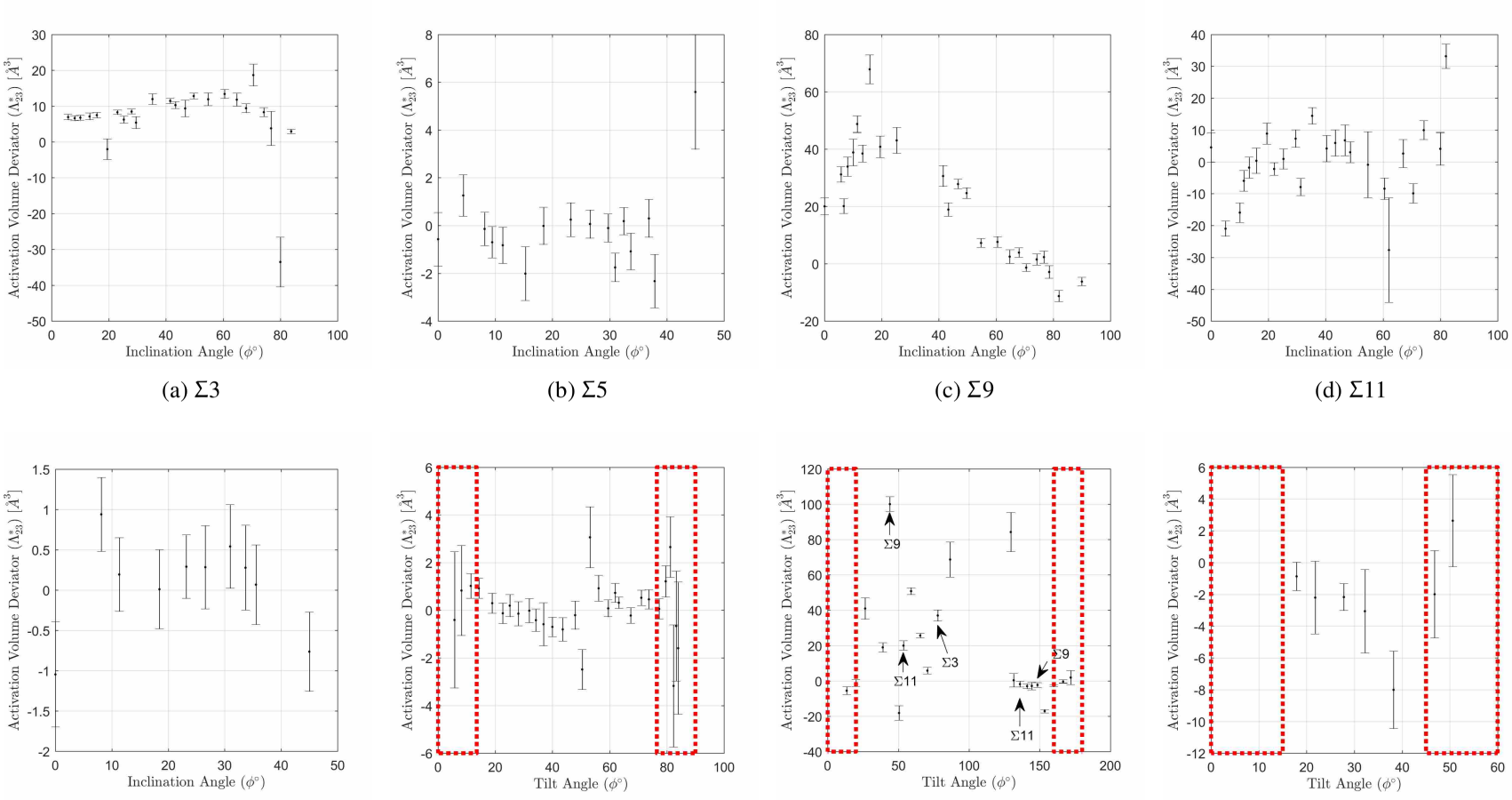
[1]The instantaneous volume V' is also without definition and is likely linked to the volume of the GB unit cell or crystal unit cell. As a consequence of this lack of definition, we cannot explicitly calculate $\tilde{\varepsilon}_{ij}$.



(a) Σ3 (b) Σ5 (c) Σ9 (d) Σ11

(e) Σ13a (f) STGB ⟨100⟩ (g) STGB ⟨110⟩ (h) STGB ⟨111⟩

Figure 5.6: Activation volume deviator tensor components Λ^*_{12} versues the GB tilt or inclination angle for the respective ATGB or STGB noted.



(a) Σ3 (b) Σ5 (c) Σ9 (d) Σ11

(e) Σ13a (f) STGB ⟨100⟩ (g) STGB ⟨110⟩ (h) STGB ⟨111⟩

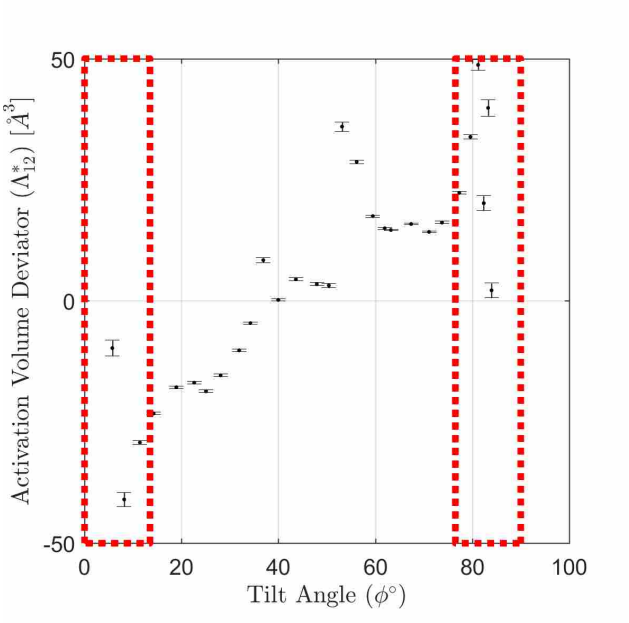
Figure 5.7: Activation volume deviator tensor components Λ^*_{23} versues the GB tilt or inclination angle for the respective ATGB or STGB noted.

two grains in the direction parallel to the tilt axis. On the other hand, shearing between grains does influence GBM when the shear direction is perpendicular to the tilt/inclination axis.

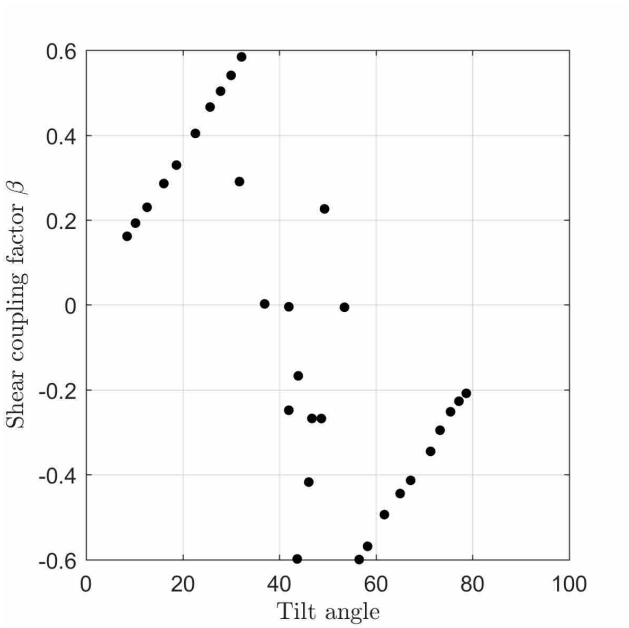
The reported values in Λ^*_{12} appear to be inversely correlated with data obtained by Homer et. al [48]. This relationship may be explained by preferred shear coupling directions reported by Homer et. al. The Λ_{12} for STGB $\langle 100 \rangle$ are directly linked with results produced by Homer. The change in direction noted by Homer is similar to the change in sign in Λ^*_{12}. Figure 5.8 shows the relationship between the shear coupling factor reported by Homer [48] and the present work.

Further verification is needed, however, it appears that the parameters Λ^*_{ij} are mechanistically linked to shear coupling of GB motion. This finding implies that study of the effect of shear states on GB mobility may be instructive for determining the multi-dimensional shear coupling factors.

Each of the Λ^*_{ij} terms appear to be entirely independent of one another and do not correlate with any other model parameter studied to this point. The volume deviator components do not have a corresponding position in an enthalpic measure. We can then determine from this lack of correlation that those terms M_0, U^* and V^* are all related through the enthalpy of the system.



(a)



(b)

Figure 5.8: Comparison in STGB $\langle 100 \rangle$ GBs showing similarities in (a) Λ_{12}^* in the present work and (b) β from Homer et. al. [48]. Shear coupling factors and the activation volume deviator have nearly an inverse relationship to one another.

CHAPTER 6. CONCLUSION

6.1 Advantages & Disadvantages of Predictive Model

The kinetic model performs exceptionally well in spite of having no considerations for GB symmetries. A high fidelity prediction confidence $R^2 = 0.936$ can be achieved using a large dataset. The model is capable of predicting acceptably well if fewer calculations were performed.

The predictive model extracts mechanistically instructive parameters to utilize in the prediction. Behavior of GBs in a wide variety of cases were considered and the model predicted well in most cases. The data produced from the predictive model provides compelling evidence of connections between activation parameters. There also exists persuasive evidence that modulation of shear stress can be directly linked to those data produced by other authors regarding shear coupling factors.

The applicability of the predictive model is of most importance. The model may appropriately be used to predict behavior of GBCs under a variety of thermoelastic states. The fact that this model has high fidelity predictive performance and connections to mechanisms for GBM implies that many of the critical behaviors to GBM can be modeled in a similar fashion.

In spite of these successes, this work does highlight some gaps in the predictive model. By utilizing the MD computational technique, some computational artifacts exist. For example, some GBCs either (a) produced insufficient mobility trends (b) the randomness in the calculated response was too great (c) no mobility was observed or (d) an irregular movement of the GB occurred. In these cases, the model attempts to extract meaningful trends, however these abnormalities prevent a high fidelity prediction. These exceptions to high predictability are characterized by a low coefficient of determination (R^2). As is shown in Fig. 4.4, within certain GB classes that behave well on the whole, some tilt/inclination angles create poorly fitting results.

Additionally, the methods employed in the present work depend on calculated GB velocity. With highly mobile GBs (mobility approaching the sensitivity of the procedure

employed 1025 m/(s-GPa)), a large amount of variance in the individual calculations may be expected. This insensitivity to highly mobile GBs may also explain some of the heteroskedacity of the predicted vs calculated mobility trends.

6.2 Summary

This thesis developed the following

- A methodology for calculating GBM in a variety of thermoelastic states (Temperature, pressure and shear stress)
- A computational procedure optimizing computational methods to obtain a large GBM dataset
- A computational method for calculation of GBM in fully periodic simulation cells
- A thermoelastic basis for use in a kinetic model

In addition to those items above, this thesis developed a kinetic model that predicts changes in GB mobility with varied thermoelastic states (Temperature, Pressure and Shear Stress). 98,786 Molecular Dynamics simulations were performed in a thermodynamic ensemble appropriate to the energy formulation of the kinetic model to validate the form of the model and obtain model parameters. This kinetic model has the features of

- having thermodyamic foundations
- possessing meaningful parameters
- fitting data exceptionally well with overall coefficient of determination $R^2 = 0.936$
- predicting GBM in a wide variety of GB characters
- provide insight to mechanistically instructive behaviors for GBM (i.e. shear coupling)

6.3 Future Work

The results of this work suggest the need for further MD studies of triaxial stress-modulated grain boundary migraiton. This work only studied 2 of the 6 independent stress deviator terms. Future efforts may expand on the present work to include these terms.

Corrections to these calculations are needed to reconcile the effects of (1) errors due to the MD synthetic driving force method employed and (2) the GB velocity calculations tracking both GBs independently rather than being computed as an average velocity.

Additionally, this work demonstrates the need to develop robust methods for characterizing vector and tensor quantities relative to the GB unit cell with consideration for GB unit cell symmetries. Such methods could then be used to definitively determine if the activation strains are related to mechanisms for GB motion.

Shear-coupling between grains with applied shear stress may be another interesting consequence of this research. No effort was exerted to determine the shear coupling factor (β) under any applied stress or temperature state. The data produced from these simulations could be re-used to determine β. Similar to the GBM, β could exhibit an Arrhenius type behavior that depends on the energy potential defined in equation 3.8.

Finally, future work investigating the differences in mechanisms employed under thermoelastic conditions should be performed. This work should attempt to connect the extensive quantities to the mechanisms of the kinetic processes of GBM. A great deal of detail is required for this type of work, however, these efforts should prove fruitful for further refinement of the predictive model.

6.4 Major Findings

- A predictive kinetic model for GBM that combines hydrostatic stress and shear stress is viable and achieves high quality predictability.
- Model parameters that are thermodynamically based and mechanistically instructive were determined and most had smooth trends with respect to changes in GBC.

- Connections between the activation entropy S^*, activation energy U^*, and activation volume V^* were discovered
- Connections between activation energy U^* and activation volume V^* show that thermally damped behavior may be related to a compressive deformation mechanism in some cases.
- Activation volume deviator tensor components Λ^*_{ij} are correlated to behavior observed in shear coupling studies.
- GB mobility does follow an Arrhenius behavior with applied hydrostatic pressure generally suppressing the GB motion; however, there is some evidence for the possibility of pressure activated GB motion for some GB characters.

6.5 Funding

This work was supported by US Synthetic Corporation.

REFERENCES

[1] Hall, "The deformation and ageing of mild steel III Discussion of results\t," *Proceedings of the Physical Society. Section B*, vol. 64, no. 9, p. 747, 1951. 1

[2] N. Petch, "The cleavage strength of polycrystals," *J. Iron Steel Inst*, vol. 174, no. 19, pp. 25 – 28, 1953. 1

[3] Z. C. Cordero, B. E. Knight, and C. A. Schuh, "Six decades of the Hall-Petch effect - a survey of grain-size strengthening studies on pure metals," *International Materials Reviews*, vol. 6608, no. July, pp. 1–18, 2016. 1

[4] M. Shimada, H. Kokawa, Z. Wang, Y. Sato, and I. Karibe, "Optimization of grain boundary character distribution for intergranular corrosion resistant 304 stainless steel by twin-induced grain boundary engineering," *Acta Materialia*, vol. 50, pp. 2331–2341, may 2002. 1

[5] L. Lu, Y. Shen, X. Chen, L. Qian, and K. Lu, "Ultrahigh Strength and High Electrical Conductivity in Copper," *Science*, vol. 304, no. 5669, pp. 422–426, 2004. 1

[6] M. A. Meyers, A. Mishra, and D. J. Benson, "Mechanical properties of nanocrystalline materials," *Progress in Materials Science*, vol. 51, no. 4, pp. 427–556, 2006. 1

[7] P. Lin, G. Palumbo, U. Erb, and K. Aust, "Influence of grain boundary character distribution on sensitization and intergranular corrosion of alloy 600," *Scripta Metallurgica et Materialia*, vol. 33, pp. 1387–1392, nov 1995. 1

[8] E. Lehockey and G. Palumbo, "On the creep behaviour of grain boundary engineered nickel 1," *Materials Science and Engineering: A*, vol. 237, no. 2, pp. 168–172, 1997. 1

[9] E. M. Lehockey, G. Palumbo, and P. Lin, "Improving the weldability and service performance of nickel-and iron-based superalloys by grain boundary engineering," *Metallurgical and Materials Transactions A*, vol. 29, pp. 3069–3079, dec 1998. 1

[10] E. M. Lehockey, G. Palumbo, a. Brennenstuhl, and P. Lin, "Mitigating intergranular attack and growth in lead-acid battery electrodes for extended cycle and operating life," *Metallurgical and Materials Transactions A*, vol. 29, no. 1, pp. 387–396, 1998. 1

[11] E. M. Lehockey, D. Limoges, G. Palumbo, J. Sklarchuk, K. Tomantschger, and A. Vincze, "On improving the corrosion and growth resistance of positive Pb-acid battery grids by grain boundary engineering," *Journal of Power Sources*, vol. 78, pp. 79–83, mar 1999. 1

[12] D. Bainbridge, H. Choh, and E. Edwards, "Recent observations on the motion of small angle dislocation boundaries," *Acta metallurgica*, vol. 2, no. 2, pp. 322–333, 1954. 1, 2

[13] C. H. Li, E. Edwards, J. Washburn, and E. Parker, "Stress-induced movement of crystal boundaries," *Acta Metallurgica*, vol. 1, no. 2, pp. 223–229, 1953. 1, 2

[14] L. Wan, W. Han, and K. Chen, "Bi-crystallographic lattice structure directs grain boundary motion under shear stress," *Scientific Reports*, vol. 5, p. 13441, 2015. 1, 2

[15] T. Gorkaya and D. Molodov, "Stress-driven migration of symmetrical h 1 0 0 i tilt grain boundaries in Al bicrystals," vol. 57, pp. 5396–5405, 2009. 1, 2

[16] D. Molodov and T. Gorkaya, "Migration of the R 7 tilt grain boundary in Al under an applied external stress," vol. 65, pp. 990–993, 2011. 1, 2

[17] M. Legros, D. S. Gianola, and K. J. Hemker, "In situ TEM observations of fast grain-boundary motion in stressed nanocrystalline aluminum films," vol. 56, pp. 3380–3393, 2008. 1, 2

[18] D. S. Gianola, D. H. Warner, J. F. Molinari, and K. J. Hemker, "Increased strain rate sensitivity due to stress-coupled grain growth in nanocrystalline Al," *Scripta Materialia*, vol. 55, no. 7, pp. 649–652, 2006. 1, 2

[19] J. W. Cahn and J. E. Taylor, "A unified approach to motion of grain boundaries, relative tangential translation along grain boundaries, and grain rotation," *Acta Materialia*, vol. 52, pp. 4887–4898, sep 2004. 1, 2, 18

[20] D. Pan, S. Kuwano, T. Fujita, and M. W. Chen, "Ultra-large room-temperature compressive plasticity of a nanocrystalline metal," *Nano Letters*, vol. 7, no. 7, pp. 2108–2111, 2007. 1, 2

[21] M. Jin, A. M. Minor, E. A. Stach, and J. W. Morris, "Direct observation of deformation-induced grain growth during the nanoindentation of ultrafine-grained Al at room temperature," *Acta Materialia*, vol. 52, no. 18, pp. 5381–5387, 2004. 1, 2

[22] K. Zhang, J. R. Weertman, and J. A. Eastman, "Rapid stress-driven grain coarsening in nanocrystalline Cu at ambient and cryogenic temperatures," *Applied Physics Letters*, vol. 87, no. 6, pp. 85–88, 2005. 1, 2

[23] G. J. Fan, L. F. Fu, H. Choo, P. K. Liaw, and N. D. Browning, "Uniaxial tensile plastic deformation and grain growth of bulk nanocrystalline alloys," *Acta Materialia*, vol. 54, no. 18, pp. 4781–4792, 2006. 1, 2

[24] M. Jin, A. M. Minor, and J. W. Morris, "Strain-induced coarsening in nano-grained films," *Thin Solid Films*, vol. 515, no. 6, pp. 3202–3207, 2007. 1, 2

[25] D. S. Gianola, C. Eben, X. Cheng, and K. J. Hemker, "Stress-driven surface topography evolution in nanocrystalline al thin films," *Advanced Materials*, vol. 20, no. 2, pp. 303–308, 2008. 1, 2

[26] S. Brandstetter, K. Zhang, A. Escuadro, J. R. Weertman, and H. Van Swygenhoven, "Grain coarsening during compression of bulk nanocrystalline nickel and copper," *Scripta Materialia*, vol. 58, no. 1, pp. 61–64, 2008. 1, 2

[27] D. Molodov, J. Swiderski, G. Gottstein, W. Lojkowski, and L. Shvindlerman, "Effect of pressure on grain boundary migration in aluminium bicrystals," *Acta Metallurgica et Materialia*, vol. 42, pp. 3397–3407, oct 1994. 1, 2, 17, 41

[28] H. Hahn, H. Gleiter, "The Effect of Pressure on Grain Growth and Boundary Mobility," *Scripta Metallurgica*, vol. 13, no. c, pp. 3–6, 1979. 1, 2, 17

[29] V. Sursaeva, S. Protasova, W. Lojkowski, and J. Jun, "Microstructure evolution during normal grain growth under high pressure in 2-D Al Foils," *Texture and Microstructures*, vol. 32, no. 1-4, pp. 175–185, 1999. 1, 2

[30] L. Shvindlerman, U. Czubayko, G. Gottstein, and D. Molodov, "High pressure effect on grain boundary migration and mechanism of grain boundary migration," *Materials Science Forum*, vol. 204-206, pp. 45–54, 1996. 1, 2

[31] D. Molodov, B. Straumal, and L. Shvindlerman, "The effect of pressure on migration of 001 tilt grain boundaries in tin bicrystals," *Scripta Metallurgica*, vol. 18, pp. 207–211, mar 1984. 1, 2

[32] O. Hunderi and N. Ryum, "Computer Simulation of Grain Growth," *Acta Metallurgicaa*, vol. 27, pp. 161–165, 1978. 1, 2

[33] E. A. Lazar, R. D. MacPherson, and D. J. Srolovitz, "A more accurate two-dimensional grain growth algorithm," *Acta Materialia*, vol. 58, pp. 364–372, jan 2010. 1, 2

[34] E. A. Lazar, J. K. Mason, R. D. MacPherson, and D. J. Srolovitz, "A more accurate three-dimensional grain growth algorithm," *Acta Materialia*, vol. 59, pp. 6837–6847, oct 2011. 1, 2

[35] J. K. Mason, E. A. Lazar, R. D. MacPherson, and D. J. Srolovitz, "Geometric and topological properties of the canonical grain-growth microstructure," *Physical Review E*, vol. 92, p. 063308, dec 2015. 1, 2

[36] C. Mieben, M. Liesenjohann, L. Barrales-Mora, L. Shvindlerman, and G. Gottstein, "An advanced level set approach to grain growth - Accounting for grain boundary anisotropy and finite triple junction mobility," *Acta Materialia*, vol. 99, pp. 39–48, oct 2015. 1, 2

[37] D. Molodov and T. Gorkaya, "Dynamics of grain boundaries under applied mechanical stress," pp. 4318–4326, 2011. 1, 2

[38] R. W. Balluffi, S. M. Allen, and W. C. Carter, *Kinetics of Materials*. Hoboken, NJ, USA: John Wiley & Sons, Inc., sep 2005. 2, 20

[39] M. Upmanyu, D. J. Srolovitz, L. S. Shvindlerman, and G. Gottstein, "Misorientation dependence of intrinsic grain boundary mobility: simulation and experiment," *Acta Materialia*, vol. 47, no. 14, pp. 3901–3914, 1999. 2

[40] D. L. Olmsted, E. A. Holm, and S. M. Foiles, "Survey of computed grain boundary properties in face-centered cubic metals-II: Grain boundary mobility," *Acta Materialia*, vol. 57, pp. 3704–3713, aug 2009. 2

[41] E. R. Homer, E. A. Holm, S. M. Foiles, and D. L. Olmsted, "Trends in Grain Boundary Mobility : Survey of Motion Mechanisms," vol. 66, no. 1, pp. 114–120, 2014. 2, 41

[42] D. Molodov, V. Ivanov, and G. Gottstein, "Low angle tilt boundary migration coupled to shear deformation," *Acta Materialia*, vol. 55, pp. 1843–1848, mar 2007. 2

[43] D. S. Gianola, S. Van Petegem, M. Legros, S. Brandstetter, H. Van Swygenhoven, and K. J. Hemker, "Stress-assisted discontinuous grain growth and its effect on the deformation behavior of nanocrystalline aluminum thin films," *Acta Materialia*, vol. 54, no. 8, pp. 2253–2263, 2006. 2

[44] T. J. Rupert, D. S. Gianola, Y. Gan, and K. J. Hemker, "Experimental observations of stress-driven grain boundary migration.," *Science (New York, N.Y.)*, vol. 326, no. December, pp. 1686–1690, 2009. 2

[45] M. Shiga and W. Shinoda, "Stress-assisted grain boundary sliding and migration at finite temperature: A molecular dynamics study," vol. 054102, pp. 6–9, 2004. 2

[46] H. Zhang, D. Du, and D. J. Srolovitz, "Effects of boundary inclination and boundary type on shear-driven grain boundary migration," vol. 6435, no. March, 2017. 2

[47] A. Rajabzadeh, F. Mompiou, M. Legros, and N. Combe, "Elementary Mechanisms of Shear-Coupled Grain Boundary Migration," vol. 265507, no. June, pp. 1–5, 2013. 2

[48] E. R. Homer, S. M. Foiles, E. A. Holm, and D. L. Olmsted, "Phenomenology of shear-coupled grain boundary motion in symmetric tilt and general grain boundaries," *Acta Materialia*, vol. 61, pp. 1048–1060, feb 2013. 2, 18, 45, 48, 49

[49] Y. Mishin, A. Suzuki, B. P. Uberuaga, and A. F. Voter, "Stick-slip behavior of grain boundaries studied by accelerated molecular dynamics," *Physical Review B*, vol. 75, p. 224101, jun 2007. 2, 18

[50] J. W. Cahn, Y. Mishin, and A. Suzuki, "Coupling grain boundary motion to shear deformation," *Acta Materialia*, vol. 54, pp. 4953–4975, nov 2006. 2, 18

[51] S. Plimpton, "Fast Parallel Algorithms for Short-Range Molecular Dynamics," *Journal of Computational Physics*, vol. 117, pp. 1–19, mar 1995. 4, 6

[52] Y. Mishin, D. Farkas, M. J. Mehl, and D. A. Papaconstantopoulos, "Interatomic potentials for monoatomic metals from experimental data and ab initio calculations," *Physical Review B - Condensed Matter and Materials Physics*, vol. 59, no. 5, pp. 3393–3407, 1999. 4, 5, 7

[53] M. S. Daw and M. I. Baskes, "Embedded-atom method: Derivation and application to impurities, surfaces, and other defects in metals," *Physical Review B*, vol. 29, pp. 6443–6453, jun 1984. 4

[54] M. A. Tschopp, S. P. Coleman, and D. L. McDowell, "Symmetric and asymmetric tilt grain boundary structure and energy in Cu and Al (and transferability to other fcc metals)," *Integrating Materials and Manufacturing Innovation*, vol. 4, no. 1, p. 11, 2015. 7

[55] A. Stukowski, "Visualization and analysis of atomistic simulation data with OVITO-the Open Visualization Tool," *Modelling and Simulation in Materials Science and Engineering*, vol. 18, no. 1, 2010. 8, 11, 13

[56] K. G. F. Janssens, D. Olmsted, E. A. Holm, S. M. Foiles, S. J. Plimpton, and P. M. Derlet, "Computing the mobility of grain boundaries," *Nature Materials*, vol. 5, no. 2, pp. 124–127, 2006. 10, 11, 59

[57] D. L. Olmsted, E. A. Holm, and S. M. Foiles, "Survey of computed grain boundary properties in face-centered cubic metals—II: Grain boundary mobility," *Acta Materialia*, vol. 57, pp. 3704–3713, aug 2009. 12, 59

[58] M. Wand and M. Jones, *Kernel Smoothing*. 1994. 14

[59] V. Atanasiu, "Kernel smoothing density estimate for circular data." 14

[60] M. D. Mckay, R. J. Beckman, and W. J. Conover, "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output From a A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output From a Computer Code," *Technometrics*, vol. 41, no. 1, pp. 55–61, 2000. 17

[61] R. C. Liebermann, "Multi-anvil, high pressure apparatus: a half-century of development and progress," *High Pressure Research*, vol. 31, pp. 493–532, dec 2011. 17

[62] V. A. Ivanov and Y. Mishin, "Dynamics of grain boundary motion coupled to shear deformation: An analytical model and its verification by molecular dynamics," *Physical Review B*, vol. 78, p. 064106, aug 2008. 18

[63] W. Jenkins, T. Digges, and C. Johnson, "Tensile properties of copper, nickel, and 70-percent-copper-30-percent-nickel and 30-percent-copper-70-percent-nickel alloys at high temperatures," *Journal of Research of the National Bureau of Standards*, vol. 58, p. 201, apr 1957. 18

[64] G. A. Holzapfel, "No Title," *Meccanica*, vol. 37, no. 4/5, pp. 489–490, 2002. 19, 20

[65] H. B. Callen and R. B. Griffiths, "Thermodynamics and an Introduction to Thermostatistics," *American Journal of Physics*, vol. 55, pp. 860–861, sep 1987. 20

[66] S. L. Ball, K. C. Alexander, and C. A. Schuh, "Stress-dependence of kinetic transitions at atomistic defects," *Modelling and Simulation in Materials Science and Engineering*, vol. 26, p. 015007, jan 2018. 21

[67] E. R. Homer, S. Patala, and J. L. Priedeman, "Grain Boundary Plane Orientation Fundamental Zones and Structure- Property Relationships," *Nature Publishing Group*, pp. 1–13, 2015. 38

[68] S. Patala and C. A. Schuh, "Symmetries in the representation of grain boundary-plane distributions," vol. 6435, 2013. 41, 44

APPENDIX A. CORRIGENDUM TO MOBILITY DUE TO SYNTHETIC DRIVING FORCE ERRORS

A major flaw in the computed mobilities in this work is that of the synthetic driving force. The synthetic driving force relies on a difference in orientation order parameters to assign the synthetic free energy to the system. The order parameter ξ is detailed here as

$$\xi_i = \sum_{j=1}^{12} \left| \vec{r}_j - \vec{r}_j^I \right| \tag{A.1}$$

These terms are fully described by Janssens et. al. [56]. The assignment has a low cutoff(ξ_{low}) and high cutoff(ξ_{high}). Any order parameter above the cutoff is assigned the full synthetic driving force, any order parameter below the low cutoff limit is assigned zero driving force. An interpolation scheme as defined by Janssens is used for those order parameters between the cutoffs.

If careful assignment of the cutoffs of the order parameters is not taken, some or all of the synthetically applied energy may be assigned to the grain that should have zero free energy applied to it. This reduces the effect of the synthetically applied free energy because some free energy is applied to the energetically favored grain. This reduces the energy potential (ΔG or G^*) for the kinetic behavior.

As documented by Olmsted et. al. [57], the actual driving force applied to the cell may be computed separately and is given as

$$\Delta G = \int_0^{u_0} \frac{1}{u} \langle \Phi_u \rangle_{u_0=u} du \tag{A.2}$$

where the terms of this equation are detailed in full by Olmsted et. al. [57].

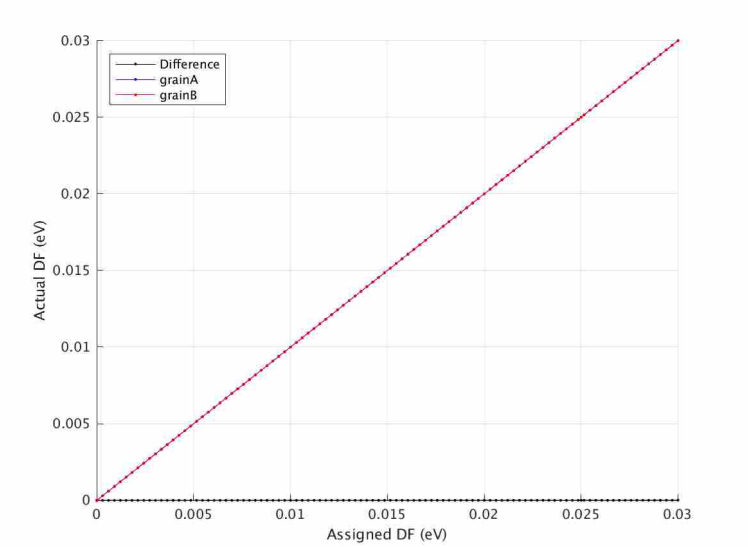


Figure A.1: Difference in synthetic potential energy due to the methods employed in the body of this work for STGB $\langle 100 \rangle$ 5.7°. These data show that grains recieve equal amounts of the applied synthetic driving force resulting in zero net difference in potential energy.

Figures A.1 through A.4 show the difference in assigned vs actual free energy differences for a specific GBC and thermodynamic state. These figures show a variety of performance characteristics with respect to changes in the free energy assignment due to the "fix orient/fcc" command in LAMMPS. It is clear from these data that special care must be taken to perform accurate calculations of GB mobility. Such care was not taken for the data in the body of this work. Particularly those GBs that exhibit low mobility are suspect because of the effects dual assignment of synthetic driving force. Additionally, special care must be taken to assign the driving force in those GBs where the orientation order parameter is very close to one another.

The error associated with these data and others is tabulated in Table A.1. These data suggest a significant error impact on the calculations of GB mobility in many simulations. Those simulations with order parameters that are very similar to one another (i.e. STGB $\langle 100 \rangle$ 5.7° and STGB $\langle 110 \rangle$ 8.1°) exhibit dual assignment of the synthetic free energy. These data will be expanded upon before reporting this work in a future publication. This error significantly

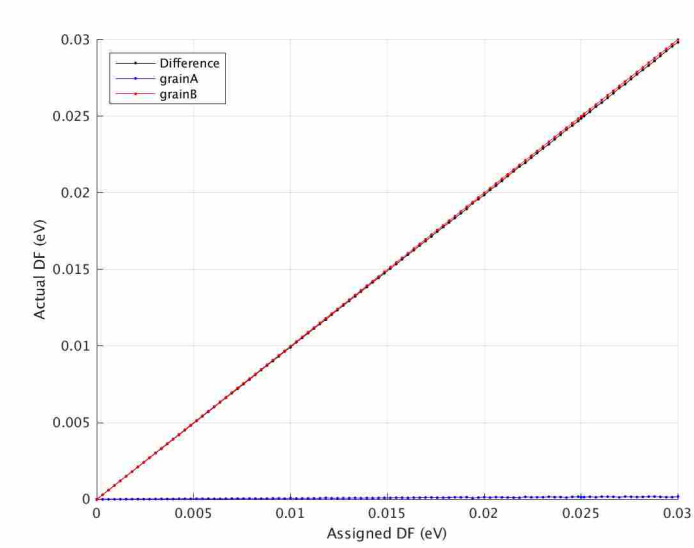


Figure A.2: Difference in synthetic potential energy due to the methods employed in the body of this work for STGB $\langle 100 \rangle$ 56.1°. Here we observe that the unfavored grain (grain A) recieves the assigned potential energy and the favored grain (grain A) receives zero additional potential energy.

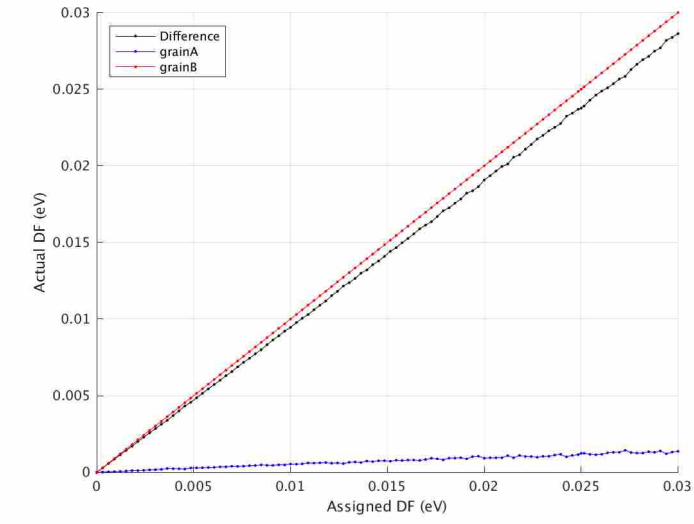


Figure A.3: Difference in synthetic potential energy due to the methods employed in the body of this work for Σ11 22.0°. Another example of partial assignment of potential energy being given to the favored grain (grain A) while the full assignment of potential energy is given to the unfavored grain (grain B).

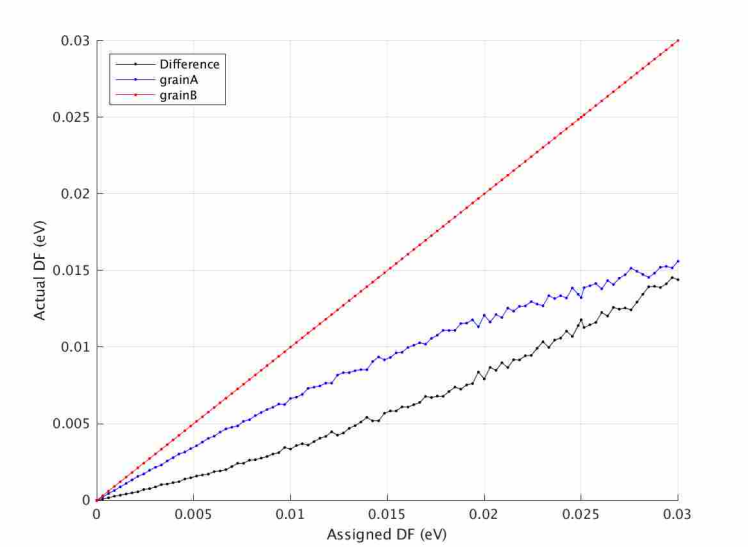


Figure A.4: Difference in synthetic potential energy due to the methods employed in the body of this work for STGB ⟨100⟩ 77.3°. This example shows that a very high amount of synthetic energy was applied to the favored grain (grain A) resulting in a net potential energy difference between the grains that is far less than the assigned driving force expected.

Table A.1: Tabulated errors of differing GBs under differing thermoelastic states

GBC	RunID	Error Mobility 1	Error Mobility 2
STGB ⟨100⟩ 5.7°	5013	-99.99%	-99.98%
STGB ⟨100⟩ 56.1°	6526	-1.24%	18.64%
STGB ⟨100⟩ 77.3°	8099	-64.50%	-55.13%
Σ11 22.0°	14311	-5.73%	13.34%
Σ9 22.9°	28102	-1.51%	18.32%
Σ9 90.0°	32494	-1.62%	18.19%
STGB ⟨110⟩ 8.1°	41865	-99.81%	-99.77%
STGB ⟨110⟩ 144.4°	46545	-1.91%	17.84%
STGB ⟨110⟩ 153.5°	46934	-11.56%	6.62%
STGB ⟨110⟩ 50.5°	49202	-4.42%	14.88%
STGB ⟨110⟩ 53.6°	49581	-5.51%	13.62%
STGB ⟨111⟩ 17.9°	49874	-21.24%	-4.71%

influences approximately 7.4% of the entire dataset as 7.4% of GBCs are low angle STGBs. The remaining 92.6% of the dataset may have induced errors due to the driving force of up to 10%.

APPENDIX B. CORRIGENDUM TO MOBILITY DUE TO GB VELOCITY CALCULATIONS

A superior method to computing the GB velocity beyond that employed in the main body of this work is discussed here. The method described in the body of this work analyzes the distance between two grains to determine the average velocity of grain boundaries as they evolve in the kinetic process. The method discussed here is one that utilizes a reference set of atoms to determine individual GB velocity. This method produces results that are very similar to those using the method discussed in the body of this work. However, there are some notable differences. These differences in velocity may assist in determining directional dependence in mobility between the two GBs for some limited case ATGBs.

The GB velocity v is simply the velocity of the GB with respect to a group of atoms that were centrally located between the two converging GBs and are located on the energetically elevated grain due to application of the synthetic driving force. This results in GBs typically converging toward the reference set of atoms. Thus, the velocity equation (in the stead of Eq. 2.4) is

$$v_i = \frac{dy_k}{dt} \tag{B.1}$$

where v_k is the velocity of the k-th GB relative to the reference set of atoms and y_k is the distance of the k-th GB to the reference set of atoms. The complete algorithm used for computing the GB velocities is provided at the end of this section.

A sample set of data was collected to compare the difference in GB velocities. If the GB velocities are very different from one another, the method discussed in the body of the text is erroneous. A comparison of these velocities for the selected sample is provided in the Table B.1. This data shows that, for the most part, relatively little differences between the GB velocities exist

Table B.1: Average ($\bar{v}$) and difference in velocity (Δv) between the two GBs for each of the two driving forces. This table shows data for a select subset of calculations under various thermodynamic states. Here $\bar{v}$ is the mean velocity of the two GBs and Δv is the difference in velocity of the two GBs.

GBC	RunID	$\bar{v}$ DF1	Δv DF1	$\bar{v}$ DF2	Δv DF2
STGB ⟨100⟩ 56.1°	6526	0.521875	0.047198	0.694445	0.0953
STGB ⟨100⟩ 77.3°	8099	0.432711	0.047993	0.572049	0.000295
Σ11 22.0°	14311	0.04457	0.072446	0.312332	0.353386
Σ9 90.0°	32494	0.497466	0.077469	0.710804	0.062139
Σ13a 23.2°	16001	0.251122	0.008411	0.325666	0.044875
Σ13a 23.2°	16002	0.185983	0.011952	0.260458	0.010470
Σ13a 31.0°	16501	0.236820	0.041571	0.3368180	0.007970
Σ13a 31.0°	16502	0.238625	0.047504	0.3648312	0.062249
STGB ⟨110⟩ 8.1°	41865	0.003174	0.006348	0.001719	0.003438
STGB ⟨110⟩ 109.5°	44350	0.008237	1.06E-05	0.001829	7.30E-06
STGB ⟨110⟩ 153.5°	46934	0.87905	0.264735	1.030537	0.037309
STGB ⟨110⟩ 50.5°	49202	0.010287	1.90E-05	0.004455	0.003923
STGB ⟨110⟩ 53.6°	49581	0.100672	0.056848	0.694054	0.381839
STGB ⟨111⟩ 17.9°	49874	3.377982	0.19943	3.685033	0.465922

when using this method of having a reference set of atoms. In general, the lower the velocity, the greater the percent error between GB velocities.

The plots in this section show GB velocities converging toward the reference set of atoms over time. The GB velocity calculations that will be reported in a future publication will make use of this superior method. What is critical to note (and made evident by Fig. B.3), is that in one case, an ATGB may be expected to have differing behavior for each GB.

This interesting behavior suggests that a separate mobility measure may be needed to appropriately characterize the motion of GBs in different directions. One mobility for motion in a given direction (positive displacement) and an entirely different mobility and kinetic behavior for motion in the opposite direction (negative displacement). From this point of view, it may be appropriate to treat the GB mobility as distinctly different GB velocities and entirely independent kinetic processes. However, this work was ill-suited to directly study this effect and more direct modeling approaches may be more useful in determining if this effect exists to a widespread degree across a variety of GBCs.

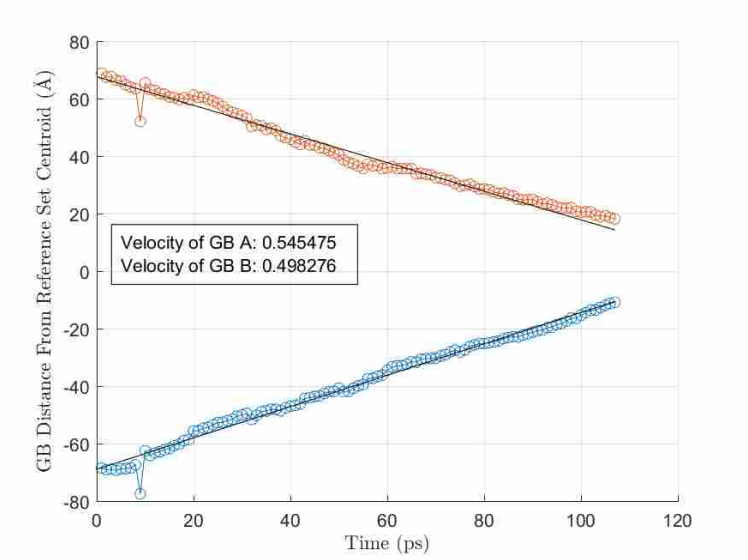


Figure B.1: Difference in GB velocity for STGB ⟨100⟩ 56.1° at a randomly selected thermodynamic state. Blue and red indicate GB positions of grain boundary "A" and "B" respectively. The two GBs approach each other and converge on the centroid of the reference set of atoms.

In those simulations interrogated using this improved method, only one GB motion simulation exhibited differing GB velocities. This simulation is shown in Fig. B.3. From this sample set, and sampling beyond what is reported in table B.1 indicate that this effect was only observed in 1 case out of 22 samples of ATGB simulations. If we were to extrapolate these results, this would indicate that less than 5% of the full dataset may have differing velocity behavior such as shown in Fig. B.3.

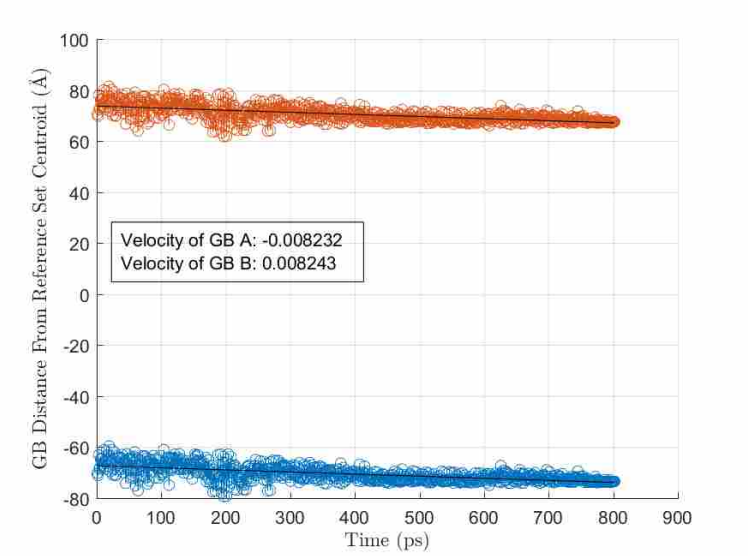


Figure B.2: Difference in GB velocity for STGB ⟨110⟩ 109.5° at a randomly selected thermodynamic state. Blue and red indicate GB positions of grain boundary "A" and "B" respectively. The two GBs approach each other and converge on the centroid of the reference set of atoms.

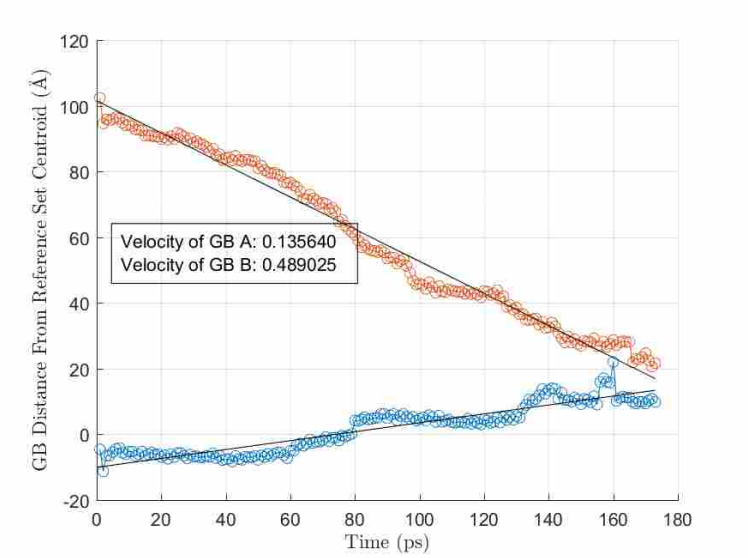


Figure B.3: Difference in GB velocity for Σ11 22.0° at a randomly selected thermodynamic state. Blue and red indicate GB positions of grain boundary "A" and "B" respectively. The two GBs approach each other and converge on the centroid of the reference set of atoms.

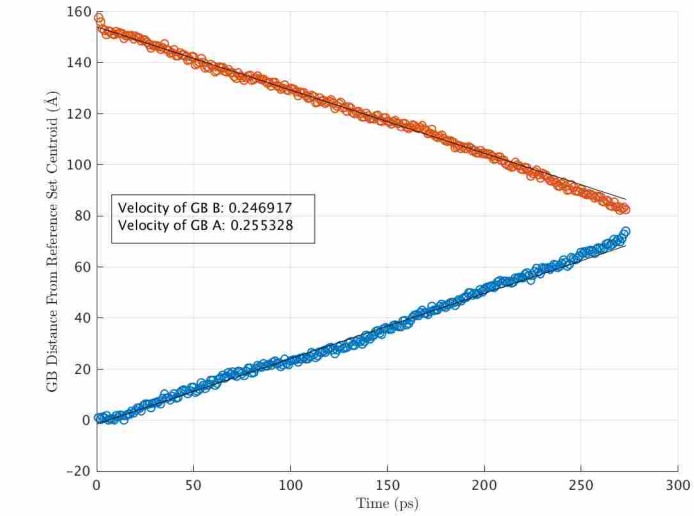


Figure B.4: Difference in GB velocity for Σ13a 23.2°. Blue and red indicate GB positions of grain boundary "A" and "B" respectively. The two GBs approach each other and converge on the centroid of the reference set of atoms.

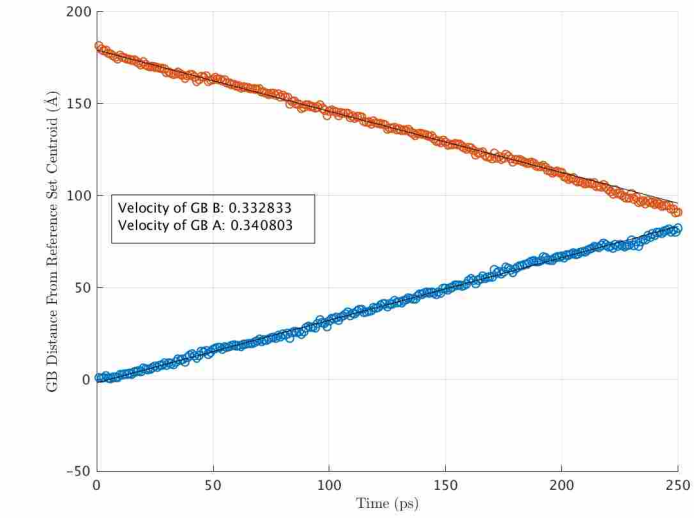


Figure B.5: Difference in GB velocity for Σ13a 31.0°. Blue and red indicate GB positions of grain boundary "A" and "B" respectively. The two GBs approach each other and converge on the centroid of the reference set of atoms.

B.1 MATLAB Script: analyzeGBvelocityRef.m

```
function [speeds] = analyzeGBvelocityRef(RunID,DF)
%%
plotme = false;
runStr = ['Run' sprintf('%05d',RunID)];
atmDiam = 1.84; %Angstroms (units of simulation cell)

%%
if ispc == 0
    prgm_loc = '~/compute/executables/ovito/bin/ovitos';
    pyth_loc = '~/compute/utilities/getGBanalysisCentrosymmetry.py';
    dump_loc = ['./zdumpall_' sprintf('%03d',DF) '.out'];
else
    prgm_loc = '"C:\Program Files\Ovito\ovitos.exe"';
    pyth_loc = 'getGBanalysisCentrosymmetry.py';
    dump_loc = 'zdump.out.gz';
end

%% Bypass the collection of reduced data if it already exists (and has data):
%filename = [runStr '_Distance' sprintf('%03d',DF) '.dat'];
filename = [runStr '_GBPositions' sprintf('%03d',DF) '.dat'];


if exist(filename,'file')==2
    % Attempt to import the reduced data if it does exist
    posData = (importdata(filename))'; %Making sure that we have a 1xcol array
    if isempty(posData)
        %If there is nothing in the datafile, go ahead and delete it
        delete(filename); %Delete the file
    end
end

```

```
if exist(filename,'file')~=2
    %% Get the number of frames to interrogate:
    fxncall = [prgm_loc ' ' pyth_loc ' ' dump_loc ' 0 2 > nframes.out'];
    system(fxncall);
    nframes = importdata('nframes.out');
    delete('nframes.out');


    %% Cycle through the frames until GBs converge

    all_loc = zeros(nframes,2);
    idx = 0;
    for frame=0:1:nframes
        %%
        idx = idx+1;
        fxncall = [prgm_loc ' ' pyth_loc ' ' dump_loc ' ' sprintf('%d',frame)
    ' 1'];
        [status,~] = system(fxncall);
        %%
        %Bring the simulation cell data in through importdata and delete
        %the file.
        simcell = importdata('simcell.dat')';
        fclose('all');
        delete('simcell.dat');
        %%
        %Bring in the atomic position/centrosymmetry data and delete the
        %file.
        atoms_csym = importdata('centrosymData.dat');
        fclose('all');
        delete('centrosymData.dat');


        atoms = atoms_csym(:,1:3);
```

```
64        csym = (atoms_csym(:,4));
65        csym = (csym>4).*csym;
66
67
68
69
70
71        %%
72        SSmin = simcell(4,2);                    %The first entry is the
    minimum of the simulation cell
73        SSmax = simcell(4,2)+simcell(2,2);                    %The second
    entry is the maximum of the simulation cell
74        SSlen = SSmax SSmin;                %Length of the simulation cell
75
76
77        %% Find a reference set of atoms
78        if frame ==0
79            targ = sum(simcell(:,2))/2+[15,15];
80            refID = atoms_csym(:,2)>targ(1) & atoms_csym(:,2)<targ(2);
81        end
82
83        %% Shift the data
84        %This method assumes that the "middle" GB1 does not move left through
85        %the periodicity. However, this is possible. A more robust code might
86        %handle that. It would have some kind of logic keeping track of the GB
87        %position in the previous frame
88        Delta = 1/8*abs(SSlen);             %Move the leftmost 1/8 of the
    simulation cell
89        Data_idx = atoms(:,2)<(SSmin+Delta);
90        atoms(:,2) = atoms(:,2)+Data_idx*SSlen;
91        SSmin = SSmin+Delta;
92        SSmax = SSmax+Delta;
93        %%
```

```
94        %Setup for and get circular ksDensity data to obtain peak positions of
95        %GBs based on a high density of non grain atoms near a position
96        Y_axis = linspace(SSmin,SSmax,10000);   %Get a linearly distributed
    space along simulation cell
97        P_axis= circ_ksdensity(atoms(:,2),Y_axis,[SSmin SSmax],atmDiam/50,csym
    ); %use a circ_ksdensity
98        %%
99        % Get the peaks
100       %Peaks representing GBs should be 1/4 the maximum of any one peak
101       %(assumes that GBs have approximately equivalent number of non grain
102       %atoms  within 40% of each other)
103       cutoff_fact = 4;
104       [pk,loc]=findpeaks(P_axis,Y_axis,'MinPeakHeight',max(P_axis)/
    cutoff_fact);
105       [~,ii]=sort(pk,'descend');
106       loc = loc(ii);
107
108       %%
109       %If there is only one GB, discontinue the cycle so that further data
    is
110       %not confused.
111       if length(loc)<2
112           disp(['Grain Boundaries Converged at Frame ' sprintf('%d',frame)])
    ;
113           break
114       else
115           loc = sort(loc(1:2)); %Take the 2 tallest peaks
116       end
117
118       %% Determine the distance of each peak relative to the reference atoms
119
120       %Find the current y position of the reference set of atoms
121       refY = mean(atoms(refID,2));
```

```
122         if frame ==0
123             dyRef = 0;
124         else
125             dyRef = refY posData(1,1);
126         end
127 
128         % Assume that the biggest peak is to the "right" of the reference
129         % set of atoms, so that the smallest peak is "A" and largest peak
130         % is "B" for the first frame. After that, we need to resolve the
131         % periodicity and nearest peak from the previous frame.
132 
133         % Previous frame:
134         %       Mirror                  REAL                    Mirror
135         % {     A   R   B   }{      A   R   B   }{      A   R   B   }
136         % {     x   x   B   }{      A   R   B   }{      A   x   x   }
137         %                       {       1   R   2   }
138         %                   Current frame ^^^
139 
140         [distAB,order] = sort(loc refY);  %Get the difference from the
    reference atoms
141         if frame ==0
142             %If it's the first frame, pick a boundary to be "A" and "B"
143             dist = distAB(1);
144             distB = distAB(2);
145             gbA  = loc(order(1));
146             gbB  = loc(order(2));
147             absAp = gbA + simcell(2,2);
148             absBp = gbB   simcell(2,2);
149         else
150             %Determine how different each "A" or "B" atom is from the
151             %previous "A" or "B" atom in the last frame
152             d_1A  = abs(loc(1) posData(idx 1,5)); %Distance from "A"
153             d_1B  = abs(loc(1) posData(idx 1,6)); %Distance from "B"
```

```
154             d_1Am = abs(loc(1) posData(idx 1,7)); %Distance from "A periodic"
155             d_1Bm = abs(loc(1) posData(idx 1,8)); %Distance from "B periodic"
156 
157             [~,idClose] = min([d_1A,d_1B,d_1Am,d_1Bm]);
158 
159             %We now know what GB the first peak is closest to. Now, time to
160             %label things and get it all sorted out.
161             switch idClose
162                 case 1 | 3
163                     %Distance to "A" is closest: This is most likely
164                     gbA = loc(1);
165                     gbB = loc(2);
166                 case 2 | 4
167                     %Distance to "B" is closest: The ids got flipped
168                     gbA = loc(2);
169                     gbB = loc(1);
170             end
171 
172             dist = gbA   refY;
173             distB = gbB   refY;
174             absAp = gbA + simcell(2,2);
175             absBp = gbB   simcell(2,2);
176         end
177 
178         %Store the data in an array for reference
179         % Column:       Meaning:
180         %  1             Position of reference atoms
181         %  2             Change from original position of reference atoms
182         %  3             Distance from reference of "A" boundary
183         %  4             Distance from reference of "B" boundary
184         %  5             Absolute location of "A" boundary
185         %  6             Absolute location of "B" boundary
186         %  7             Periodic image of "A" boundary absolute location
```

```
187         %  8              Periodic image of "B" boundary absolute location
188         posData(idx,:)=[refY,dyRef,dist,distB,gbA,gbB,absAp,absBp];
189 
190         if plotme
191             h=figure(1);clf;hold on;
192             plot(abs(posData(:,3:4)),'o')
193             pause(.1)
194         end
195 
196         %Distance(idx)=sum(abs(posData(idx,3:4)));
197         disp(['Completed Frame ' sprintf('%d',frame) ' of ' sprintf('%d',
    nframes)]);
198 
199     end
200 
201     %% Save the reduced data     fileID = fopen(filename,'w');
202     dlmwrite(filename,posData,'precision','%.6f');
203 
204 
205 else
206     %% Import the reduced data if it does exist
207     posData = (importdata(filename)); %Making sure that we have a 1xcol array
208 end
209 %%
210 
211 
212 
213 
214 %% Get Time Series:
215 steps_perFrame = 1000;
216 time_perStep = 0.001;
217 frames = 1:1:length(posData);
218 time = frames*steps_perFrame*time_perStep;
```

```
time_0 = time;
%% Filter data
firstN = 3;

gbA = abs(posData(:,3))';  %Range from reference atoms of gb "A" (absolute)
gbB = abs(posData(:,4))';  %Range from reference atoms of gb "B" (absolute)

dist = mean([gbA;gbB],1); %Get an average of the range of the two gb's for
    cutoff purposes

%Filtering serves the purposes of:
%    1: Kick out data that is smaller than a minimum range (5 angstroms)
%    2: Find a good convergence point

cutoffA = 5;  %Minimum trustable distance
%Arbitrary low cutoff There seems to be a problem if you have low distance.
iKeep = dist>cutoffA; %Find indicies where the distance is greater than the
    cutoff
distK = dist(iKeep); %This is all of the distance data that is within the
    cutoff

%Low cutoff for convergence point
cutoffB = 0.20 * mean(distK(1:firstN));
iKeep = dist>cutoffB & dist>cutoffA;

% Using this filter, keep the good data
gbA = gbA(iKeep);
gbB = gbB(iKeep);
posData2 = posData(iKeep,:);
gbA = posData2(:,3)';
gbB = posData2(:,4)';
time = time(iKeep);

```

```
%% Find out the gb "A" mobility (COLUMN 3)
% This gb is expected to always have a positive trend provided the
% conventions used through the remainder of SMGBM are followed, so positive
% slope is equal to positive mobility. I use some ABS here to get cutoffs.
% The slope, however, should be negative after doing those ABS. If you get
% a positive slope, there's a problem.


% Fit the Data to get gb velocity for GB "A"
rangA = range(gbA); %The total range of the GB movement
posSt = mean(gbA(1:firstN));  %The starting point of the GB distance
if rangA > .1*posSt
    % > Case where appreciable movement has occurred

    %Fit up until the data gets close to the cutoff applied earlier
     cutoffE = mean(gbA(1:firstN))+.9*rangA;          %Cutoff for finding end
    point
     [~,idEnd] = find(gbA>cutoffE,1,'first');    %Find the end point
     if isempty(idEnd)
         idEnd = length(gbA);                   %If we didn't find one, then
    just say it's the last one
     end

     pA = polyfit(time(1:idEnd),gbA(1:idEnd),1);

else
    % > Case where no movement has occurred
     pA=[0,posSt]; %Initialize a polyfit array with no slope and starting point
      position
end

fA = polyval(pA,[0,time(end)]);

```

```
velocityA = pA(1);  %The velocity of GB A toward the reference set of atoms


%% Find out the gb "B" mobility (COLUMN 4)
% This gb is expected to always have a negative trend provided the
% conventions used through the remainder of SMGBM are followed, so negative
% slope is equal to positive mobility

% Fit the Data to get gb velocity for GB "A"
rangB = range(gbB); %The total range of the GB movement
posSt = mean(gbB(1:firstN));  %The starting point of the GB distance
if rangB > .1*posSt
    % > Case where appreciable movement has occurred

    %Fit up until the data gets close to the cutoff applied earlier
    cutoffE =  mean(gbB(1:firstN)) .9*rangB;          %Cutoff for finding end
    point
    [~,idEnd] = find(gbB<cutoffE,1,'first');    %Find the end point
    if isempty(idEnd)
        idEnd = length(gbB);                  %If we didn't find one, then
    just say it's the last one
    end

    pB = polyfit(time(1:idEnd),gbB(1:idEnd),1);

else
    % > Case where no movement has occurred
    pB=[0,posSt]; %Initialize a polyfit array with no slope and starting point
     position
end

fB = polyval(pB,[0,time(end)]);

```

```
309 velocityB =  pB(1);  %The velocity of GB A toward the reference set of atoms
310
311
312 %% Store the velocities to a file:
313
314 speeds = ([velocityA, velocityB]);
315 filename = [runStr '_GBvelocity' sprintf('%03d',DF) '.dat'];
316 fileID = fopen(filename,'w');
317 fprintf(fileID,'%6.8f \n',speeds);
318 fclose(fileID);
319
320 %% Plot the data & save to file
321 h=figure(1);clf;hold on;
322 %plot(time,gbA,'o')
323 %plot(time,gbB,'o')
324 plot(time,posData2(:,3),'o')
325 plot(time,posData2(:,4),'o')
326
327 plot([0,time(end)],fA,'k')
328 plot([0,time(end)],fB,'k')
329
330 % Dress up the plot
331 xlabel('Time (ps)','interpreter','Latex')
332 ylabel('GB Distance From Reference Set Centroid (\AA)','interpreter','Latex')
333 grid on;
334 %title([runStr '  Driving force:' sprintf('%03d',DF) '  V Bar = ' sprintf('%f
    ',mean(speeds))])
335 atext = {['Velocity of GB A: ' sprintf('%f',speeds(1))],...
336     ['Velocity of GB B: ' sprintf('%f',speeds(2))]}
337 annotation('textbox',[.15 .5 .1 .1],'String',atext,'FitBoxToText','on')
338
339 figfn = ['./Run' sprintf('%05d',RunID) 'plot' sprintf('%03d',DF) '.png'];
340 saveas(h,figfn);
```

```
341 %close('all')
342
343 %%
344
345
346
347 end
348
349 function [vfEstimate] = circ_ksdensity(vfObservations, vfPDFSamples, vfDomain,
        fSigma, vfWeights)
350
351 % circ_ksdensity FUNCTION   Compute a kernel density estimate over a periodic
      domain
352 %
353 % Usage: [vfEstimate] = circ_ksdensity(vfObservations, vfPDFSamples,
354 %                                      <vfDomain, fSigma, vfWeights>)
355 %
356 % This function calculates a kernel density estimate of an (optionally
357 % weighted) data sample, over a periodic domain.
358 %
359 % 'vfObservations' is a set of observations made over a periodic domain,
360 % optionally defined by 'vfDomain': [fMin fMax]. The default domain is
361 % [0..2*pi]. 'vfPDFSamples' defines the sample points over which to perform
362 % the kernel density estimate, over the same domain as 'vfObservations'.
363 %
364 % Weighted estimations can be performed by providing the optional argument
365 % 'vfWeights', where each element in 'vfWeights' corresponds to the
366 % matching element in 'vfObservations'.
367 %
368 % The kernel density estimate will be performed using a wrapped Gaussian
369 % kernel, with a width estimated as
370 %    (4/3)^0.2 * circ_std(vfObservations, vfWeights) * (length(vfObservations
      ^ 0.2)
```

```
371 %
372 % The optional argument 'fSigma' can be provided to set the width of the
373 % kernel.
374 %
375 % 'vfEstimate' will be a vector with a (weighted) estimate of the
376 % underlying distribution, with an entry for each element of
377 % 'vfPDFSamples'. If no weighting is supplied, the estimate will be scaled
378 % such that it forms a PDF estimate over the supplied sample domain, taking
379 % into account sample bin widths. If a weight vector is supplied then the
380 % estimate will be scaled such that the sum over the domain attempts to
381 % match the sum of weights, taking into account sample bin widths.
382
383 % Author: Dylan Muir <dylan.muir@unibas.ch>
384 % Created: 23rd October, 2013
385
386 %    Defaults
387
388 DEF_vfDomain = [0 2*pi];
389
390
391 %    Check arguments
392
393 if (nargin < 2)
394     help circ_ksdensity;
395     error('circ_ksdensity:Usage', ...
396         '*** circ_ksdensity: Incorrect usage');
397 end
398
399 if (~exist('vfDomain', 'var') || isempty(vfDomain))
400     vfDomain = DEF_vfDomain;
401 end
402
403 %   Do we need to estimate fSigma?
```

```
404 if (~exist('fSigma', 'var'))
405     %   Sigma will be estimated
406     fSigma = [];
407 end
408
409 vfObservations = vfObservations(:);
410 vnPSFSamplesSize = size(vfPDFSamples);
411 vfPDFSamples = vfPDFSamples(:);
412
413 %   If weights are not provided, weight each observation equally
414 if (~exist('vfWeights', 'var'))
415     vfWeights = ones(size(vfObservations)) ./ numel(vfObservations);
416
417     %   Check the number of observations matches the number of weights
418 elseif (numel(vfObservations) ~= numel(vfWeights))
419     error('circ_ksdensity:Usage', ...
420         '*** circ_ksdensity: The number of observations must be equal to the
    number of weights.');
421 end
422
423
424 %    Map everything to [0..2pi] and wrap over domain
425
426 vfObservations = (vfObservations   vfDomain(1)) ./ diff(vfDomain) .* 2*pi;
427 vfObservations = mod(vfObservations, 2*pi);
428
429 vfPDFSamples = (vfPDFSamples   vfDomain(1)) ./ diff(vfDomain) .* 2*pi;
430 vfPDFSamples = mod(vfPDFSamples, 2*pi);
431
432
433 %   Estimate sigma, if necessary
434 if (isempty(fSigma))
```

```
435     fSigma = (4/3)^0.2 * circ_std(vfObservations, vfWeights) * (numel(
        vfObservations)^ 0.2);
436 end
437
438
439 %    Pad observations above and below domain
440
441 vfObservations = [vfObservations;
442     vfObservations   2*pi;
443     vfObservations + 2*pi];
444 vfWeights = repmat(vfWeights, 3, 1) ./ 3;
445
446
447 %    Perform kernel density estimate
448
449 vfEstimate = ksdensity(vfObservations, vfPDFSamples, 'weights', vfWeights', '
        width', fSigma);
450
451 %   Reshape return to match shape of 'vfPDFSamples'
452 vfEstimate = reshape(vfEstimate, vnPSFSamplesSize);
453
454 %   Correct scaling of histogram estimate
455 vfEstimate = vfEstimate .* 3 .* sum(vfWeights);
456
457 end
458 %     END of circ_ksdensity FUNCTION
459
460
461 % circ_std FUNCTION Estimate the weighted circular standard deviation of a
        dataset
462 function [s s0] = circ_std(alpha, w)
463
464 % compute mean resultant vector length
```

```
r = circ_r(alpha,w);

s = sqrt(2*(1 r));        % 26.20
s0 = sqrt( 2*log(r));     % 26.21
end

% circ_r FUNCTION Compute the weighted resultant of a dataset
function r = circ_r(alpha, w)

% compute weighted sum of cos and sin of angles
r = sum(w.*exp(1i*alpha),1);

% obtain length
r = abs(r)./sum(w,1);
end

%      END of circ_ksdensity.m
```

APPENDIX C. HPC OPTIMIZATION & WORKFLOW

Due to the large number of simulations required, it was critical to investigate the computational effectiveness of the compiled code and determine methods to best utilize the compute resources at BYU's FSL.

One of the constraints that limits use of the FSL resources is "wall-time." Wall-time is the total time observed by a standard "wall" clock. For example, m7 nodes have a wall time of 3 days. For this reason, it is advantageous to make use of the Message Passing Interface (MPI) utilized in LAMMPS for using parallel processing. A study was conducted on molecular dynamics simulations like those performed in this research. As observed in figure C.1 , it was found that as the number of threads (cores utilized) increases, the total wall time does decrease. However, there is little benefit of using more than 8 cores. The underlying problem is that more computational time is spent in the MPI subsystem than in actual computation as the number of threads increases. Additionally, the greatest common multiple of the number of cores per node is 4.

For these reasons, the computations performed with this research was conducted on 4 cores per node. This permitted the maximum amount of parallel work to be performed across multiple different simulation types. Utilizing parallel processing also reduced wall-time so that more simulations could be performed in a reasonable amount of time. The following statistics summarize the total usage of the FSL in conducting this research.

C.1 Dataset

The resultant dataset with atomic positions at each timestep were saved. The dataset is approximately 30TB in size (compressed) and is currently stored on a cloud storage service.

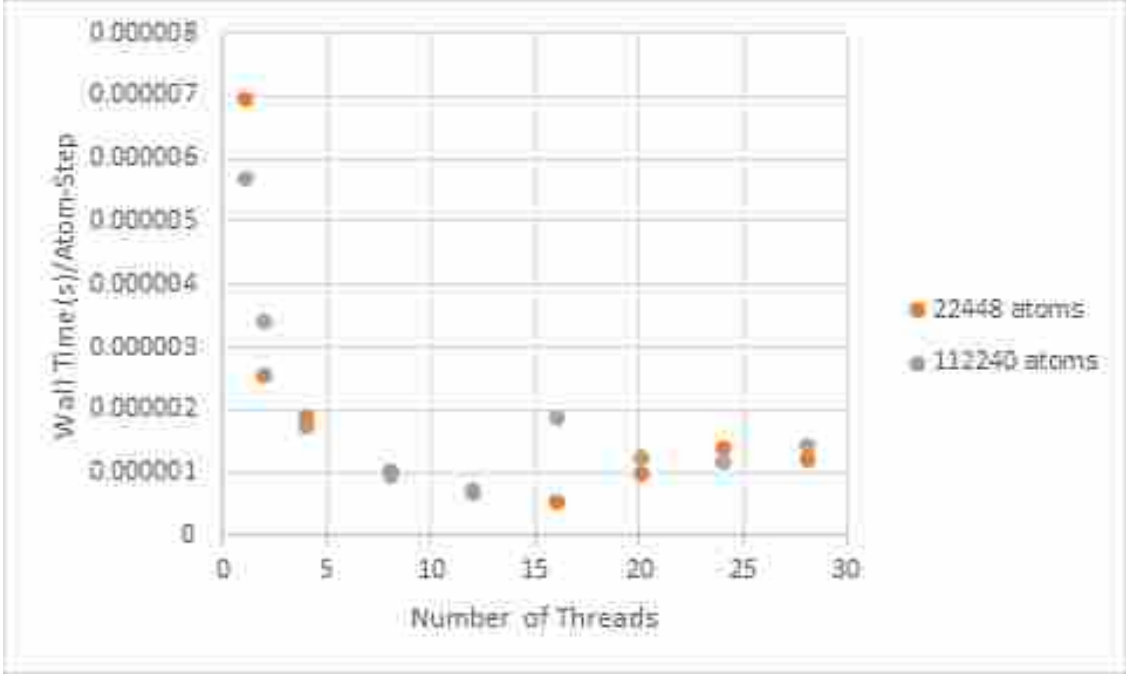


Figure C.1: Characteristic performance curves showing parallel efficiency of LAMMPS while performing a random walk (Mobility Run)

C.2 Usage Statistics

Total processor time: 316,621,308 CPU Minutes

APPENDIX D. EXAMPLE LAMMPS INPUT FILES & CORRESPONDING SHELL SCRIPTS

D.1 LAMMPS Input Scripts

D.1.1 Thermalizing Calculations

```
# Thermalizing Run for Grain Boundary Structure
# Derek Lontine
# October 19 2017
# This code will read in the file ./rawstructure.dat (a atom position file)
    and apply temperature and mechanical loads to it
# It will equilibrate at the given temperature and stress for 40ps (40000
    steps). Typical runs for files having 3000 particles
# is about 10 minutes with 12 processors on an MPI run. Note that this setup
    replicates the geometry, and speedups or slowdowns
# may be required depending on the geometrical size of the provided structure
log ./log.thermalize

### Initialization
units metal                     #Angstroms for distance, bar for pressure
boundary p p p                    #Fully periodic GB structures will be used
atom_style atomic                 #
pair_style eam/alloy                #The potential used is a eam potential

#Variables that are modified per run basis:
variable atm_dist equal 2.8638
variable r_temp equal 739
variable r_sxx equal 39186
```

```
variable r_syy equal 39186
variable r_szz equal 39186
variable r_sxy equal 454
variable r_syz equal 1050
variable r_sxz equal 0
variable rep_x equal 2
variable rep_y equal 1
variable rep_z equal 5
variable rand_seed equal 13925
variable r_tkick equal ${r_temp}*2        #The value for initializing velocity
variable lat_param equal ${atm_dist}*sqrt(2)  #The value for lattice parameter
    from atm_dist (static fur Therm Run)

### Set up Simulation box   The structure is already minimized
lattice fcc ${lat_param}              #Derived from inputs above (from
   minimized structures)
read_data ./rawStructure.dat          #This line should change to show the
   actual structure used
replicate ${rep_x} ${rep_y} ${rep_z}      #This line will need to be modified
   on a case by case basis
pair_coeff * * /fslhome/dlontine/compute/lammps/Tschopp/Potentials/Al99.eam.
   alloy Al Al Al Al Al      #The location of the eam potential
neighbor 2.0 bin
neigh_modify every 1 delay 10 check yes     #Updates the neighbor list every
   10 steps
change_box all triclinic            #This line permits the application of
   shear stresses

### Set up simulation Parameters
timestep 0.001                  #1 fempto second is the time step

### Run NVE Equilization
```

```
velocity all create ${r_tkick} ${rand_seed}   # The 2x temperature kick and the random number seed
run 0                       # Prep for next line
velocity all scale ${r_tkick}          # Ensures that temperature is fixed for all atoms
fix enviy all nve                 # Sets everything to follow NVE (number of atoms, volume, energy)

dump 1 all custom 500 ./zdumpTherm.all.out id type x y z #This may not be needed

thermo 1000                     # Outputs thermo information every 1000 timesteps
thermo_modify flush yes               # Makes sure that output is up to date when pushed to file
run 10000                     # cycles the model 10000 cycles to start random vibrations
unfix enviy                     # removes the condition in fix

### Run NPT Equilization

#The following line is what changes the temperature and stress as per the variables detailed above
fix my_npt all npt &
     temp ${r_temp} ${r_temp} 0.1 &
      x ${r_sxx} ${r_sxx} 0.5 &
      y ${r_syy} ${r_syy} 0.5 &
      z ${r_szz} ${r_szz} 0.5 &
      xy ${r_sxy} ${r_sxy} 0.5 &
      yz ${r_syz} ${r_syz} 0.5 &
      xz ${r_sxz} ${r_sxz} 0.5
thermo 1000                     #Update output every 1000 cycles
```

```
run 40000                    #Cycle for 40ps to equilibrate at stress and
    temperature

### Cleanup
write_dump all custom ./thermDump.out id type x y z #Write a single dump file
    of the final state
unfix my_npt                   #remove the fixity (mobility run gets confused
    if it is in the restart file)
write_restart ./thermalized.f.restart      #Write a restart file for use by
    mobility run
```

D.1.2 Mobility Calculations

```
# Mobility experiment on stress modulated grain boundaries
# Derek Lontine
# This code will run a grain boundary mobility experimet using orient/fcc to
    apply an artificial driving force
log ./log.mobility025
### Initialization
units metal                    #Angstroms for distance, bar for pressure
boundary p p p                   #Fully periodic GB structures will be used
atom_style atomic                #
pair_style eam/alloy               #The potential used is a eam potential

### Variables that are modified per run basis:
variable atm_dist2 equal 2.8638
variable r_temp2 equal 725
variable r_sxx2 equal 776
variable r_syy2 equal 776
variable r_szz2 equal 776
variable r_sxy2 equal  497
variable r_syz2 equal 1050
variable r_sxz2 equal 0
```

```
variable drv_frce equal 0.025
variable lat_param2 equal ${atm_dist2}*sqrt(2)  #The value for lattice
    parameter from atm_dist

lattice fcc ${lat_param2}

### Pull in restart file and apply potential
read_restart ./thermalized.f.restart        #This is from the ThermRun output
pair_coeff * * /fslhome/dlontine/compute/lammps/Tschopp/Potentials/Al99.eam.
    alloy Al Al Al Al Al
neighbor 2.0 bin
neigh_modify every 1 delay 10 check yes
change_box all triclinic
reset_timestep 0
timestep 0.001

### Mobility Experiment Definition
#The following line is what changes the temperature and stresses as per the
    variables detailed above
fix my_npt2 all npt &
  temp ${r_temp2} ${r_temp2} 0.1 &
  x ${r_sxx2} ${r_sxx2} 0.5 &
  y ${r_syy2} ${r_syy2} 0.5 &
  z ${r_szz2} ${r_szz2} 0.5 &
  xy ${r_sxy2} ${r_sxy2} 0.5 &
  yz ${r_syz2} ${r_syz2} 0.5 &
  xz ${r_sxz2} ${r_sxz2} 0.5
fix gb all orient/fcc 0 1 ${lat_param2} ${drv_frce} 0.25 0.75 ./grainA__.vec
    ./grainB__.vec
dump 1 all custom 1000 ./zdumpall_025.out id type x y z


thermo 1000                    # update screen output every 1000 steps
```

```
run 800000                          # run mobility experiment
write_restart ./Mob025.restart.800000
```

D.2 Shell Scripts

D.2.1 Thermalizing Job

```
#!/bin/bash

#Submit this script with: sbatch try

#SBATCH   time=01:07:54 # walltime
#SBATCH   ntasks=4 # number of processor cores
#SBATCH   mem per cpu=512M # memory per CPU core
#SBATCH   workdir=/fslhome/dlontine/compute/lammps/SMGBM/Run00007
#SBATCH  J "R00007_T"


# Set the max number of threads to use for programs using OpenMPI. Should be
    <= ppn. Does nothing if the program does not use OpenMP

#OUTFILE=""
module purge
module load gcc/5.5 mpi/openmpi 3.0_gcc 5.5
module load matlab/r2017a
module load lammps_mpi/2017 08 11_openmpi 3.0_gcc 5.5
#Run LAMMPS
mpirun lmp_mpi < ./ThermRun.in
#Post Process
matlab  r 'try MakeVecsLinux();catch;end;quit;'

sbatch ./MobRun025.sh > mobBatch025.out

gzip zdumpTherm.all.out
```

```
#exit 0
```

D.2.2 Mobility Job

```
#!/bin/bash

#Submit this script with: sbatch try

#SBATCH    time=29:47:06 # walltime
#SBATCH    ntasks=4 # number of processor cores
#SBATCH    mem per cpu=512M # memory per CPU core
#SBATCH    workdir=/fslhome/dlontine/compute/lammps/SMGBM/Run51381
#SBATCH   J "R51381_M25"

# Set the max number of threads to use for programs using OpenMPI. Should be
    <= ppn. Does nothing if the program does not use OpenMP
module purge
module load gcc/5.5 mpi/openmpi 3.0_gcc 5.5
module load lammps_mpi/2017 08 11_openmpi 3.0_gcc 5.5
#OUTFILE=""
mpirun lmp_mpi < ./MobRun025.in

#Post process the data
gzip zdumpall_025.out

# Submit next job
sbatch ./MobRun030.sh > mobBatch030.out
#exit 0
```

APPENDIX E. GB POSITION TRACKING ALGORITHM

E.1 OVITO Scripts

```
#getNonFCCatomsArray.py
#Author: Derek Lontine
#Start date: 9/22/2017
#See EOF for revision history.

# Intended for use with fully periodic grain boundary structures where multiple grains are moving
# Extract the positions of all GB atoms from a dump file that contains multiple timesteps in one dump file.

#Excecuted by running the function call
# ovitos.exe this_script filename_in frame type

#The filename_in MAY be a compressed file output.out.gz, however performance suffers significantly (typically 4x longer compute time)
#If you desire to use compressed files, run a decompression, then this code, then compression


# Import OVITO modules.
from ovito import *
from ovito.io import *
from ovito.modifiers import *
from ovito.data import *
import numpy as np
```

```
22 import os
23 from sys import argv
24 from sys import stdout
25 import pdb #used for debugging, should be removed later
26
27
28 assert(version[0] >= 2 and version[1] >=7)
29
30
31 # Get the input and output files from std::in
32 if len(argv) < 2:
33   raise Exception("Filename input required as an argument.")
34 if len(argv) < 3:
35   raise Exception("Frame number required as an argument.")
36 if len(argv) < 4:
37   raise Exception("Type required as an argument.")
38
39 #Only two additional arguments are required (both filepath/name)
40 #If being executed from the target directory the self reference './' is
       acceptable
41 filename_in = argv[3]
42 req_frame = int(argv[2])
43 type = int(argv[1])
44
45 #Import the file
46 ## There is no dictated frequency for framerate that will need to be handled
       separately
47 node = import_file(filename_in, multiple_frames = True)
48
49 #Apply the centrosysmmetry modifier
50 csym = CentroSymmetryModifier()
51 node.modifiers.append(csym)
52 node.compute
```

```
53
54 #Cycle through the group of frames
55 numframes = node.source.num_frames
56 frames = range(numframes)
57
58
59
60 if type ==1:
61   dataset.anim.current_frame = req_frame
62   node.compute()
63   #Simulation cell and output
64   simcell = node.output.cell.matrix
65   np.savetxt('simcell.dat',simcell,delimiter="\t") #Save to the text file
       simcell.dat
66
67   #Get the positions for each remaining atom
68   posn = node.output.particle_properties.position.marray
69   csym = node.output.particle_properties.centrosymmetry.marray
70   nYs = len(posn)
71
72   #Combine these data into a single array
73   cdata = np.zeros((nYs,4))
74   cdata[:,:1]=posn
75   cdata[:,1] =csym
76   #Save the array to a tab delimited dat file.
77   np.savetxt('centrosymData.dat',cdata,delimiter="\t") #Save to a text file
78 if type==2:
79   #Print out the number of frames
80   print(numframes)
81 #EOF
82
83 #Revision: 2018 01 30  Derek Lontine
```

```
# Changed so that io through standard out was not used. Created problem with
    parsing when using matlab interface
```

E.2 MATLAB Script: analyzeMobility.m

```
function [mobility] = analyzeMobility(RunID,DF)
%%
runStr = ['Run' sprintf('%05d',RunID)];
atmDiam = 1.84; %Angstroms (units of simulation cell)

%%
if ispc == 0
    prgm_loc = '~/compute/executables/ovito/bin/ovitos';
    pyth_loc = '~/compute/utilities/getGBanalysisCentrosymmetry.py';
    dump_loc = ['./zdumpall_' sprintf('%03d',DF) '.out'];
else
    prgm_loc = '"C:\Program Files\Ovito\ovitos.exe"';
    pyth_loc = 'getGBanalysisCentrosymmetry.py';
    dump_loc = 'zdump.out.gz';
end

%% Bypass the collection of reduced data if it already exists (and has data):
filename = [runStr '_Distance' sprintf('%03d',DF) '.dat'];

if exist(filename,'file')==2
    % Attempt to import the reduced data if it does exist
    Distance = (importdata(filename))'; %Making sure that we have a 1xcol
    array
    if isempty(Distance)
        %If there is nothing in the datafile, go ahead and delete it
        delete(filename); %Delete the file
    end
end
```

```
28
29 if exist(filename,'file')~=2
30     %% Get the number of frames to interrogate:
31     fxncall = [prgm_loc ' ' pyth_loc ' ' dump_loc ' 0 2 > nframes.out'];
32     system(fxncall);
33   nframes = importdata('nframes.out');
34   delete('nframes.out');
35
36     %% Cycle through the frames until GBs converge
37
38     all_loc = zeros(nframes,2);
39     for frame=134:1:nframes
40         %%
41         idx = frame+1;
42         fxncall = [prgm_loc ' ' pyth_loc ' ' dump_loc ' ' sprintf('%d',frame)
    ' 1'];
43         [status,~] = system(fxncall);
44         %%
45         %Bring the simulation cell data in through importdata and delete
46         %the file.
47         simcell = importdata('simcell.dat')';
48         delete('simcell.dat');
49         %%
50         %Bring in the atomic position/centrosymmetry data and delete the
51         %file.
52         atoms_csym = importdata('centrosymData.dat');
53         delete('centrosymData.dat');
54         atoms = atoms_csym(:,1:3);
55         csym = (atoms_csym(:,4));
56         csym = (csym>4).*csym;
57
58         %%
```

```
59      SSmin = simcell(4,2);                       %The first entry is the
    minimum of the simulation cell
60      SSmax = simcell(4,2)+simcell(2,2);                      %The second
    entry is the maximum of the simulation cell
61      SSlen = SSmax SSmin;                %Length of the simulation cell
62
63      %%
64
65      %Shift the data
66      %This method assumes that the "middle" GB1 does not move left through
67      %the periodicity. However, this is possible. A more robust code might
68      %handle that. It would have some kind of logic keeping track of the GB
69      %position in the previous frame
70      Delta = 1/8*abs(SSlen);             %Move the leftmost 1/8 of the
    simulation cell
71      Data_idx = atoms(:,2)<(SSmin+Delta);
72      atoms(:,2) = atoms(:,2)+Data_idx*SSlen;
73      SSmin = SSmin+Delta;
74      SSmax = SSmax+Delta;
75      %%
76      %Setup for and get circular ksDensity data to obtain peak positions of
77      %GBs based on a high density of non grain atoms near a position
78      Y_axis = linspace(SSmin,SSmax,10000);   %Get a linearly distributed
    space along simulation cell
79      P_axis= circ_ksdensity(atoms(:,2),Y_axis,[SSmin SSmax],atmDiam/50,csym
    ); %use a circ_ksdensity
80      %%
81      % Get the peaks
82      %Peaks representing GBs should be 1/4 the maximum of any one peak
83      %(assumes that GBs have approximately equivalent number of non grain
84      %atoms  within 40% of each other)
85      cutoff_fact = 4;
```

```
86        [pk,loc]=findpeaks(P_axis,Y_axis,'MinPeakHeight',max(P_axis)/
    cutoff_fact);
87        [~,ii]=sort(pk,'descend');
88        loc = loc(ii);
89
90        %%
91        %If there is only one GB, discontinue the cycle so that further data
    is
92        %not confused.
93        if length(loc)<2
94            disp(['Grain Boundaries Converged at Frame ' sprintf('%d',frame)])
    ;
95            break
96        else
97            loc = loc(1:2); %Take the 2 tallest peaks
98        end
99
100       %% Plot the peaks:
101       %{
102       h=figure(2);clf;hold on;
103       plot(Y_axis,P_axis,'.')
104       %set(gca,'YScale','log')
105       cut = ones(2,1)*max(P_axis)/cutoff_fact;
106       cuty = [max(Y_axis),min(Y_axis)];
107       plot(cuty,cut);
108       %}
109
110
111       %Calculate the distance in the "Right" direction. This is assuming
    that
112       %the "first" GB is nearest to the center of the structure and that it
113       %ALWAYS stays like that.
114
```

```
115         if frame>0
116             %How far from the last frame's GB1 are you?
117             relY = all_loc(idx-1,1); %Position of previous frame's GB1
118             [~,GB1i]=min(abs(loc - relY)); %Inner GB (GB1)
119             [~,GB2i]=max(abs(loc - relY)); %Outer GB (GB2)
120         else
121             [~,GB1i]=min(abs(loc)); %Inner GB (GB1)
122             [~,GB2i]=max(abs(loc)); %Outer GB (GB2)
123         end
124
125         all_loc(idx,:)=[loc(GB1i),loc(GB2i)]';
126
127         %Store the positions
128         GB1y=loc(GB1i);
129         GB2y=loc(GB2i);
130
131         %Resolve periodicity
132         if loc(GB2i)<SSmin/2
133             GB2y=loc(GB2i)+(SSmax - SSmin);
134         end
135
136
137         Distance(idx)=GB2y - GB1y; %#ok<AGROW>
138         disp(['Completed Frame ' sprintf('%d',frame) ' of ' sprintf('%d',
    nframes)]);
139
140     end
141
142     %% Save the reduced data
143     fileID = fopen(filename,'w');
144     fprintf(fileID,'%6.8f \n',Distance);
145     fclose(fileID);
146
```

```
147
148 else
149     %% Import the reduced data if it does exist
150     Distance = (importdata(filename))'; %Making sure that we have a 1xcol
        array
151 end
152 %%
153 %
154 %
155 %
156 %
157 %
158
159 %% Get Time Series:
160 steps_perFrame = 1000;
161 time_perStep = 0.001;
162 frames = 1:1:length(Distance);
163 time = frames*steps_perFrame*time_perStep;
164
165 %% Fit the data to get mobility
166 ragD = max(abs(Distance)) min(abs(Distance));
167 if ragD > .1*max(Distance)
168     %Mobility has occurred
169     DDev= Distance(1) .1*ragD; %Start of mobility
170     DEnd = min(abs(Distance))+.1*ragD;
171
172     [~,is] = find(DDev<Distance(1),1,'first');
173     [~,ie] = find(Distance<DEnd,1,'first');
174     ie = ie 1;
175
176     p = polyfit(time(is:ie),Distance(is:ie),1);
177 else
178     %Mobility has not occurred
```

```
179     p=[0,Distance(1)];
180     is = 1;
181     ie = length(Distance);
182 end
183
184 f = polyval(p,[0,time(ie)]);
185 %% Calculate the mobility from the fit
186 speed = mean(abs(p(1)))/2;
187 DForce = DF/1000;
188 mobility = speed / DForce; %Angstroms/(ps eV)
189
190 filename = [runStr '_Mobility' sprintf('%03d',DF) '.dat'];
191 fileID = fopen(filename,'w');
192 fprintf(fileID,'%6.8f \n',mobility);
193 fclose(fileID);
194
195 %% Plot the data & save to file
196 h=figure(1);clf;hold on;
197 plot(time,Distance);
198 plot([0,time(ie)],f);
199 plot(time(is),Distance(is),'or')
200 plot(time(ie),Distance(ie),'ok')
201 % Dress up the plot
202 xlabel('Time (ps)','interpreter','Latex')
203 ylabel('Distance Between GB''s (\AA)','interpreter','Latex')
204 grid on;
205 title([runStr '  Driving force:' sprintf('%03d',DF) '  Mobility = ' sprintf('%
      f',mobility)])
206
207
208 figfn = ['./Run' sprintf('%05d',RunID) '_plot' sprintf('%03d',DF) '.png'];
209 saveas(h,figfn);
210
```

```
211 %%
212
213
214
215 end
216
217 function [vfEstimate] = circ_ksdensity(vfObservations, vfPDFSamples, vfDomain,
        fSigma, vfWeights)
218
219 % circ_ksdensity FUNCTION   Compute a kernel density estimate over a periodic
       domain
220 %
221 % Usage: [vfEstimate] = circ_ksdensity(vfObservations, vfPDFSamples,
222 %                                      <vfDomain, fSigma, vfWeights>)
223 %
224 % This function calculates a kernel density estimate of an (optionally
225 % weighted) data sample, over a periodic domain.
226 %
227 % 'vfObservations' is a set of observations made over a periodic domain,
228 % optionally defined by 'vfDomain': [fMin fMax]. The default domain is
229 % [0..2*pi]. 'vfPDFSamples' defines the sample points over which to perform
230 % the kernel density estimate, over the same domain as 'vfObservations'.
231 %
232 % Weighted estimations can be performed by providing the optional argument
233 % 'vfWeights', where each element in 'vfWeights' corresponds to the
234 % matching element in 'vfObservations'.
235 %
236 % The kernel density estimate will be performed using a wrapped Gaussian
237 % kernel, with a width estimated as
238 %    (4/3)^0.2 * circ_std(vfObservations, vfWeights) * (length(vfObservations
       ^ 0.2)
239 %
240 % The optional argument 'fSigma' can be provided to set the width of the
```

```
241 % kernel.
242 %
243 % 'vfEstimate' will be a vector with a (weighted) estimate of the
244 % underlying distribution, with an entry for each element of
245 % 'vfPDFSamples'. If no weighting is supplied, the estimate will be scaled
246 % such that it forms a PDF estimate over the supplied sample domain, taking
247 % into account sample bin widths. If a weight vector is supplied then the
248 % estimate will be scaled such that the sum over the domain attempts to
249 % match the sum of weights, taking into account sample bin widths.
250
251 % Author: Dylan Muir <dylan.muir@unibas.ch>
252 % Created: 23rd October, 2013
253
254 %    Defaults
255
256 DEF_vfDomain = [0 2*pi];
257
258
259 %    Check arguments
260
261 if (nargin < 2)
262     help circ_ksdensity;
263     error('circ_ksdensity:Usage', ...
264         '*** circ_ksdensity: Incorrect usage');
265 end
266
267 if (~exist('vfDomain', 'var') || isempty(vfDomain))
268     vfDomain = DEF_vfDomain;
269 end
270
271 %   Do we need to estimate fSigma?
272 if (~exist('fSigma', 'var'))
273     %   Sigma will be estimated
```

```
274     fSigma = [];
275 end
276 
277 vfObservations = vfObservations(:);
278 vnPSFSamplesSize = size(vfPDFSamples);
279 vfPDFSamples = vfPDFSamples(:);
280 
281 %   If weights are not provided, weight each observation equally
282 if (~exist('vfWeights', 'var'))
283     vfWeights = ones(size(vfObservations)) ./ numel(vfObservations);
284 
285     %   Check the number of observations matches the number of weights
286 elseif (numel(vfObservations) ~= numel(vfWeights))
287     error('circ_ksdensity:Usage', ...
288         '*** circ_ksdensity: The number of observations must be equal to the
    number of weights.');
289 end
290 
291 
292 %    Map everything to [0..2pi] and wrap over domain
293 
294 vfObservations = (vfObservations   vfDomain(1)) ./ diff(vfDomain) .* 2*pi;
295 vfObservations = mod(vfObservations, 2*pi);
296 
297 vfPDFSamples = (vfPDFSamples   vfDomain(1)) ./ diff(vfDomain) .* 2*pi;
298 vfPDFSamples = mod(vfPDFSamples, 2*pi);
299 
300 
301 %   Estimate sigma, if necessary
302 if (isempty(fSigma))
303     fSigma = (4/3)^0.2 * circ_std(vfObservations, vfWeights) * (numel(
    vfObservations)^ 0.2);
304 end
```

```
305
306
307 %    Pad observations above and below domain
308
309 vfObservations = [vfObservations;
310     vfObservations   2*pi;
311     vfObservations + 2*pi];
312 vfWeights = repmat(vfWeights, 3, 1) ./ 3;
313
314
315 %    Perform kernel density estimate
316
317 vfEstimate = ksdensity(vfObservations, vfPDFSamples, 'weights', vfWeights', '
    width', fSigma);
318
319 %   Reshape return to match shape of 'vfPDFSamples'
320 vfEstimate = reshape(vfEstimate, vnPSFSamplesSize);
321
322 %   Correct scaling of histogram estimate
323 vfEstimate = vfEstimate .* 3 .* sum(vfWeights);
324
325 end
326 %     END of circ_ksdensity FUNCTION
327
328
329 % circ_std FUNCTION Estimate the weighted circular standard deviation of a
    dataset
330 function [s s0] = circ_std(alpha, w)
331
332 % compute mean resultant vector length
333 r = circ_r(alpha,w);
334
335 s = sqrt(2*(1 r));      % 26.20
```

```
336 s0 = sqrt( 2*log(r));     % 26.21
337 end
338
339 % circ_r FUNCTION Compute the weighted resultant of a dataset
340 function r = circ_r(alpha, w)
341
342 % compute weighted sum of cos and sin of angles
343 r = sum(w.*exp(1i*alpha),1);
344
345 % obtain length
346 r = abs(r)./sum(w,1);
347 end
348
349 %      END of circ_ksdensity.m
```

APPENDIX F. VALIDATION MODEL TOOLS

F.1 Primary Validation Script (Including Plots)

```
% Model Random Reduced Data for Validation
% Derek Lontine
% Brigham Young University


clc;
clear;
%%
load ../structures.mat;
load ../DOEv2.mat;

id = 33;

GBstructure = categorical(cellstr([structures{id}.Name]));
SubSet = DOEv2(DOEv2.Structure_Name==GBstructure,:);
nMeas = height(SubSet);

samples = 10:2:height(SubSet);
nFULL = height(SubSet);


%% What is the most precise data we can get?
structure= LontineModelFitting(structures{id},SubSet,false);

% Estimates
```

```
M0_FULL  = structure.mCoefficients.Estimate(1) * ones(1,length(samples));
U_FULL   = structure.mCoefficients.Estimate(2) * ones(1,length(samples));
V_FULL   = structure.mCoefficients.Estimate(3) * ones(1,length(samples));
V12_FULL = structure.mCoefficients.Estimate(4) * ones(1,length(samples));
V23_FULL = structure.mCoefficients.Estimate(5) * ones(1,length(samples));
% Errors
M0_FSE   = structure.mCoefficients.SE(1) * ones(1,length(samples));
U_FSE    = structure.mCoefficients.SE(2) * ones(1,length(samples));
V_FSE    = structure.mCoefficients.SE(3) * ones(1,length(samples));
V12_FSE  = structure.mCoefficients.SE(4) * ones(1,length(samples));
V23_FSE  = structure.mCoefficients.SE(5) * ones(1,length(samples));

R2_FULL   = structure.model.Rsquared.Ordinary * ones(1,length(samples));


%% How good can we get with sub sampling

count = 1;
for i = 1:1:length(samples)
    mn = min(100,nFULL samples(i));
    for j=1:1:mn

        npts = round(samples(i));
        idxUSE = sort(randsample(nMeas,npts));
        SubSet_Use = SubSet(idxUSE,:);

        SubSet_Val = SubSet;
        SubSet_Val(idxUSE,:)=[];
        idxVAL = sort(randsample(height(SubSet_Val),mn));
        SubSet_Val=SubSet_Val(idxVAL,:);


        try
```

```
            structure= LontineModelFitting(structures{id},SubSet_Use,false);
            M0(count) = structure.mCoefficients.Estimate(1);
            U(count) = structure.mCoefficients.Estimate(2);
            V(count) = structure.mCoefficients.Estimate(3);
            V12(count) = structure.mCoefficients.Estimate(4);
            V23(count) = structure.mCoefficients.Estimate(5);
            STAT(count) = ValidateModel(structure,SubSet_Val,'Rsq');
            ST2(j) = STAT(count);
            SAMP(count) = samples(i);
            count = count+1;
        catch
            disp('Weirdness is here')
        end


    end
    ST(i) = mean(ST2(ST2>0));
end


%% Make some nice looking plots!
fx = .5;    %figure position in cm
fy = 20;     %figure position in cm
fw = 10;   %figure width in cm
fh = 10;   %figure height in cm
fx_step = 11;
fy_step =  15;

%% Plot Validation Data Fitting

for i=1:1:length(samples)
    xpos = samples(i);
    subVal = STAT(SAMP==xpos);
```

```
92      CT(i,1) = prctile(subVal,5);
93      CT(i,2) = prctile(subVal,95);
94  end
95
96  h=figure(1);clf;hold on;
97  set(h, 'WindowStyle', 'normal');
98  set(h, 'Units', 'centimeters');
99  set(h, 'PaperUnits', 'centimeters');
100 set(h, 'PaperSize', [fw fh]);
101 set(h, 'Position',[(fx+0*fx_step) (fy+0*fy_step) fw fh]);
102
103
104
105 plot(SAMP/nFULL*100,STAT,'.','color',[0,0,0]+.6)
106 plot(samples/nFULL*100,R2_FULL,'  r','LineWidth',3)
107 plot(samples/nFULL*100,CT(:,1),' m','Linewidth',2)
108 plot(samples/nFULL*100,CT(:,2),' m','Linewidth',2)
109 plot(samples/nFULL*100,ST,' k','Linewidth',2)
110
111 xlabel('\% of Data Used for Model','Interpreter','latex')
112 ylabel('Goodness of Fit for Validation Data vs Model ($R^2$)','Interpreter','
        latex')
113 grid on;
114 xlim([0,80])
115 grid on;
116 ylim([.8,1])
117 MatlabFigure2Latex('ValidateR2.png',1,'png',false)
118
119 %% Plot Exponential Prefactor
120
121 for i=1:1:length(samples)
122     xpos = samples(i);
123     subVal = M0(SAMP==xpos);
```

```
124     CT(i,1) = prctile(subVal,5);
125     CT(i,2) = prctile(subVal,95);
126 end
127
128 h=figure(2);clf;hold on;
129 set(h, 'WindowStyle', 'normal');
130 set(h, 'Units', 'centimeters');
131 set(h, 'PaperUnits', 'centimeters');
132 set(h, 'PaperSize', [fw fh]);
133 set(h, 'Position',[(fx+1*fx_step) (fy+0*fy_step) fw fh]);
134
135
136
137 plot(SAMP/nFULL*100,M0,'.','color',[0,0,0]+.6)
138 plot(samples/nFULL*100,CT(:,1),' g','Linewidth',2)
139 plot(samples/nFULL*100,CT(:,2),' g','Linewidth',2)
140 plot(samples/nFULL*100,M0_FULL,' r','LineWidth',3)
141 plot(samples/nFULL*100,M0_FULL+M0_FSE,'  r','LineWidth',1.5)
142 plot(samples/nFULL*100,M0_FULL M0_FSE,'  r','LineWidth',1.5)
143 xlim([0,100])
144 xlabel('\% of Data Used for Model Construction','Interpreter','latex')
145 ylabel('$M_0$ [m s\textsuperscript{ 1} GPa\textsuperscript{ 1}]','Interpreter'
    ,'latex')
146 grid on;
147 ylim([max(M0_FULL) 15*max(M0_FSE),max(M0_FULL)+15*max(M0_FSE)])
148 MatlabFigure2Latex('ValidateM0.png',2,'png',false)
149
150 %% Plot Activation Energy
151
152 for i=1:1:length(samples)
153     xpos = samples(i);
154     subVal = U(SAMP==xpos);
155     CT(i,1) = prctile(subVal,5);
```

```
156     CT(i,2) = prctile(subVal,95);
157 end
158 
159 h=figure(3);clf;hold on;
160 set(h, 'WindowStyle', 'normal');
161 set(h, 'Units', 'centimeters');
162 set(h, 'PaperUnits', 'centimeters');
163 set(h, 'PaperSize', [fw fh]);
164 set(h, 'Position',[(fx+2*fx_step) (fy+0*fy_step) fw fh]);
165 
166 
167 
168 plot(SAMP/nFULL*100,U,'.','color',[0,0,0]+.6)
169 plot(samples/nFULL*100,CT(:,1),' g','Linewidth',2)
170 plot(samples/nFULL*100,CT(:,2),' g','Linewidth',2)
171 plot(samples/nFULL*100,U_FULL,' r','LineWidth',3)
172 plot(samples/nFULL*100,U_FULL+U_FSE,'  r','LineWidth',1.5)
173 plot(samples/nFULL*100,U_FULL U_FSE,'  r','LineWidth',1.5)
174 xlim([0,100])
175 xlabel('\% of Data Used for Model Construction','Interpreter','latex')
176 ylabel('$U^*$ [eV]','Interpreter','latex')
177 grid on;
178 ylim([max(U_FULL) 15*max(U_FSE),max(U_FULL)+15*max(U_FSE)])
179 MatlabFigure2Latex('ValidateU.png',3,'png',false)
180 
181 %% Plot Activation Volume
182 
183 for i=1:1:length(samples)
184     xpos = samples(i);
185     subVal = V(SAMP==xpos);
186     CT(i,1) = prctile(subVal,5);
187     CT(i,2) = prctile(subVal,95);
188 end
```

```
189
190 h=figure(4);clf;hold on;
191 set(h, 'WindowStyle', 'normal');
192 set(h, 'Units', 'centimeters');
193 set(h, 'PaperUnits', 'centimeters');
194 set(h, 'PaperSize', [fw fh]);
195 set(h, 'Position',[(fx+3*fx_step) (fy+0*fy_step) fw fh]);
196
197
198
199 plot(SAMP/nFULL*100,V,'.','color',[0,0,0]+.6)
200 plot(samples/nFULL*100,CT(:,1),' g','Linewidth',2)
201 plot(samples/nFULL*100,CT(:,2),' g','Linewidth',2)
202 plot(samples/nFULL*100,V_FULL,' r','LineWidth',3)
203 plot(samples/nFULL*100,V_FULL+V_FSE,'  r','LineWidth',1.5)
204 plot(samples/nFULL*100,V_FULL V_FSE,'  r','LineWidth',1.5)
205 xlim([0,100])
206 xlabel('\% of Data Used for Model Construction','Interpreter','latex')
207 ylabel('Activation Volume ($V^*$) [${\AA}^3$]','Interpreter','latex')
208 grid on;
209 ylim([max(V_FULL) 15*max(V_FSE),max(V_FULL)+15*max(V_FSE)])
210 MatlabFigure2Latex('ValidateV.png',4,'png',false)
211
212
213 %% Plot Vol 12
214
215 for i=1:1:length(samples)
216     xpos = samples(i);
217     subVal = V12(SAMP==xpos);
218     CT(i,1) = prctile(subVal,5);
219     CT(i,2) = prctile(subVal,95);
220 end
221
```

```
h=figure(5);clf;hold on;
set(h, 'WindowStyle', 'normal');
set(h, 'Units', 'centimeters');
set(h, 'PaperUnits', 'centimeters');
set(h, 'PaperSize', [fw fh]);
set(h, 'Position',[(fx+4*fx_step) (fy+0*fy_step) fw fh]);



plot(SAMP/nFULL*100,V12,'.','color',[0,0,0]+.6)
plot(samples/nFULL*100,CT(:,1),' g','Linewidth',2)
plot(samples/nFULL*100,CT(:,2),' g','Linewidth',2)
plot(samples/nFULL*100,V12_FULL,' r','LineWidth',3)
plot(samples/nFULL*100,V12_FULL+V12_FSE,'  r','LineWidth',1.5)
plot(samples/nFULL*100,V12_FULL V12_FSE,'  r','LineWidth',1.5)
xlim([0,100])
xlabel('\% of Data Used for Model Construction','Interpreter','latex')
ylabel('Activation Volume Deviator ($\Lambda^*_{12}$) [${\AA}^3$]','
    Interpreter','latex')
grid on;
ylim([max(V12_FULL) 15*max(V12_FSE),max(V12_FULL)+15*max(V12_FSE)])
MatlabFigure2Latex('ValidateV12.png',5,'png',false)

%% Plot Vol 23

for i=1:1:length(samples)
    xpos = samples(i);
    subVal = V23(SAMP==xpos);
    CT(i,1) = prctile(subVal,5);
    CT(i,2) = prctile(subVal,95);
end

h=figure(6);clf;hold on;
```

```
set(h, 'WindowStyle', 'normal');
set(h, 'Units', 'centimeters');
set(h, 'PaperUnits', 'centimeters');
set(h, 'PaperSize', [fw fh]);
set(h, 'Position',[(fx+5*fx_step) (fy+0*fy_step) fw fh]);



plot(SAMP/nFULL*100,V23,'.','color',[0,0,0]+.6)
plot(samples/nFULL*100,CT(:,1),' g','Linewidth',2)
plot(samples/nFULL*100,CT(:,2),' g','Linewidth',2)
plot(samples/nFULL*100,V23_FULL,' r','LineWidth',3)
plot(samples/nFULL*100,V23_FULL+V23_FSE,'  r','LineWidth',1.5)
plot(samples/nFULL*100,V23_FULL V23_FSE,'  r','LineWidth',1.5)
xlim([0,100])
xlabel('\% of Data Used for Model Construction','Interpreter','latex')
ylabel('Activation Volume Deviator ($\Lambda^*_{23}$) [${\AA}^3$]','
    Interpreter','latex')
grid on;
ylim([max(V23_FULL) 15*max(V23_FSE),max(V23_FULL)+15*max(V23_FSE)])
MatlabFigure2Latex('ValidateV23.png',6,'png',false)
```

F.2 Model Builder Script

```
function [ STATISTIC ] = ValidateModel( structure, SubSet_Val ,TYPE)
%VALIDATEMODEL Summary of this function goes here
%   Detailed explanation goes here

inData = SubSet_Val(:,[9,3,6,7,11]); %These are the critial parameters in
    order
%This is only true because SXX,SYY,SZZ are equal. This must be corrected if
%generalized triaxial motion is desired

```

```
inData.Properties.VariableNames{1} = 'T'; %Temperature
inData.Properties.VariableUnits{1} = 'K';
inData.Properties.VariableNames{2} = 'P'; %Pressure  Note: this is only true if SXX=SYY=SZZ
inData.Properties.VariableUnits{2} = 'Bar';
inData.Properties.VariableNames{3} = 'SXY';
inData.Properties.VariableUnits{3} = 'Bar';
inData.Properties.VariableNames{4} = 'SYZ';
inData.Properties.VariableUnits{4} = 'Bar';
inData.Properties.VariableNames{5} = 'Mobility';
inData.Properties.VariableUnits{5} = 'Angstroms atom/(ps eV)';

%% Units conversion
%Pressure Units
inData.P = inData.P*100000; %Convert pressure from Bar to Pa
inData.Properties.VariableUnits{2} = 'Pa';
%Shear stress Units
inData.SXY = inData.SXY*100000; %Convert shearXY from Bar to Pa
inData.Properties.VariableUnits{2} = 'Pa';
inData.SYZ = inData.SYZ*100000; %Convert shearYZ from Bar to Pa
inData.Properties.VariableUnits{2} = 'Pa';

%%% Weird crap %%%%
%Pressure Units
inData.P = inData.P/160.21766e9; %Convert pressure from Pa to eV/Ang^3
%Shear stress Units
inData.SXY = inData.SXY/160.21766e9; %Convert shearXY from Pa to eV/Ang^3
inData.Properties.VariableUnits{2} = 'eV/Ang^3';
inData.SYZ = inData.SYZ/160.21766e9; %Convert shearYZ from Pa to eV/Ang^3
inData.Properties.VariableUnits{2} = 'eV/Ang^3';


%% Mobility Units
```

```
lat = 4.032; %Lattice constant in angstroms
FCCatoms = 4; %Atoms per unit cell FCC: 6 face atoms (1/2each) = 3, 8 corner
    atoms (1/8 each)=1
eV2GPa = 160.21766280; %Convert eV/Ang^3 to GPa
aPs2mPs = 100; %Convert angstroms per seconds to meters per second
convert = (lat^3/FCCatoms)*(aPs2mPs/eV2GPa);

inData.Mobility = inData.Mobility*convert; %Make the conversion in units to
    mobility
inData.Properties.VariableUnits{5} = 'm/(s GPa)';

%% Boltzmann constant

k = 8.6173303e 5;       %Boltzmann constant eV / K



%% Model
mFun = @(C,X) C(1)*...
     exp( C(2)./(k*X(:,1))).*...
     exp( C(3).*X(:,2)./(k*X(:,1))).*...
     exp(2*C(4).*(X(:,3))./(k*X(:,1))).*...
     exp(2*C(5).*abs(X(:,4))./(k*X(:,1)));

Coeff = structure.mCoefficients.Estimate(:);
X_Param = inData{:,1:4};
MODELED = mFun(Coeff,X_Param);
ACTUAL = inData{:,5};

RMSE = sqrt(mean((MODELED   ACTUAL).^2));  % Root Mean Squared Error;
RESID = mean(abs(MODELED ACTUAL));
SStot = sum((ACTUAL mean(ACTUAL)).^2);
SSres = sum((ACTUAL MODELED).^2);
```

```
RSQ = 1 SSres/SStot;

switch TYPE
    case 'RMSE'
        STATISTIC = RMSE;
    case 'Resid'
        STATISTIC = RESID;
    case 'Rsq'
        STATISTIC = RSQ;
end

end
```