Chapter 1

Methods

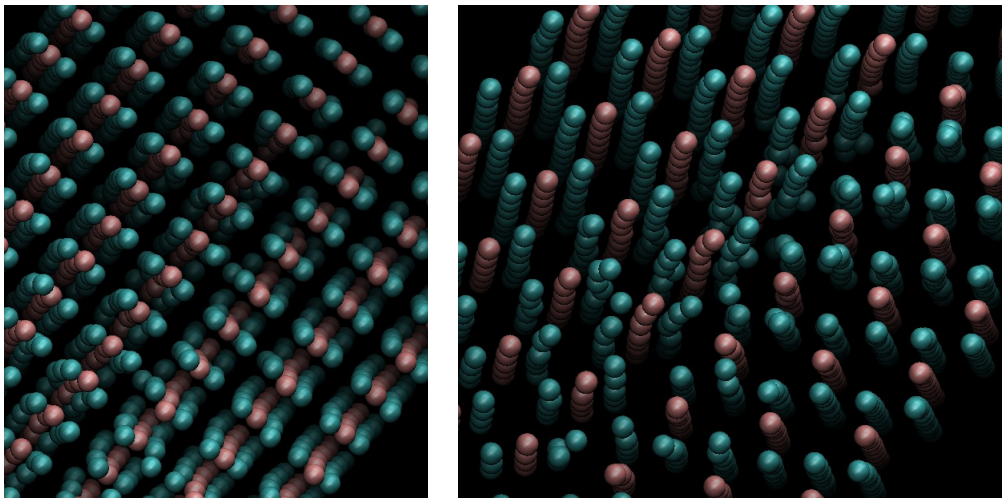
Any fitting procedure requires a sufficient amount of data, but unfortunately such data does not exist for uranium dioxide (UO_2) in the literature. As a work-around, this work used molecular dynamics (MD) simulations[1,2] as fitting data to calculate the GB energies of various lattices based on the coincident site lattice (CSL) model. This model builds off of the idea that the GB energy has lower values when more lattice sites coincide. 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 script[6] to fit parameters to the gathered data. A reduced chi-square statistic determined the effectiveness of the fit.

1.1 Molecular Dynamics

Zhang[1] and Hansen[2] collected simulation results from the Large-scale Atomic/Molecular Massively Parallel Simulation (LAMMPS) software (developed at Sandia National Laboratory[7]) for a number of twist, tilt, and mixed GBs. They performed these calculations by simulating two crystals of UO_2 and placing them together in various orientations. A GB forms at the interface, creating GB energy. Calculating the energy of the system from the interatomic forces inside the crystal, and comparing that energy to the energy of a single grain (of the same size as the combined two grains) determines the energy at the GB.[6] Calculating the GB energy follows the form:[8]

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

Here, E_{GB} represents the energy at the grain boundary, and $E_{\text{single grain}}$ and $E_{\text{two grains}}$ represent the energies of the single and double grains respectively. A_{GB} represents the area of the grain boundary. Figure 1.1 shows an example of how the atoms align. Harbison's original calculations[6] used no anneal ($T_{\max} \approx 0$ 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 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. (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 show results from a $\langle 111 \rangle$ twist simulation. 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

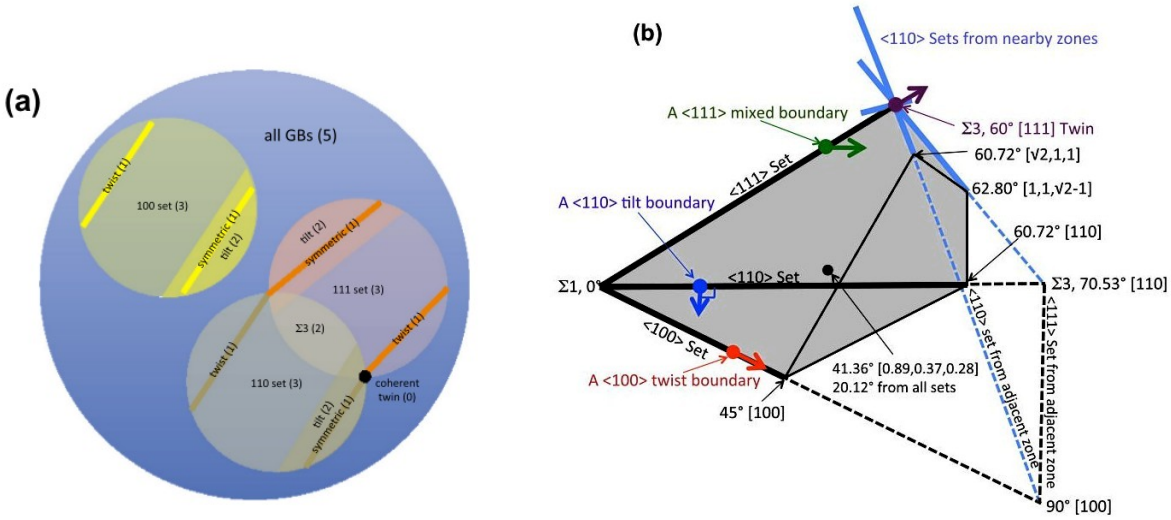
This work implemented Bulatov *et al.*'s hierarchical interpolation method to find the energy of an arbitrary GB in the five-space.[5] They chose three three-dimensional (3D) axes with at least two-fold symmetry (called high-symmetry axes) to use as scaffolding to build the entire five-dimensional (5D) function. Bulatov *et al.* and this work chose 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 builds from 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. The rotation angle fully defines the energies for these subsets, making them one-dimensional (in Figure 1.2a, the darker bands in the smaller circles). Having the parameters from the symmetric tilt subset allows interpolation to the asymmetric, or general, tilt subset. A second rotation angle defining the rotation of the second grain makes this subset two-dimensional (the lighter, wider band around the symmetric subset). A combination of the general tilt (two dimensions) and the twist subsets (one dimension) interpolates the 3D subset for each high-symmetry axis (the three smaller circles). Bulatov *et al.* and this work used these three 3D subsets to interpolate the GB 5D space. Figure 1.2b shows the simplified GB space using the Rodrigues fundamental zone representation. **??** provides a further explanation of Rodrigues space and the fundamental zone.

*Symmetry operations for cubic crystals include rotating by 90°, 180°, or 120° about any $\langle 100 \rangle$, $\langle 110 \rangle$, or $\langle 111 \rangle$ axis respectively.[9] Thus, the $\langle 100 \rangle$ set has four-fold symmetry (360°/90°= 4), the $\langle 110 \rangle$ set has two-fold symmetry (360°/180°= 2), and the $\langle 111 \rangle$ set has three-fold symmetry (360°/120°= 3).

Bulatov *et al.* and this work used the Read-Shockley-Wolf (RSW) functions,[10] which 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} and θ_{max} represent the minimum and maximum angles on the domain respectively, a is a shaping parameter, and E_{min} and E_{max} represent the energy at θ_{min} and θ_{max} respectively. Each RSW function covers a "low-angle" subset (around 15°, with larger domains being less accurate)[4,10] of the domain in the 1D GB space. Figure 1.3 shows an example of a simple RSW function. Stitching together multiple RSW functions forms 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 interpolates 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.

1.3 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 the parameters listed in **??**.

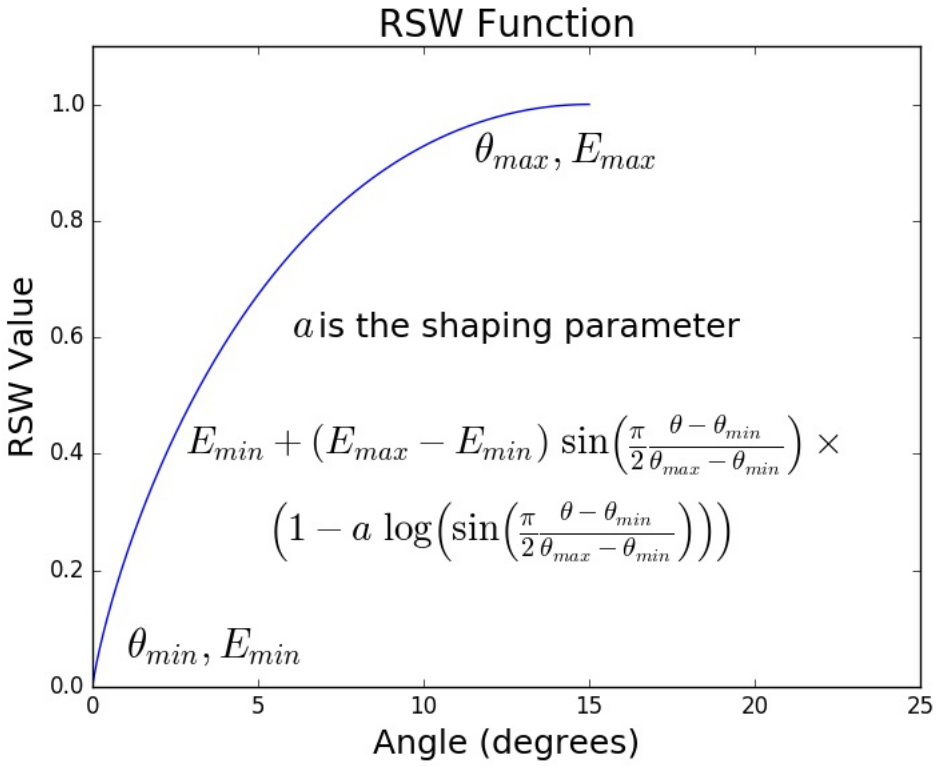


Figure 1.3: An example of an RSW function with $\theta_{min} = 0°$, $\theta_{max} = 15°$, and a (the shaping parameter) $= 0.5$. Combining these functions into a Piecewise set over a given domain gives the GB energy curves their distinct, cusp-like behavior. The RSW functions scale based on E_{min} and E_{max}. In this example, $E_{min} = 0$ and $E_{max} = 1$. Note that E_{min} and E_{max} do not represent the lowest and highest energies in the domain, but rather represent the energy at θ_{min} and θ_{max} respectively, meaning that the value of E_{min} could be higher than the value of E_{max}.

1.3.1 The Fitting Code

This work performed an extensive analysis of Harbison's code to learn how it works and to implement the ideas therein. The basic outline for the fitting procedure follows. First, a database containing energies associated with either a twist or tilt GB on one of the three high-symmetry axes provides the fitting data. A separate database provides the test parameters which define starting points for the fitted parameters. The parameters found from the 1D fits assist in fitting the higher-dimensional sets. Important angles specify where to expect low energies, such as the $\Sigma 5$ boundary for the $\langle 100 \rangle$ symmetric tilt subset. The *Sigma*-number from CSL theory determines the angles. Because the Σ-number designates the number of lattice points between each coincident site (and assuming the separation distance between each lattice site, or the lattice constant is known) the angle of the GB misorientation can be determined. A value known as the e_{RGB} parameter scales every energy in the parameter vector to minimize the potential for error in calculations. The e_{RGB} parameter represents the energy of an arbitrary, random GB, and represents an average of the material's GB energies.

Thus, making relevant comparisons requires unscaling the energies based on the units of energy desired (typically J/m^2). All of the parameters and the angle-energy pairs from the database get 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.

After fitting the six one-dimensional subsets and the three two-dimensional subsets, interpolating the twist (1D) and asymmetric tilt (2D) subsets calculates the mixing parameters to fit the three-dimensional subsets. The mixing parameters define 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 calculates 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.[5]

1.3.2 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 UO_2, and proceeds as follows. First, it compares all symmetrically equivalent representations of a GB (on a per-axis basis) to calculate metrics defining the "distance" between the GB and all three high-symmetry axes (for cubic crystals, there are 24 equivalent representations[9]). Because of the three, six, and four unique axes for the $\langle 100 \rangle$, $\langle 110 \rangle$, and $\langle 111 \rangle$ axes respectively, the script calculates a maximum of $6 \times 24 = 144$ distances, afterwards discarding any that exceed a predefined cutoff distance. After calculating all distances, the script keeps only the unique representations to avoid double-counting.[5] It then calculates energies for each unique distance in each subset, then weights and sums them to give the interpolated energy for the specified GB.

1.4 Reduced Chi-Square Statistic

A good way to test how well a function fits the data uses a reduced chi-square goodness-of-fit statistic.[11] Bulatov *et al.*'s function required the orientation matrices (which Bulatov *et al.* calls the P and Q matrices for the first and second grains respectively) input parameters 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.[11]

1.4.1 Developing the P and Q Matrices

This work created the P and Q matrices. Because of the vast quantity of work done with crystallography over the past few decades, many different methods can 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. This work used three methods, following the method prescribed in MARMOT, using the Rodrigues rotation formula, and using the Bunge rotation matrix, in the process of developing these matrices.

MARMOT Method

MARMOT, Idaho National Laboratory's (INL's) mesoscale phase-field modeling platform,[12] calculates the P and Q matrices for the grains using Euler angles as input parameters. MARMOT uses the Bunge convention to convert the Euler angles to orientation matrices. The Bunge convention uses the ZXZ or $ZX'Z''$ rotation set, which rotates first about the z axis, then the rotated x axis, and finally about the rotated z axis. Multiplying the z, x and z rotation matrices together in that order generates the formula to convert from Bunge Euler angles to the rotation matrix:

$$\begin{bmatrix} c_1\ c_3 - c_2\ s_1\ s_3 & -c_1\ s_3 - c_2\ c_3\ s_1 & s_1\ s_2 \\ c_3\ s_1 + c_1\ c_2\ s_3 & c_1\ c_2\ c_3 - s_1\ s_3 & -c_1\ s_2 \\ s_2\ s_3 & c_3\ s_2 & c_2 \end{bmatrix} \tag{1.3}$$

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, usually referred to as[13] φ_1, Φ, φ_2).

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

Rodrigues Rotation Formula

The Rodrigues rotation formula[14] (RRF) calculates the rotation matrices given an axis and an angle using the following formula:

$$\boldsymbol{R} = \boldsymbol{I} + \sin\,\theta\ \boldsymbol{K} + (1 - \cos\,\theta)\ \boldsymbol{K}^2, \tag{1.4}$$

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:

$$\begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix}. \tag{1.5}$$

This work calculated the rotation matrices two different ways with this orientation matrix formulation. The first method used the MARMOT-generated rotation matrices. A second

method calculated the rotation matrices using geometric arguments (see Figure 1.4). From the geometric arguments this work identified the normals given in Table 1.1. **??** shows the code used to generate the rotation matrices.

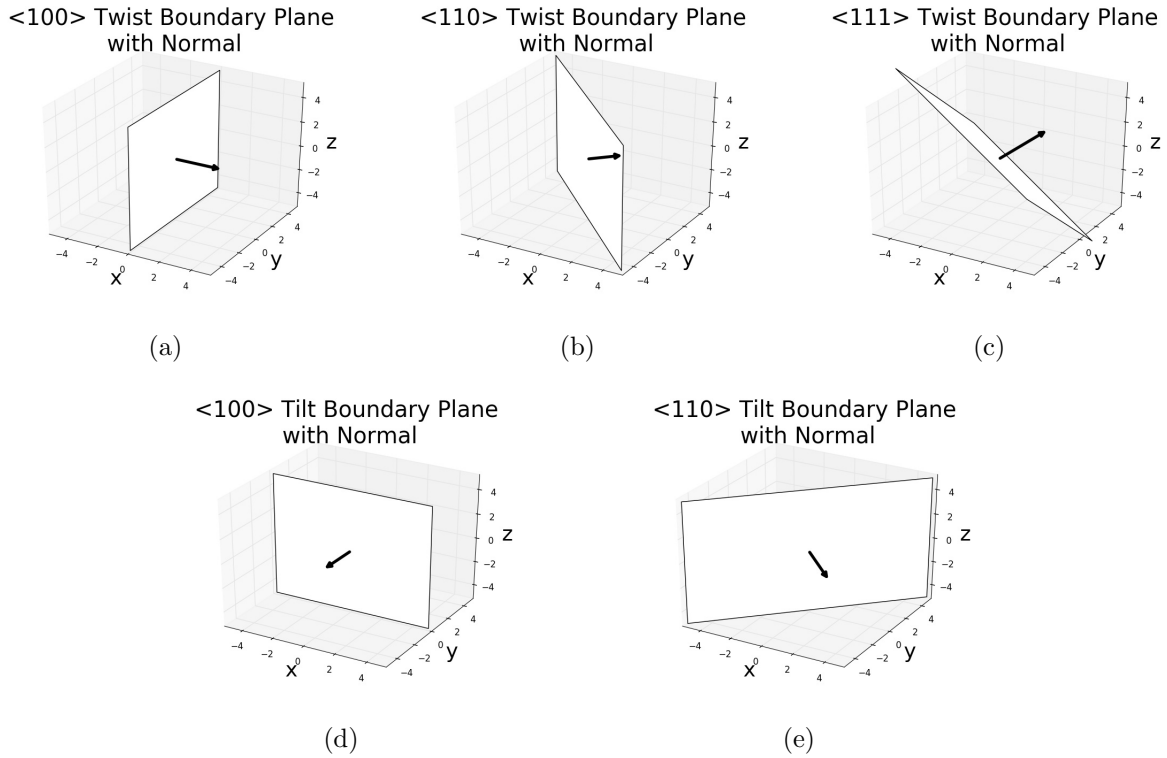


Figure 1.4: A geometric method of determining the normals of a GB. (a) to (c) show the normals for a GB perpendicular to the axis of rotation (a twist GB). The axis about which the grains rotate defines the GB normal. (d) and (e) show the normals for a GB parallel to the axis of rotation (a tilt GB).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

Bunge Rotation Matrix

MARMOT uses the Bunge rotation matrix (see Equation (1.3)) to create the orientation matrices. This work used various methods to calculate the Euler angles, of which three are briefly described here. The first two methods use the Euler angles 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 $[\bar{1}00]$. This work assumed boundary conditions to be either perpendicular or parallel to the rotation axis, while MARMOT's boundary conditions assume GB normals along the x- or y-axes.

The second method used an open-source MATLAB® package called MTEX.[15] This package calculates Euler angles using quaternions. These Euler angles did not generate the correct

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. Each subset has two options for the grain boundary normals 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	$[111]$ $[\bar{1}\bar{1}\bar{1}]$

results either, for the most part creating the same sorts of graphs as the MARMOT method.

The working method used the mathematics of quaternions directly.[16] This work calculated the quaternions based on the misorientation axis and angle. A quaternion is a four-dimensional vector containing one real part, and three imaginary parts, calculated as follows:

$$\boldsymbol{q} = \left[\cos\left(\frac{\theta}{2}\right),\ a_x \sin\left(\frac{\theta}{2}\right),\ a_y \sin\left(\frac{\theta}{2}\right),\ a_z \sin\left(\frac{\theta}{2}\right)\right], \tag{1.6}$$

with axis $\boldsymbol{a}$ and misorientation angle θ. After converting the axis and misorientation angle to a quaternion, another conversion changes the quaternion to a set of Bunge Euler angles. Calculation of the angles uses Python's atan2() method, allowing all four quadrants in Cartesian space to be accounted for.

$$\begin{aligned}
\chi &= \sqrt{(q_0^2 + q_3^2)(q_1^2 + q_2^2)} \\
\varphi_1 &= \operatorname{atan2}\left(\frac{q_0 q_2 + q_1 q_3}{2\chi}, \frac{q_0 q_1 - q_2 q_3}{2\chi}\right) \\
\Phi &= \operatorname{atan2}\left(2\chi, q_0^2 + q_3^2 - q_1^2 - q_2^2\right) \\
\varphi_2 &= \operatorname{atan2}\left(\frac{q_1 q_3 - q_0 q_2}{2\chi}, \frac{q_0 q_1 + q_2 q_3}{2\chi}\right).
\end{aligned} \tag{1.7}$$

Inputting the Euler angles into Equation (1.3) created the orientation matrices for the grains. **????** provide the codes used to generate the orientation matrices.

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. Different methods experienced various levels of success.

?????? show the matrices giving the best results.

While the first method works well for MARMOT, the MATLAB® script does not necessarily expect the same GB normal assumed by MARMOT. Thus, the results coming from using this combination of matrices ended up working only for the ⟨100⟩ tilt, ⟨110⟩ tilt, and ⟨100⟩ twist subsets. The ⟨110⟩ twist subset had issues with singularities, and the ⟨111⟩ subsets did not remotely match the expected outcome.

1.4.2 Calculating Reduced Chi Squared

This work used two methods 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. **??** discusses the results from these calculations.

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. This work used Equation (1.8) to calculate the χ^2_{red} value for each subset and for the entire fit, producing the results in **??** under the 800 K anneal column under the "χ^2_{red} using P and Q matrices" section,

$$\chi^2_{\text{red}} = \frac{1}{N-n-1} \sum \frac{(\epsilon_{\text{md}} - \epsilon)^2}{e\ \epsilon_{\text{md}}}. \tag{1.8}$$

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 that create 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. This work implemented the same methods to calculate the χ^2_{red} values for the data without an anneal.

The statistic calculated using these methods differs from the χ^2 statistic used in the grid-search function. The grid-search function used Equation (1.9), and generated values of the same order as Equation (1.8).

$$\chi^2 = \sum (E_{\text{measured}} - E_{\text{calculated}})^2 \tag{1.9}$$

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

[12] 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

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

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

[15] 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

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