Chapter 1

Methods

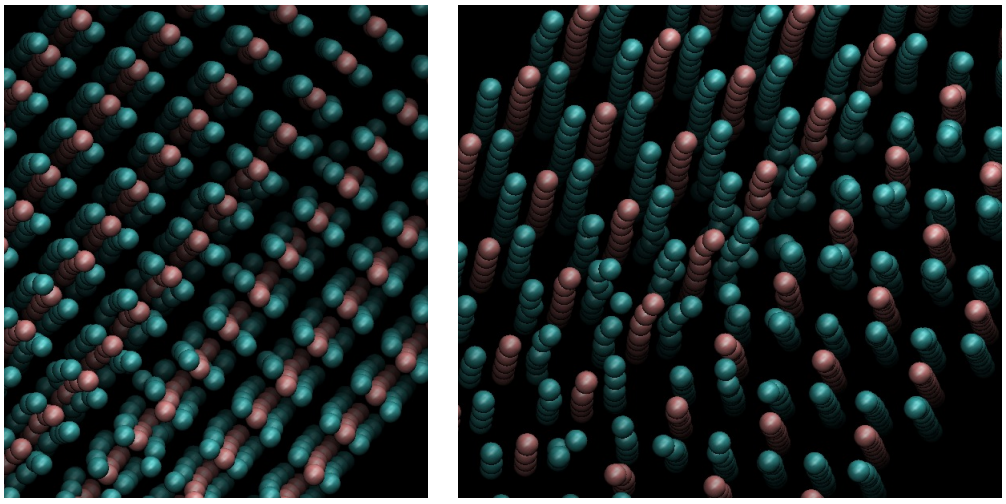
Sufficient data is a requirement for any fitting procedure, but unfortunately there is a lack of data on uranium dioxide (UO_2) grain boundary (GB) energies in the literature. As a work-around, this work used molecular dynamics (MD) simulations[1,2] as fitting data to calculate the GB energies of various lattices based on the coincident site lattice (CSL) model. This model is built off of the idea that GB energy is low when more lattice sites are in coincidence. A number defined as the Σ-number describes the number of coincident sites per total number of lattice sites in a given unit cell of a crystal.[3,4] This work developed a MATLAB® script using Bulatov *et al.*'s methods[5] and building off of Harbison's[6] script to fit parameters to the gathered data. A reduced chi-square statistic was calculated to determine the effectiveness of the fit.

1.1 Molecular Dynamics

Simulation results from the Large-scale Atomic/Molecular Massively Parallel Simulation (LAMMPS) software (developed at Sandia National Laboratory[7]) were gathered for a number of twist, tilt, and mixed GBs. These calculations were performed by simulating two crystals of UO_2 and placing them together in various orientations. A GB is introduced at the interface, creating GB energy. The energy of the system is calculated from the interatomic forces inside the crystal. That energy is compared to the energy of a single grain (of the same size as the combined two grains) of UO_2 to determine the energy at the GB.[6] The GB energy is calculated as:[8]

$$E_{\mathrm{GB}} = \frac{|E_{\text{single grain}} - E_{\text{two grains}}|}{2 * A_{\mathrm{GB}}}. \tag{1.1}$$

An example of how the atoms align is shown in Figure 1.1. Harbison's original calculations[6] were done using no anneal (maximum temperature was approximately 0 K), only allowing the atoms to relax to their local minima. This work used an anneal of 800 K, allowing the atoms to relax to a better estimate of their global minimum value as is shown in **??**. This work used the same misorientation angles for the GB energy calculations that Harbison used. The fitting procedure uses these energies to produce parameters describing the five-dimensional GB space.



(a) (b)

Figure 1.1: These figures demonstrate example crystal structures of UO_2 after an annealing process. The better the atoms line up, the lower the energy is. (a) shows an example of a mostly aligned GB, indicative of a lower energy. (b) shows an example of a misaligned GB, indicative of a higher energy. These two images are from a $\langle 111 \rangle$ twist image. Different viewpoints show different amounts of alignment. The LAMMPS simulation package takes care of all the calculations to determine the energy at these GBs. Images courtesy of Dr. Evan Hansen, used with permission.

1.2 Bulatov *et al.*'s Methods

This work implemented Bulatov *et al.*'s hierarchical interpolation method to find the energy of an arbitrary GB in the five-space.[5] They chose three three-dimensional (3D) axes with at least two-fold symmetry (called high-symmetry axes) to use as scaffolding to build the entire five-dimensional (5D) function. The axes chosen for both Bulatov *et al.* and this work were the $\langle 100 \rangle$, $\langle 110 \rangle$, and the $\langle 111 \rangle$ sets for their four-, two-, and three-fold rotational symmetries respectively.* Each 3D subset is built from an interpolation of its own one- and two-dimensional subsets. The symmetric tilt and twist GBs for each set were fitted first because of their simplicity. Only the rotation angle is needed to fully define the energies for these subsets, making them one-dimensional (in Figure 1.2a, the darker bands in the smaller circles). From the symmetric tilt subset, the asymmetric, or general, tilt subset was interpolated. A second rotation angle defining the rotation of the second grain makes this subset two-dimensional (the lighter, wider band of color around the symmetric subset). A combination of the general tilt subset (two dimensions) and the twist subset (one dimension) was used to interpolate the 3D subset for each high-symmetry axis (the three smaller circles). These three 3D subsets were then used to interpolate the GB 5D space.

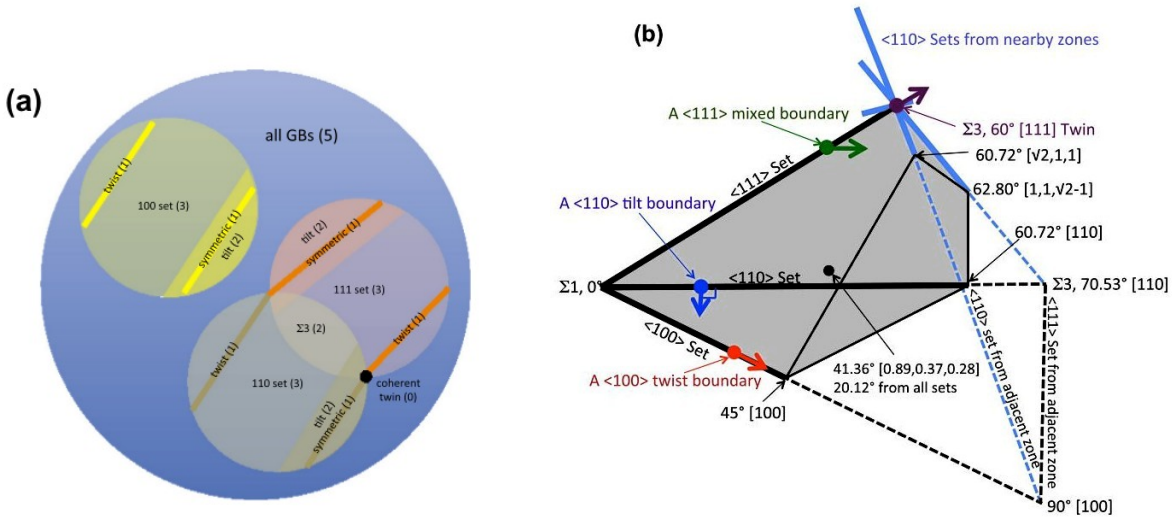
Bulatov *et al.* and this work used the Read-Shockley-Wolf (RSW) functions,[10] which

*For cubic crystals, rotations of 90°, 180°, or 120° about any $\langle 100 \rangle$, $\langle 110 \rangle$, or $\langle 111 \rangle$ axis respectively is a symmetry operation.[9] Thus, the $\langle 100 \rangle$ set is four-fold symmetric (360°/90°= 4), the $\langle 110 \rangle$ set is two-fold symmetric (360°/180°= 2), and the $\langle 111 \rangle$ set is three-fold symmetric (360°/120°= 3).

take the form:

$$E_{min} + (E_{max} - E_{min}) \sin\left(\frac{\pi}{2}\frac{\theta - \theta_{min}}{\theta_{max} - \theta_{min}}\right)\left(1 - a\log\left(\sin\left(\frac{\pi}{2}\frac{\theta - \theta_{min}}{\theta_{max} - \theta_{min}}\right)\right)\right), \quad (1.2)$$

where θ is the misorientation angle, θ_{min} is the minimum angle on the domain, θ_{max} is the maximum angle on the domain, and a is a shaping parameter. Each RSW function covers a "low-angle" subset (around 15°, with higher angles being less accurate)[4,10] of the domain in the 1D GB space. An example of a simple RSW function is shown in Figure 1.3. Multiple RSW functions are stitched together to form the 1D subsets.



(a) (b)

Figure 1.2: Figure 2 from Bulatov *et al.*[5] (a) demonstrates the theoretical relationship between the high-symmetry subsets of the 5D GB space. Each multi-dimensional subset is interpolated from smaller-dimensional subsets. (b) shows the Rodrigues space representation of the fundamental zone of all GBs as built from three high-symmetry axes ($\langle 100 \rangle$, $\langle 110 \rangle$, and $\langle 111 \rangle$). The unit vectors along the axis identify the boundary plane inclination in the frame of grain one. A parallel vector thus represents a twist boundary, a perpendicular vector represents a tilt boundary, and neither parallel nor perpendicular vectors represent a mixed boundary. The full misorientation space has 1152 symmetrically equivalent copies of the fundmental zone[5,11] which allows the infinite nature of Rodrigues space to be accounted for.

1.3 Representations of Grain Boundary Space

Part of Bulatov *et al.*'s development of their 5D function was accomplished through visual representations of the GB space. However, the size of the five-space in which GBs reside makes representing them difficult. Different methods have been developed to represent them, each with their advantages and disadvantages. Three of these methods are the axis-angle representation, the Rodrigues representation, and the fundamental zone representation. These

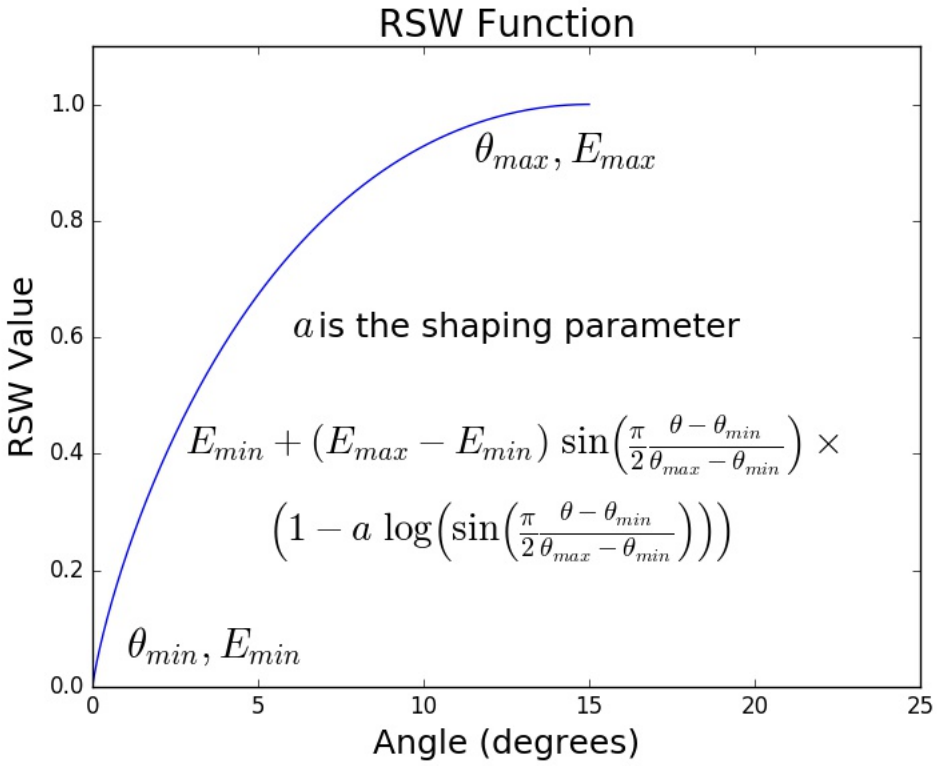


Figure 1.3: An example of an RSW function with $\theta_{min} = 0°$, $\theta_{max} = 15°$, and a (the shaping parameter) $= 0.5$ Combining these functions into a Piecewise set over a given domain gives the GB energy curves their distinct, cusp-like behavior. The RSW functions are scaled using E_{min} and E_{max}. In this example, $E_{min} = 0$ and $E_{max} = 1$.

methods, though described separately, can be used together to form a better picture of what the GB space looks like (see for example Figure 1.2b which combines the Rodrigues representation and the fundamental zone representation).

1.3.1 Axis-Angle Representation

The axis-angle representation is the simplest of the three described here. The axis of rotation of the GB specifies the point in axis-angle space, and the angle of misorientation between the two grains at the GB specifies the magnitude of the vector. Thus, the axis ($\boldsymbol{a}$, where $\boldsymbol{a}$ has components a_x, a_y, and a_z) and the angle (θ) mathematically represent an axis-angle vector as:

$$\boldsymbol{A} = \boldsymbol{a}\ \theta \tag{1.3}$$

The axis-angle space can only take into account three degrees of freedom: the two angles specifying the axis, and the angle rotated through. Thus, axis-angle space cannot fully visualize all of the necessary information contained in the full 5D space.[12] An added difficulty of using this representation is its infinite space because it is a mapping of an axis and an angle onto a Cartesian coordinate system. This mapping means understanding the entire GB space is difficult without the help of additional methods. The axis-angle representation is

best used as a starting point to move to other, more robust representations, and to represent the misorientation between two grains.[11]

1.3.2 Rodrigues Representation

The Rodrigues representation (sometimes called the "Rodrigues-Frank" representation) uses Rodrigues vectors to represent rotations in Rodrigues space. This representation takes ideas from the axis-angle space, but makes a few changes allowing crystal symmetries to be taken into account. The axis about which a GB is oriented still specifies the point in space, but the tangent of half the angle represents the magnitude of the vector. Thus, a Rodrigues vector can be represented as:[11–15]

$$\boldsymbol{R} = \boldsymbol{a} \tan\left(\frac{\theta}{2}\right) \tag{1.4}$$

Some researchers favor this representation over others because of the lack of curvature such a mapping entails.[11,12] However, still only three of the five degrees of freedom are specified. Bulatov *et al.* attached a unit vector at the points along the axis to represent the other two in Figure 1.2b. A parallel vector represents a twist boundary, and a perpendicular vector represents a tilt boundary. Anything else represents a mix of twist and tilt (or a mixed boundary). One difficulty in using Rodrigues space is it also is an infinite space, as it also simply maps an axis and an angle onto a Cartesian coordinate system.[12,16]

1.3.3 Fundamental Zone Representation

The fundamental zone is perhaps the best graphical representation for the 5D GB. This representation takes advantage of the symmetries inherent in crystals[9] to simplify an infinite space into a compact, finite area called the fundamental zone.[5,13,17–19] Every point within the space represents a unique orientation, and every point outside the space can be represented as a point inside the space through symmetry operations.[12–14] Bulatov *et al.* used this idea in connection with Rodrigues space to create Figure 1.2b. In Rodrigues space, the crystal symmetric of the material determine the shape of the fundamental zone.[13,17] For fcc crystals, the fundamental zone takes the form of a truncated tetrahedron.[5] The edges of the fundamental zone in Rodrigues space represent the high-symmetry rotation axes, and points on one face can represent another point on a different face of the fundamental zone.

1.4 Code Analysis

Harbison[6] and Bulatov *et al.*[5] developed MATLAB® scripts for their work. This work analyzed these codes and used the ideas from them to develop the code that generated these parameters.

1.4.1 The Fitting Code

This work performed an extensive analysis of Harbison's[6] code to learn how it works and implement the ideas. The basic outline for the fitting procedure follows. First, data is read

in from a database containing energies associated with either a twist or tilt GB on one of the three high-symmetry axes. Test parameters which are used as starting points for the fitted parameters are also read in from a separate database. The one-dimensional sets are fitted first because of their simplicity. The parameters found from the 1D fits are used in fitting the higher-dimensional sets. Important angles are specified where low energies are expected, such as the $\Sigma 5$ boundary for the $\langle 100 \rangle$ symmetric tilt subset. These angles are calculated based on the Σ-number from CSL theory. Because the Σ-number designates how many lattice points are between each coincident site (and assuming that the space between each lattice site, the lattice constant, is known) the angle of the GB misorientation can be determined. Every energy in the parameter-vector is listed as a scaled value based on the e_{RGB} parameter to minimize the potential for error in calculations. The e_{RGB} parameter represents the energy of an arbitrary, random GB, and can be seen as an average of the material's GB energies. Thus, to make relevant comparisons, the energies are unscaled based on the units of energy desired (typically J/m^2). All of the parameters and the angle-energy pairs from the database are then passed into a grid-search fitting function. This work gave each subset a different initial step size to avoid a numerical error where the steps would take the angles currently being looked at outside of their domain. Without this, the grid-search procedure would not return the correct amount of values, preventing the code from running to completion.

Once the six one-dimensional subsets and the three two-dimensional subsets are fitted, the three-dimensional subsets are fitted using the twist and asymmetric tilt subsets to calculate the mixing parameters (defining the relationship between the twist and general tilt subsets within a high-symmetry axis - i.e. the relationship between the small dark bands representing the twist boundaries and the lighter, wider bands representing the tilt boundaries in Figure 1.2a). The final step is to calculate the weighting parameters, which defines the relationship between the three high-symmetry subsets. Equations defining the various relationships can be found in Bulatov *et al.*'s work.

1.4.2 The Energy Calculation Code

Bulatov *et al.*'s open-source MATLAB® code,[5] GB5DOF.m, calculates the energy of an arbitrary GB in certain fcc metals. This work uses this script for calculating an arbitrary GB in uranium dioxide, and proceeds as follows. First, metrics defining the "distance" between the GB and all three high-symmetry axes are calculated. These distances are calculated by looking at all symmetrically equivalent representations of a GB on a per-axis basis (for cubic crystals, there are 24 equivalent representations[9]). Because there are three, six, and four unique axes for the $\langle 100 \rangle$, $\langle 110 \rangle$, and $\langle 111 \rangle$ axes respectively, a maximum of $6 \times 24 = 144$ distances are calculated. Any distances exceeding a predefined cutoff distance are discarded. After all distances have been calculated, only the unique representations are kept to avoid double-counting certain representations.[5] Energies are calculated for each unique distance in each subset. These energies are then weighted and summed to give the interpolated energy for the specified GB.

1.5 Reduced Chi-Square Statistic

A good way to test how well a function fits the data is to use a reduced chi-square goodness-of-fit statistic.[20] The orientation matrices (which Bulatov *et al.* calls the P and Q matrices for the first and second grains respectively) were needed as input parameters to Bulatov *et al.*'s function to calculate this statistic. These three by three matrices specify the orientation in a lab frame of the two grains individually. A good fit will have a reduced chi-square value close to one, while those values greater than one indicate an under fit, and those values less than one indicate an over fit.[20]

1.5.1 Developing the P and Q Matrices

Creating the P and Q matrices was a non-trivial task. Because there has been so much work done with crystallography over the past few decades, many different methods have been developed to specify the orientation matrices of grains. A rotation matrix also needed to be calculated which rotates the axis of rotation to the [100] direction, as Bulatov *et al.*'s energy calculation code assumes. Three methods, following the method prescribed in MARMOT, using the Rodrigues rotation formula, and using the Bunge rotation matrix, were used in this work in the process of developing these matrices and are described below.

MARMOT Method

MARMOT, Idaho National Laboratory's (INL's) mesoscale phase-field modeling platform,[21] calculates the P and Q matrices for the grains using Euler angles as input parameters. The Euler angles are converted to the orientation matrices using the Bunge convention, i.e. the ZXZ or $ZX'Z''$ rotation, where the first rotation is about the z axis, the second rotation is around the new x axis, and the final rotation is about the new z axis. The formula for this conversion from Bunge Euler angles to the rotation matrix is found by multiplying the z, x, and z rotation matrices together in that order to get:

$$\begin{bmatrix} c_1\ c_3 - c_2\ s_1\ s_3 & -c_1\ s_3 - c_2\ c_3\ s_1 & s_1\ s_2 \\ c_3\ s_1 + c_1\ c_2\ s_3 & c_1\ c_2\ c_3 - s_1\ s_3 & -c_1\ s_2 \\ s_2\ s_3 & c_3\ s_2 & c_2 \end{bmatrix} \tag{1.5}$$

where c_n and s_n represent the cosine and sine of the respective angles (1 represents the first z rotation, 2 represents the x rotation, and 3 represents the second z rotation. These angles are usually referred to as[11] φ_1, Φ, φ_2).

The rotation matrix is calculated in MARMOT by using the GB normal and finding the rotation matrix required to rotate that vector to the [100] direction. In MARMOT, simulations are set up through input files. In the input files there are different sections (called blocks) specifying material parameters, boundary conditions, initial conditions, and the physical models to use to solve the problem (among others). The initial condition used to calculate the rotation matrices in MARMOT for this set of problems was a horizontal or vertical line for tilt or twist boundaries respectively. Because of this set up, the GB normals were either along the [010] axis for the tilt boundaries, or $[\bar{1}00]$ for twist boundaries.

Rodrigues Rotation Formula

The Rodrigues rotation formula[22] (RRF) is a way of calculating the rotation matrices given an axis and an angle using the following formula:

$$\boldsymbol{R} = \boldsymbol{I} + \sin\,\theta\;\boldsymbol{K} + (1 - \cos\,\theta)\;\boldsymbol{K}^2, \tag{1.6}$$

where $\boldsymbol{I}$ is the 3x3 identity matrix, θ is the angle rotated through, and $\boldsymbol{K}$ is the skew-symmetric matrix formed by the axis of rotation ($\boldsymbol{a}$) by:

$$\begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \tag{1.7}$$

This work calculated the rotation matrices two different ways with this orientation matrix formulation. The first method simply used the MARMOT-generated rotation matrices. A second method calculated the rotation matrices using geometric arguments (see Figure 1.4). From these arguments the normals are identified in Table 1.1.

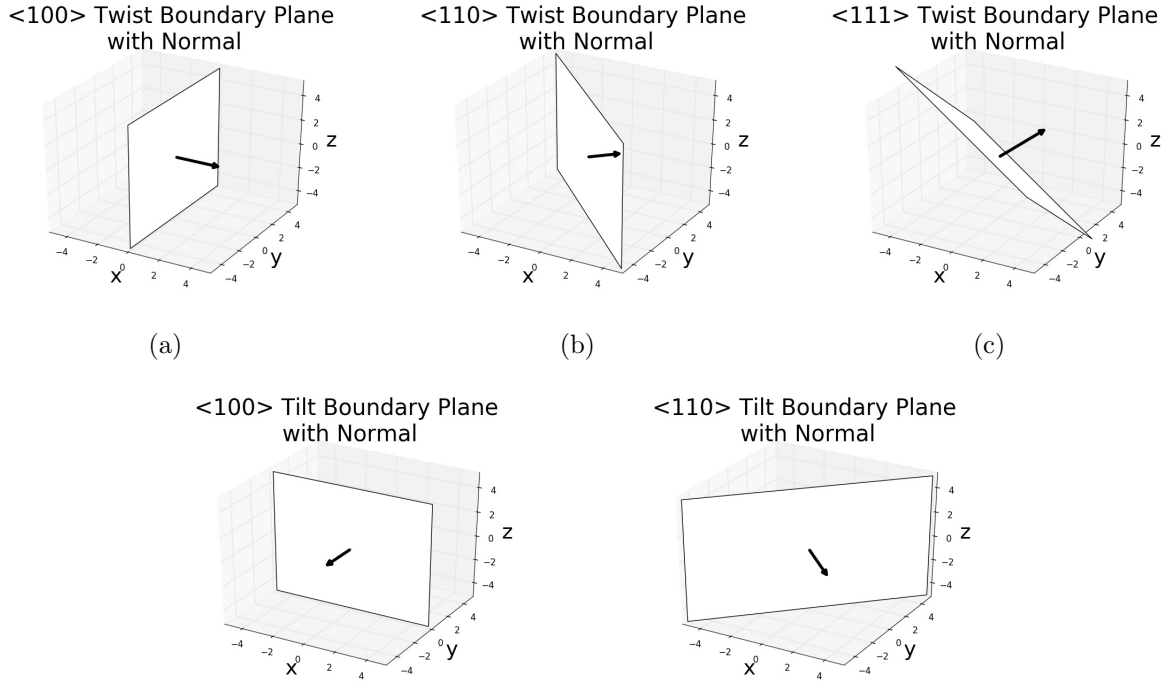


Figure 1.4: A geometric method of determining the normals of a GB. (a) to (c) show the GB normals for a GB perpendicular to the axis of rotation (a twist GB). The GB normal is simply the axis about which the grains are rotated. (d) and (e) show the GB normals for a GB parallel to the axis of rotation (a tilt GB). The GB normal is perpendicular to the axis of rotation. The same GB normal for $\langle 110 \rangle$ Tilt can be used for $\langle 111 \rangle$ Tilt.

Table 1.1: Table of GB normals for different GB types. The normalized dot product of the axis with the GB normal is zero in all tilt cases and one in all twist cases. There are two options for the grain boundary normals of each subset because of inversion symmetries.

Axis	Boundary Type	GB Normal
$\langle 100 \rangle$	Tilt	$[010]$ $[0\bar{1}0]$
$\langle 110 \rangle$	Tilt	$[1\bar{1}0]$ $[\bar{1}10]$
$\langle 111 \rangle$	Tilt	$[1\bar{1}0]$ $[\bar{1}10]$
$\langle 100 \rangle$	Twist	$[100]$ $[\bar{1}00]$
$\langle 110 \rangle$	Twist	$[110]$ $[\bar{1}\bar{1}0]$
$\langle 111 \rangle$	Twist	$[111]$ $[\bar{1}\bar{1}\bar{1}]$

Bunge Rotation Matrix

The Bunge rotation matrix (see Equation (1.5)) is what MARMOT uses to create the orientation matrices. This work used various methods to calculate the Euler angles, of which three are briefly described here. The Euler angles (once calculated) were used in the first two methods to calculate the entirety of the rotation matrix.

First, this work tried to use scripts developed to calculate the various Euler angles for MARMOT. These scripts did not work because of the same assumptions made earlier about the orientation of the GB, namely, that all pure tilt GBs have a normal of 010, and that all pure twist GBs have a normal of $\bar{1}00$. The difference between MARMOT's boundary conditions and the boundary conditions of this work is that GBs in this work are assumed to be either perpendicular or parallel to the rotation axis, as opposed to always being along the x or y axis.

The second method used an open-source MATLAB® package called MTEX.[23] This package calculates Euler angles using quaternions. These Euler angles did not generate the correct results either, for the most part creating the same sorts of graphs as the MARMOT method. It is uncertain why this method did not work.

The working method used the mathematics of quaternions directly.[24] The quaternions were calculated based on the misorientation axis and angle. A quaternion is a four-dimensional vector containing one real part, and three imaginary parts. The components of the vector are calculated as follows:

$$\boldsymbol{q} = \left[\cos\left(\frac{\theta}{2}\right),\ a_x \sin\left(\frac{\theta}{2}\right),\ a_y \sin\left(\frac{\theta}{2}\right),\ a_z \sin\left(\frac{\theta}{2}\right)\right], \tag{1.8}$$

with axis $\boldsymbol{a}$ and misorientation angle θ. Once the axis and misorientation angle are converted to a quaternion, another conversion from a quaternion to the Bunge Euler angles is performed. The angles are calculated using the atan2 method which allows for all four

quadrants in the Cartesian space to be accounted for. The angles are calculated using the following formulas:

$$\begin{aligned}
\chi &= \sqrt{(q_0^2 + q_3^2)(q_1^2 + q_2^2)} \\
\varphi_1 &= \text{atan2}\left(\frac{q_0 q_2 + q_1 q_3}{2\chi}, \frac{q_0 q_1 - q_2 q_3}{2\chi}\right) \\
\Phi &= \text{atan2}\left(2\chi, q_0^2 + q_3^2 - q_1^2 - q_2^2\right) \\
\varphi_2 &= \text{atan2}\left(\frac{q_1 q_3 - q_0 q_2}{2\chi}, \frac{q_0 q_1 + q_2 q_3}{2\chi}\right).
\end{aligned} \tag{1.9}$$

Once the Euler angles were calculated, they were put into Equation (1.5), and the resulting matrices were used as the orientation for the grains. The codes used to generate the orientation matrices can be found in **????**.

Testing The Matrices

This work attempted to reproduce the 1D subset graphs as shown in Bulatov *et al.* as a way to test the different methods. Various levels of success were observed for the different methods. The matrices giving the best results are shown in **??????**.

While the first method works well for MARMOT, the challenge accompanying its use was that MARMOT specifies a specific GB normal with the set up of the problem that is not necessarily what the MATLAB® script expects. Thus, the results coming from using this combination of matrices ended up working only for the ⟨100⟩ tilt, ⟨110⟩ tilt, and ⟨100⟩ twist subsets, while the ⟨110⟩ twist had issues with singularities, and the ⟨111⟩ subsets did not remotely match the expected outcome.

1.5.2 Calculating Reduced Chi Squared

There were two methods that this work used to calculate the χ^2_{red} statistic. The first method used the P and Q matrices as developed above to test the entirety of the fit. The second method calculated the statistic for each 1D subset, then calculated the full χ^2_{red} value using the statistics from the subsets. Results from these calculations are discussed in the next chapter.

The test for the entire fit used the P and Q matrices to calculate the energy in 1° intervals for each subset, using Bulatov *et al.*'s GB5DOF.m script. The χ^2_{red} value was calculated for each subset and for the entire fit using Equation (1.10), producing the results in **??** under the 800 K anneal column under the "χ^2_{red} using P and Q matrices" section.

$$\chi^2_{\text{red}} = \frac{1}{N - n - 1} \sum \frac{(\epsilon_{\text{md}} - \epsilon)^2}{e\ \epsilon_{\text{md}}}. \tag{1.10}$$

In this equation, N is the number of observations, n is the number of parameters, ϵ_{md} are the energies from MD, ϵ are the energies from the model, and e is the uncertainty in the MD results.

Using the second method, the same angles used in the fitting procedure were used in the RSW equations creating the 1D subsets. The differences between the values resulting

from there and the MD simulation values lead to the χ^2_{red} values shown in the 800 K anneal column under the χ^2_{red} comparing the 1D fits section. The same methods were implemented to calculate the χ^2_{red} values for the data without an anneal.

The statistic calculated using these methods is different from the χ^2 statistic used in the grid-search function. The grid search used Equation (1.11), and generated values of the same order as Equation (1.10).

$$\chi^2 = \sum (E_{\text{measured}} - E_{\text{calculated}})^2 \tag{1.11}$$

Bibliography

[1] Y. Zhang, *Unpublished*, Personal Communication

[2] E. Hansen, *Unpublished*, Personal Communication

[3] P. Lejček, *Grain Boundaries: Description, Structure and Thermodynamics* (Springer Berlin Heidelberg, Berlin, Heidelberg, 2010), pp. 5–24

[4] G. S. Rohrer, *Grain Boundary Energy Anisotropy: A Review*, J. Mater. Sci. **46** (2011), pp. 5881–5895

[5] V. V. Bulatov, B. W. Reed and M. Kumar, *Grain boundary energy function for fcc metals*, Acta Mater. **65** (2014), pp. 161–175

[6] T. Harbison, *Anisotropic grain boundary energy function for uranium dioxide*, B.S. Thesis, Brigham Young University - Idaho (2015)

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

[8] A. S. Butterfield, *Exploration of the phase-field framework MARMOT to include anisotropic grain boundaries with molecular dynamics*, B.S. Thesis, Brigham Young University - Idaho (2013)

[9] H. T. Stokes, *Solid State Physics: For advanced undergraduate students* (BYU Academic Publishing, Provo, Utah, 2007)

[10] D. Wolf, *A Read–Shockley model for high–angle grain boundaries*, Scripta Metallurgica **23** (1989), pp. 1713–1718

[11] V. Randle and O. Engler, *Introduction to Texture Analysis: Macrotexture, Microtexture and Orientation Mapping* (CRC Press, USA, 2000)

[12] F. C. Frank, *Orientation Mapping*, Metallurgical Transactions A **19A** (1988), pp. 403–408

[13] A. Morawiec and D. P. Field, *Rodrigues parameterization for orientation and misorientation distributions*, Phil. Mag. A **73** (1996), pp. 1113–1130

[14] R. Becker and S. Panchanadeeswaran, *Crystal rotations represented as Rodrigues vectors*, Textures and Microstructures **10** (1989), pp. 167–194

[15] L. Priester, *Geometrical Order of Grain Boundaries* (Springer Netherlands, Dordrecht, 2013), pp. 3–28

[16] D. M. Kirch, *Fundamentals of grain boundaries and triple junctions* (Cuvillier, Göttingen, 2008), pp. 1–7

[17] S. Patala and C. A. Schuh, *Symmetries in the representation of grain boundary-plane distributions*, Philosophical Magazine **93** (2013), pp. 524–573

[18] E. R. Homer, S. Patala and J. L. Priedeman, *Grain boundary plane orientation fundamental zones and structure-property relationships*, Sci. Rep. **5** (2015), p. 15476

[19] S. Patala, J. K. Mason and C. A. Schuh, *Improved representations of misorientation information for grain boundary science and engineering*, Progress in Materials Science **57** (2012), pp. 1383–1425

[20] P. R. Bevington and D. K. Robison, *Data Reduction and Error Analysis for the Physical Sciences* (McGraw–Hill, New York, NY, 2003)

[21] M. R. Tonks, D. Gaston, P. C. Millett, D. Andrs and P. Talbot, *An object-oriented finite element framework for multiphysics phase field simulations*, Computational Materials Science **51** (2012), pp. 20–29

[22] S. Belongie, *Rodrigues Rotation Formula From MathWorld–A Wolfram Web Resource, created by Eric W. Weisstein* (2006), `mathworld.wolfram.com/RodriguesRotationFormula.html`

[23] F. Bachmann, R. Hielscher and H. Schaeben, *Texture Analysis with MTEX – Free and Open Source Software Toolbox*, Solid State Phenomena **160** (2010), pp. 63–68

[24] E. W. Weisstein, *Quaternion From MathWorld–A Wolfram Web Resource* (2004), `mathworld.wolfram.com/Quaternion.html`