Five Degree-of-Freedom Property Interpolation of Arbitrary Grain Boundaries via Voronoi Fundamental Zone Octonion Framework

Sterling Gregory Baird
Brigham Young University

Follow this and additional works at: https://scholarsarchive.byu.edu/etd

Part of the Engineering Commons

BYU ScholarsArchive Citation
https://scholarsarchive.byu.edu/etd/9014

This Thesis is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen amatangelo@byu.edu.
Five Degree-of-Freedom Property Interpolation of Arbitrary Grain Boundaries via Voronoi
Fundamental Zone Octonion Framework

Sterling Gregory Baird

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

Copyright © 2021 Sterling Gregory Baird
All Rights Reserved
ABSTRACT

Five Degree-of-Freedom Property Interpolation of Arbitrary Grain Boundaries via Voronoi Fundamental Zone Octonion Framework

Sterling Gregory Baird
Department of Mechanical Engineering, BYU
Master of Science

In this work I introduce the Voronoi fundamental zone octonion (VFZO) interpolation framework for grain boundary (GB) structure-property models and surrogates. The VFZO framework offers an advantage over other five degree-of-freedom (5DOF) based property interpolation methods because it is constructed as a point set in a Riemannian manifold. This means that directly computed Euclidean distances approximate the original octonion distance with significantly reduced computation runtime (∼7 CPU minutes vs. 6.6 CPU years for a 50000 × 50000 pairwise-distance matrix). This increased efficiency facilitates lower interpolation error through the use of significantly more input data. I demonstrate grain boundary energy (GBE) interpolation results for a non-smooth validation function and simulated bi-crystal datasets for Fe and Ni using four interpolation methods: barycentric interpolation, Gaussian process regression (GPR) or Kriging, inverse-distance weighting (IDW), and nearest neighbor (NN) interpolation. These are evaluated for 50000 random input GBs and 10000 random prediction GBs. The best performance was achieved with GPR, which resulted in a reduction of the root mean square error (RMSE) by 83.0% relative to RMSE of a constant, average model. Likewise, interpolation on a large, noisy, molecular statics (MS) Fe simulation dataset improves performance by 34.4% compared to 21.2% in prior work. Interpolation on a small, low-noise MS Ni simulation dataset is similar to interpolation results for the original octonion metric (57.6% vs. 56.4%). A vectorized, parallelized, MATLAB interpolation function (interp5DOF.m) and related routines are available in my VFZO repository (github.com/sgbaird-5dof/interp) which can be applied to any of the 32 crystallographic point groups. The VFZO framework offers advantages for computing distances between GBs, estimating property values for arbitrary GBs, and modeling surrogates of computationally expensive 5DOF functions and simulations.

Keywords: grain boundary, structure-property model, interpolation, octonion, machine learning
ACKNOWLEDGMENTS

I thank my advisor, Dr. Johnson for his extensive mentoring, encouragement, patience, and willingness to let me pursue new ideas and make course corrections based on new information. I have become a better researcher and scholar through his help. I thank my committee members, Dr. Homer and Dr. Fullwood for valuable feedback and suggestions and their willingness to teach and support. I thank my wife, Victoria Baird, for her continuous support, technical expertise, troubleshooting, and sacrifice. Aided by our discussions, I overcame several fundamental challenges associated with the VFZO framework. I also express appreciation for individuals both within and outside of BYU who have helped me: Ian Chesser, Toby Francis, Christian Kurniawan, Tyler Critchfield, Brandon Snow, Jose Nino, Chris Adair, Jeremy Green, and Nathan Miller. This material is based upon work supported by the National Science Foundation under Grant No. 1610077. This work was supported in part through computational resources provided by Brigham Young University’s Office of Research Computing.

CRediT statement: Sterling Baird: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Writing - Original Draft, Writing - Review & Editing, Visualization. Oliver Johnson: Supervision, Project administration, Funding acquisition, Conceptualization, Writing - Review & Editing. David Fullwood: Funding acquisition, Writing - Review & Editing. Eric Homer: Funding acquisition, Writing - Review & Editing
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Comparison between Voronoi fundamental zone octonion and traditional octonion frameworks. *6D Cartesian representation used only for mesh triangulation efficiency in barycentric interpolation and *7D Cartesian representation only required for barycentric interpolation. 7D Cartesian representation is also implemented (though not required) for GPR, NN, and IDW. For pairwise distance complexity, $N_p$ is the symmetry cardinality ($N_p = 24$ for $m\bar{3}m$ face-centered cubic (FCC) point group) and $L$ is the number of GBs.</td>
</tr>
<tr>
<td>3.1</td>
<td>Comparison of average interpolation RMSE (approximately 10 trial runs) for each interpolation method in the present work, using 50000 points in the definition of the Voronoi fundamental zone (VFZ) and GBEs obtained by evaluating the Bulatov Reed Kumar (BRK) validation function ([1]) at these points. A constant model (Cst, Avg RMSE), whose value was chosen to be the mean of the input GBE was used as a control. The last two columns represent the reduction ($\downarrow$) in RMSE in absolute units of Jm$^{-2}$ and % relative to the control model, respectively.</td>
</tr>
<tr>
<td>3.2</td>
<td>Comparison of average interpolation mean absolute error (MAE) (approximately 10 trial runs) for each interpolation method in the present work, using 50000 points in the definition of the VFZ and GBEs obtained by evaluating the BRK validation function ([1]) at these points. A constant model (Cst, Avg MAE), whose value was chosen to be the mean of the input GBE was used as a control. The last two columns represent the reduction ($\downarrow$) in MAE in absolute units of Jm$^{-2}$ and % relative to the control model, respectively.</td>
</tr>
<tr>
<td>3.3</td>
<td>Approximate coordinates of VFZO A and B used for the interpolation in Figure 3.3. Individual quaternions of each octonion are given in the active sense and in the laboratory reference frame with an assumed GB normal pointing in the +z direction, also in the laboratory reference frame.</td>
</tr>
<tr>
<td>3.4</td>
<td>Comparison of average runtime (s) for 10 trials for barycentric, GPR, IDW, and NN interpolation methods for various input VFZO set sizes using 12 cores and evaluated on 10000 prediction VFZOs. Because GPR, IDW, and NN method defaults do not use parfor loops but may have internal multi-core vectorization, it is unclear to what extent the number of cores affects the runtime of methods other than barycentric interpolation. VFZO symmetrization runtime was not included; however, symmetrization of 50000 grain boundary octonions (GBOs) takes approximately 76 seconds on 6 cores (Intel i7-10750H, 2.6 GHz) and is a common step in every interpolation method (i.e. it is fundamental to the VFZO framework) We used the BRK validation function for GBE [1].</td>
</tr>
<tr>
<td>3.5</td>
<td>Comparison of interpolation MAE (1 trial run) for 0 K molecular statics (MS) datasets. A constant model (Cst, Avg MAE), whose value was chosen to be the mean of the input GBE was used as a control. The last two columns, MAE $\downarrow$ (Jm$^{-2}$ and MAE $\downarrow$ (%)), represent the reduction in MAE in units of Jm$^{-2}$ and % relative to the control model, respectively. Non-sym refers to distances calculated in [2] without regard for crystal symmetries.</td>
</tr>
</tbody>
</table>
3.6 Comparison of interpolation RMSE (1 trial run) for 0K molecular statics (MS) datasets. A constant model (Cst, Avg RMSE), whose value was chosen to be the mean of the input GBE was used as a control. The last two columns, RMSE ↓ (Jm$^{-2}$ and RMSE ↓ (%)), represent the reduction in RMSE in units of Jm$^{-2}$ and % relative to the control model, respectively. Non-sym refers to distances calculated in [2] without regard for crystal symmetries.

3.7 Approximate coordinates of VFZO$s A$ and $B$ used for the MS Fe simulation dataset interpolation in Figure 3.4. Individual quaternions of each octonion are given in the laboratory reference frame with an assumed GB normal pointing in the +z direction, also in the laboratory reference frame.
2.1 (a) 3D Cartesian analogue to a non-degenerate 7D Cartesian representation of U(1)-symmetrized GBOs and VFZOs (VFZOs are inherently U(1)-symmetrized) which demonstrates the symmetrization of many points relative to a fixed, reference point (white circle). This produces a 3D Cartesian VFZ point set (dark blue points). (b) To further illustrate, a single input point (magenta points) is symmetrized (dark blue point) relative to a fixed, reference point (white circle) demonstrating that only one symmetrized point is found within the borders (black) of each of the Voronoi cells (light blue). The Voronoi tessellation is defined by the symmetric images of the reference point, and the spherical Voronoi diagram for this illustration is constructed using a modified version of [3].

2.2 (a) Histogram of NN octonion distances ($\omega$) in a VFZO set of 50000 points. The average NN distance was $(2.87090 \pm 0.69112)$. (b) The average k-th nearest neighbor distances demonstrate that many nearest neighbors fall within a tight tolerance (less than 10°) out of approximately 10 trial runs.

2.3 Hexagonally binned parity plots of pairwise distances of 388 Ni bicrystals [4]. Euclidean distance approximation is converted to octonions ($x_{i,j,k} = 2 \left( \frac{180}{\pi} \right) |\hat{o}_{i,k}^{sym} - \hat{o}_{j,k}^{sym}|$) for comparison with the traditional octonion metric [5]. The minimum distance among an ensemble of VFZO sets ($\min_{\forall k \in [1,k_{max}]} x_{i,j,k}$) is used for (a) 1, (b) 2, (c) 10, and (d) 20 VFZO sets. As the number of VFZO sets increases, the correlation between the Euclidean distance and the traditional octonion distance improves.

2.4 RMSE and MAE of pairwise distance errors for 388 Ni bicrystals [6] of scaled Euclidean distance approximation relative to the traditional octonion metric [5] (compare with Figure 2.3). The minimum distance among an ensemble of VFZO sets ($\min_{\forall k \in [1,k_{max}]} x_{i,j,k}$, where $x_{i,j,k}$ is the scaled Euclidean distance) is taken, iteratively adding consecutive sets up to $k_{max} = 20$. As the number of VFZO sets increases, RMSE and MAE between the scaled Euclidean distance approximation and the traditional octonion distance decreases.

2.5 NN VFZO ($\omega_{NN}$) distances (°) versus VFZO set size out of 70-80 random VFZO sets per set size.

3.1 Hexagonally binned parity plots for 50000 input and 10000 prediction octonions formed via pairs of a random cubochorically sampled quaternion and a spherically sampled random boundary plane normal. Interpolation via (a) GPR, (b) IDW, (c) NN, and (d) barycentric coordinates. BRK GBE function for FCC Ni [1] was used as the test function.

3.2 (a) Average RMSE and (b) average MAE vs. number of input points for (planar) barycentric (blue), GPR (orange), IDW (yellow), and NN (purple) interpolation for approximately 10 random runs with different input and prediction points. Standard deviations of approximately 10 runs are also included. Compare with approximately 0.1283 Jm$^{-2}$ and 0.0955 Jm$^{-2}$ RMSE and MAE, respectively, for a constant, average model (green) using the average of the input properties (approximately 1.16 Jm$^{-2}$).
3.3 Predictions of GPR (blue circles), barycentric (red circles), NN (magenta circles), and IDW (green circles) as a function of distance along a 1D arc ($\overline{AB}$) between two VFZOs ($A$ and $B$). The true, underlying BRK function is also shown (black line). 50000 random input VFZOs were generated and used for each of the models. 150 equally spaced points between $A$ and $B$ obtained via octonion Spherical Linear Interpolation (oSLERP) [7] were used as prediction points. GPR uncertainty standard deviation is plotted as shaded error band.

3.4 Interpolation results for a large Fe simulation database [8] using 46883 input GBs and 11721 prediction GBs in an 80%/20% split and a Gaussian process regression mixture (GPRM) model to better approximate low GBEs. Use of a GPRM model predicts low GBE better than the standard GPR model (compare with Figure 4.5d). (a) Hexagonally binned parity plot of the GPR mixing model with RMSE and MAE of 0.055035 Jm$^{-2}$ and 0.039185 Jm$^{-2}$, respectively, relative to typical, constant average models of 0.0854 Jm$^{-2}$ and 0.0617 Jm$^{-2}$, respectively. (b) Predictions of GPRM model (blue circles) as a function of distance along a 1D arc ($\overline{AB}$) between two VFZOs ($A$ and $B$).

4.1 Parity plot of 8D Cartesian hyperspherical arc length vs. 8D Cartesian Euclidean distance for pairwise distances in a ($m^3m$) symmetrized set of 10000 randomly sampled VFZOs. The max arc length is approximately 0.58 rad, indicating a max octonion distance of approximately 1.16 rad or 66.5° between any two points in the VFZ. The close correlation between arc length and Euclidean distance supports the validity of using Euclidean distance instead of arc length in the interpolation methods. This is separate from the correlation between VFZO Euclidean or arc length distances with the traditional octonion distance [5].

4.2 (a) RMSE and (b) MAE vs. ensemble size for mean, median, minimum, and maximum homogenization functions. A GPR model with 50000 input and 10000 prediction VFZOs was used.

4.3 Hexagonally binned parity plots for (a) mean, (b) median, (c) minimum, and (d) maximum ensemble homogenization functions. A GPR model with 50000 input and 10000 prediction VFZOs was used.

4.4 Illustration of two prediction points (red) for which no intersecting facet is found due to being positioned within a high-aspect ratio facet. The inset shows that facets connected to the NN do not contain the prediction point. Many NNs would need to be considered before an intersection is found. Additionally, it is expected that if found, the interpolation will suffer from higher error due to use of facet vertices far from the interpolation point. Proper intersections of prediction points with the mesh are shown in blue.
4.5 (a) Hexagonally binned parity plot of the standard GPR model. (b) All prediction GBs based on the model using only training GBs with a GBE less than 1.2 J m\(^{-2}\). (c) Combined disjoint model as explained in the text. (d) Hexagonally binned parity plots of the final GPR mixing model. Points in (c) are produced by splitting the prediction data into less than and greater than 1.2 J m\(^{-2}\). A sigmoid mixing function (Figure 4.6) is then applied where the predicted GBEs shown in (c) determines the mixing fraction \(f\) to produce a weighted average of models (a) and (b). A large Fe simulation database [8] using 46,883 training datapoints and 11,721 validation datapoints in an 80%/20% split. The GPR mixture model decreases error for low GBE and changes overall RMSE and MAE from 0.057932 J m\(^{-2}\) and 0.039857 J m\(^{-2}\) in the original model (shown in (a)) to 0.057502 J m\(^{-2}\) and 0.041272 J m\(^{-2}\) (shown in (d)), respectively.

4.6 Sigmoid mixing function used in the GPR mixing model with \(m = 30\) and \(b = 1.1\) J m\(^{-2}\) (Eq. (4.2)).

4.7 Histogram of number of sets vs. number of degenerate GBs per set for the Fe simulation dataset [8]. Most sets have a degeneracy of fewer than 5.

4.8 Degenerate GBs sets are split into those with a degeneracy of fewer than 5 and greater than or equal to 5 and plotted as ( (a) and (c), respectively) error histograms and ( (b) and (d), respectively) hexagonally-binned parity plots. Large degenerate sets tend to have very low error, whereas small degenerate sets tend to have higher error. In other words, GBs that are more likely to be repeated many times based on the sampling scheme in [8] tend to give similar results, whereas GBs that are less likely to be repeated often have larger variability in the simulation output. We do not know if this has physical meaning or is an artifact of the simulation setup.

4.9 (left) Hexagonally binned parity plot for Ni simulation grain boundary energy (GBE) interpolation using LOOCV. (right) Parity plot for leave-one-out cross validation (LOOCV) interpolation results reproduced from Figure 6a of [5] under CC-BY Creative Commons license.

B.1 3D Cartesian to 2D Cartesian analogue of 8D Cartesian to 7D Cartesian degeneracy removal via rigid singular value decomposition (SVD) transformation as used in barycentric interpolation approach. (a) Starting spherical arc points on surface of 2-sphere, (b) rotational symmetrization applied with respect to the z-axis (analogous to U(1) symmetrization), and (c) degenerate dimension removed via singular value decomposition transformation to 2D Cartesian with either the origin (black plus) preserved (black asterisks, \(\text{zero}Q=T\)) for triangulation or ignored (red asterisks, \(\text{zero}Q=F\)) for mesh intersection. The spheres (a,b) and circle (c) each have a radius of 0.8 and are used as a visualization aid only.

B.2 3D Cartesian to 2D Cartesian analogue of 7D Cartesian to 6D Cartesian mesh triangulation used in barycentric interpolation approach. (a) 3D Cartesian input points are linearly projected onto hyperplane that is tangent to mean of starting points. (c) The degenerate dimension is removed via a rigid SVD transformation to 2D Cartesian and the Delaunay triangulation (black lines) is calculated, with input vertices (red). Delaunay triangulation superimposed onto normalized input points (d). The spheres in (a), (b), and (d) have a radius of 0.8 and are used for visualization only.
B.3 A ray (red line) is linearly projected from the 2-sphere onto the hyperplane of a mesh facet (transparent black), shown as a red asterisk. The barycentric coordinates are computed as \( \lambda_i \in [1, 3] = \frac{1}{3} \). Because all barycentric coordinates are positive, it is determined that the projected point is an intersection with the mesh. Given vertex values of 8.183, 3.446, and 3.188 for vertices 1, 2, and 3, respectively, the interpolated value is calculated as 4.94 via Eq. (B.1).
CHAPTER 1. INTRODUCTION

We present a new method for interpolating and predicting GB properties from a set of measured/calculated values (e.g. GBE from MS simulations) We term this approach the VFZO framework. It is highly efficient and facilitates the use of large data sets to enhance prediction accuracy We discuss motivation for (Section 1.1) and prior implementations of 5DOF property prediction (Section 1.2) and then highlight unique properties of the VFZO framework that offer advantages over other methods (Section 1.3).

1.1 Motivation

High fidelity GB structure-property models can accelerate the design and understanding of materials for GB engineering applications such as grain growth (GBE [9], mobility [10], and grain rotation [11–14]), stress-corrosion cracking (diffusivity [15, 16] and solubility [17]) [18–22], strength [23–25], ceramics [26, 27], and electronics [28, 29]. With the increased use of nanomaterials [23,29], GBs take on increasingly larger roles as the GB volume fraction becomes significant; this is complicated by the fact that properties of GBs can span orders of magnitude depending on the five macroscopic degrees of freedom (DOFs) [30–32] as well as the three microscopic DOFs [33, 34]. However, the mentioned studies generally only consider a binary classification of GBs or variation of a few DOFs which represents a small “slice” of the full grain boundary character (GBC) space. Recent advances in high-throughput simulation [32, 35–41], experimental characterization [32, 40, 42–44], and availability of rich GB datasets [4, 6, 39, 41, 45–51] warrant high-fidelity structure-property models capable of handling large amounts of input data to aid in the aforementioned applications.
1.2 Prior Work

In prior work, a number of strategies have been developed for predicting \(^1\) 5DOF GB properties from experimental or simulated data. Because different works use different validation functions and data, it is difficult to objectively compare their performance. To facilitate meaningful comparisons, in addition to quoting absolute performance in terms of RMSE or MAE, we will also report the percent reduction in error compared to a constant-valued control model whose value is chosen as the mean of the respective input data.

Several researchers have taken the approach of discretizing unsymmetrized 5DOF GBC space, and then using a least squares objective function and gradient descent to fit a piecewise-constant function, resulting in 5DOF grain boundary energy distributions (GBEDs) for nickel [47], yttria [52], and copper [48] based on experimentally characterized 3D microstructures.

[2] used an artificial neural network (ANN) and approximately 17000 and 51000 Fe bicrystal simulations from [45] as training and validation data, respectively, to achieve MAEs of 0.0486 Jm\(^{-2}\) and approximately 0.09 Jm\(^{-2}\) in the best fitted ANNs for randomly selected and special GBs, respectively. If a constant, average value (i.e. average of the input GBEs) was chosen as the model, the MAE would be 0.0617 Jm\(^{-2}\), implying that predictions of randomly selected GBs were improved by 21.2\% relative to this simple, control model.

Recently, a new GB representation, GBOs, was reported [7] and tested [5]. The GBO representation is valuable for a number of applications. Most relevant to the present work is the resulting distance metric. The GBO distance metric offers an advantage over other metrics in that it “correctly determines the angular distances between GBs with a common normal or misorientation” and “closely approximates the geodesic metric on \(SO(3) \times SO(3)\) for all grain boundary pairs while maintaining the ability to be analytically minimized with respect to the \(U(1)\) symmetry” [7]. In this context, [7] derived oSLERP and provided examples showing that oSLERP produces smooth, minimum distance paths through GB character space between two arbitrary GBs.

Laplacian kernel regression (LKR) (similar to IDW) involving scaled pairwise distance matrices was later used with GBOs to predict properties of arbitrary GBs from a set of known val-

---

\(^1\)We use the term “predict” throughout this work to refer to interpolation, inference, and/or extrapolation as some approaches can individually involve multiple prediction types.
ues [5]. Using k-fold cross validation (kFCV) with \( k = 10 \) for 388 Ni GBE simulations [4] and an optimized scaling parameter, a RMSE of 0.0977 J m\(^{-2}\) was obtained compared to a constant, average model RMSE of 0.2243 J m\(^{-2}\) (56.4% improvement). Due to computation time of pairwise distance matrices, this approach is currently “limited to datasets with several thousand or fewer” GBs [5].

1.3 Voronoi Fundamental Zone Octonion Framework

The VFZO interpolation framework introduced in this work offers an advantage over other methods because it is defined as a VFZ point set in a Riemannian manifold\(^2\) for which directly computed, scaled Euclidean distances approximate the original octonion distance given by [7]. This advantage is manifest in the ability to triangulate a mesh using standard routines (e.g. quick-hull [54]) and interpolate using barycentric coordinates or machine learning methods such as GPR. Building on prior work on GBOs [5, 7], we create a VFZ point set by obtaining a set of octonions minimized with respect to Euclidean distance and an arbitrary reference octonion after considering all symmetrically equivalent octonions (SEOs). Because GBOs are guaranteed to reside on the surface of a hypersphere [7] (a type of Riemannian manifold) a point set which locally resembles Euclidean space is the result (Section 2.1.3). Below we provide the detailed description of the method, followed by numerical test results (Chapter 3).

We also provide a vectorized, parallelized implementation of the VFZO framework and related functions. These are contained in what we will refer to as the VFZO repository, which is available at github.com/sgbaird-5dof/interp. In what follows, when we refer to built-in MATLAB functions, we refer to them with parentheses as in interp1(). When we refer to functions in the VFZO repository, we do so with the .m extension as in interp5DOF.m unless specifying the usage with arguments as in interp5DOF(qm,nA,qm2,nA2,y).

\(^2\)A “Riemannian manifold is a smooth differential manifold equipped with a metric tensor, which can be seen as a tool for calculating distances” [53].
CHAPTER 2. METHODS IN THE VFZO FRAMEWORK

The core operations of the VFZO framework are:

1. generating GBOs (Section 2.1.1)
2. mapping GBOs into a VFZ (Section 2.1.2)
3. calculating distances within the VFZ (Section 2.1.3)

Three degree-of-freedom (3DOF) fundamental zones (FZs) have typically been defined using linear inequalities (e.g. the orientation [55] and misorientation [55, 56] FZs). Instead of using linear inequalities\(^1\), we take a numerical approach to define what we will call a VFZ.

The construction of the VFZ dramatically reduces the computational burden of pairwise distance calculations. The mechanism by which this reduction is achieved can best be illustrated with an example. Let \(o_1\) and \(o_2\) denote two GBs represented as GBOs. To perform a traditional distance calculation it is necessary to compare all SEOs of \(o_1\) to all of the SEOs of \(o_2\) and take the smallest distance. If \(N_p\) is the cardinality of the crystallographic point group, this single minimum distance calculation requires a total of \(N_p^4\) SEOs to be considered. If one desires to compute a pairwise distance matrix between \(L\) GBs, all SEOs of each GB are compared against all SEOs of all others so that the total number of SEO computations will be \(O(N_p^4L^2)\).

In contrast, for a single distance calculation using the VFZO framework, \(o_1\) and \(o_2\) are first mapped into the VFZ, and then only a single distance calculation is required between them. Mapping \(o_1\) into the VFZ requires comparison of all \(N_p^2\) SEOs of \(o_1\) with a fixed reference GB in the interior of the VFZ; and likewise for \(o_2\). Consequently, a single distance calculation between \(o_1\) and \(o_2\) under the VFZO framework requires \(2N_p^2\) SEO computations. If one desires to compute a pairwise distance matrix between \(L\) GBs, the total computational cost\(^2\) will be \(O(N_p^2L)\), which rep-

---

\(^1\) If desired, linear inequalities can be obtained for a VFZ by determining a Voronoi tessellation’s junction points (similar to what is shown in Figure 2.1 by e.g. \texttt{voronoin()}), transforming to 6D Cartesian coordinates via a SVD transformation (Appendix B) and defining the bounded region by e.g. MATLAB FEX function \texttt{vert2lcon.m}.

\(^2\) See Section 3.2.2 for a detailed explanation of why this is not \(O(N_p^2L^2)\).
Table 2.1: Comparison between Voronoi fundamental zone octonion and traditional octonion frameworks. *6D Cartesian representation used only for mesh triangulation efficiency in barycentric interpolation and *7D Cartesian representation only required for barycentric interpolation. 7D Cartesian representation is also implemented (though not required) for GPR, NN, and IDW. For pairwise distance complexity, \( N_p \) is the symmetry cardinality (\( N_p = 24 \) for \( m\bar{3}m \) FCC point group) and \( L \) is the number of GBs.

<table>
<thead>
<tr>
<th>Property</th>
<th>Traditional</th>
<th>This Work</th>
</tr>
</thead>
<tbody>
<tr>
<td>Symmetrizing Distance</td>
<td>GBO</td>
<td>VFZ Euclidean</td>
</tr>
<tr>
<td>Dimensionality</td>
<td>8D Cartesian</td>
<td>6*/7*/8D Cartesian</td>
</tr>
<tr>
<td>Bounded by FZ</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Pairwise Distance Complexity</td>
<td>( O(N_p^4L^2) )</td>
<td>( O(N_p^2L) )</td>
</tr>
<tr>
<td>Rotation Convention</td>
<td>Passive</td>
<td>Active</td>
</tr>
</tbody>
</table>

represents a dramatic reduction compared to the traditional approach. A summary of the differences between the two approaches is provided in Table 2.1.

In this section we describe the methods for each of the three VFZO framework operations named above. We then describe our implementation of four different interpolation schemes for prediction of GB properties based on the VFZO approach.

2.1 The Voronoi Fundamental Zone Octonion Framework

2.1.1 Defining the Voronoi Fundamental Zone

To define a VFZ, an arbitrary, fixed, low-symmetry reference GBO is chosen (\( o_{\text{ref}} \)) and the VFZ is formally defined as the region of \( S^7 \) (the unit 7-sphere in 8 dimensions) closer to \( o_{\text{ref}} \) than any of its symmetric images\(^3\). However, use of the VFZ does not require its explicit construction. Rather, practical calculations require only the selection of the single point \( o_{\text{ref}} \) (which completes the definition of the VFZ), followed by mapping of query points into the VFZ by comparison of their SEOs with \( o_{\text{ref}} \).

To illustrate the process of mapping points into the VFZ, we describe a 3D Cartesian analogue (Figure 2.1) to a 7D Cartesian non-degenerate (i.e. U(1) degeneracy removed) representation\(^3\)

\(^3\)We also refer to lower-dimensional representations of the 8D Cartesian VFZ as VFZs (described in Section 2.1.2) and describe which dimensionality we are referring to as appropriate.
Figure 2.1: (a) 3D Cartesian analogue to a non-degenerate 7D Cartesian representation of U(1)-symmetrized GBOs and VFZOs (VFZOs are inherently U(1)-symmetrized) which demonstrates the symmetrization of many points relative to a fixed, reference point (white circle). This produces a 3D Cartesian VFZ point set (dark blue points). (b) To further illustrate, a single input point (magenta points) is symmetrized (dark blue point) relative to a fixed, reference point (white circle) demonstrating that only one symmetrized point is found within the borders (black) of each of the Voronoi cells (light blue). The Voronoi tessellation is defined by the symmetric images of the reference point, and the spherical Voronoi diagram for this illustration is constructed using a modified version of [3].

of a VFZ. A set of 500 points \( p_i, i \in [1, 500] \) randomly scattered on the surface of the 2-sphere comprise the data (red points in Figure 2.1a). A random point, \( p_{ref} \), also on the surface of the 2-sphere, is chosen as the reference point (white circle). In this illustration, \( O_h \) or \( m\bar{3}m \) point group rotations are used as symmetry operators, \( S_j, j \in [1, N_p] \), where \( N_p \) is the cardinality as before and \( N_p = 24 \) for the \( O_h \) point group. For each data point, 24 symmetrically equivalent representations \( p_{i,j}^{\text{sym}} = S_j(p_i), j \in [1, 24] \) are produced by applying each of the relevant symmetry operators. After calculating the Euclidean distance between \( p_{ref} \) and \( p_{i,j}^{\text{sym}} \), the point \( p_i^{*} \) closest to \( p_{ref} \) is chosen and retained as the unique representative of \( p_{i,j}^{\text{sym}} \). As illustrated in Figure 2.1a, the projected points \( p_i^{*} \) (dark blue points) all fall in the VFZ without ever having to construct or define it explicitly, we call this group of projected points a *VFZ point set*. Note also that there is only one \( p_i^{*} \) in the VFZ for each \( p_{i,j}^{\text{sym}} \) (see Figure 2.1b).
To calculate the distance between a given octonion, and the reference octonion, we employ the standard 8D Euclidean distance

\[ d_E(o_A, o_B) = \left( \sum_{k=1}^{8} (o_{A,k} - o_{B,k})^2 \right)^{1/2} \]  

(2.1)

where \( o_{A,k} \) and \( o_{B,k} \) represent the \( k \)-the element of normalized octonions \( o_A \), and \( o_B \), respectively.

Euclidean distance is an approximation to the true geodesic arc length on \( S^7 \), which is given by

\[ d_S(o_A, o_B) = \cos^{-1}(o_A \cdot o_B) \]  

(2.2)

where \( \cdot \) is the dot product, \( \cos^{-1} \) is the inverse cosine operator, and \( o_A \) and \( o_B \) are each normalized and \( d_S \approx d_E \) (Figure 4.1). In [7], the original octonion distance metric was defined by

\[ d_\Omega(o_A, o_B) = 2\cos^{-1}(o_A \cdot o_B) \]  

(2.3)

where \( o_A \) and \( o_B \) are each normalized and \( d_\Omega \) can be seen to be simply twice the geodesic arc length: \( d_\Omega = 2d_S \). Thus, \( d_E \approx \frac{1}{2}d_\Omega \).

The definition of \( d_\Omega \) has certain aesthetic benefits in that it mirrors the definition of a misorientation angle, \( \omega_{AB} \), between two crystal orientations in the quaternion parameterization:

\[ \omega_{AB} = 2\cos^{-1}(q_A \cdot q_B) \].

Our choice to use \( d_E \) instead of \( d_S \) or \( d_\Omega \) is motivated by the fact that it enables the use of standard algorithms, for a variety of operations, that require or assume Euclidean distances. In addition to enabling us to leverage the machinery of efficient and established algorithms, this choice can be justified by the following observations:

- The minimum Euclidean distance SEO will be the same as the minimum arc length distance SEO because \( d_S \) is a monotonically increasing function of \( d_E \), for \( d_S(d_E) \in [0, \pi] \) (Figure 4.1).

- For the FCC point group symmetry \((m\overline{3}m)\) the portion of \( S^7 \) subtended by the VFZ is sufficiently small that the approximation \( d_E \approx d_S \) holds to very high accuracy\(^4\) as shown in Figure 4.1.

\(^4\)This is true for a specific pair of octonions within a VFZ. When calculating the minimum distance between SEOs of two points, there are additional considerations that must be attended to as discussed in detail in Section 2.1.3.
Figure 2.2: (a) Histogram of NN octonion distances ($\omega$) in a VFZO set of 50000 points. The average NN distance was $(2.87090 \pm 0.69112)^\circ$. (b) The average k-th nearest neighbor distances demonstrate that many nearest neighbors fall within a tight tolerance (less then $10^\circ$) out of approximately 10 trial runs.

- Calculation of $d_E$ does not require the use of any inverse trigonometric functions and is about 23 % faster than calculation of $d_S$ or $d_\Omega$.

2.1.2 Mapping GBs to the Voronoi Fundamental Zone

As described above in the 3D analogy, with a reference GBO chosen ($o_{\text{ref}}$), and consequently the VFZ defined (Section 2.1.1), a GBO is mapped into the VFZ by finding among its SEOs the one that is closest to $o_{\text{ref}}$ according to $d_E$ (Eq. (2.1)). This is performed for all input and prediction points with respect to $o_{\text{ref}}$, and the result is a VFZ point set.

2.1.3 Distance Calculations in the Voronoi Fundamental Zone

Euclidean distances are an accurate approximation of arc length distances in a VFZ because the difference between the two metrics for the maximum pairwise distance ($pd_{\text{max}} \simeq 60^\circ$) in a VFZ is small as shown in Figure 4.1. However, when compared with the traditional octonion distance [7], due to the presence of low-symmetry GBs near the exterior of a VFZ, some GB
pairs will exhibit larger Euclidean or arc length distances than is truly representative (see e.g. Figure 2.3a). In other words, moving “past” the low-symmetry border of a VFZ will result in an instantaneous relocation to a possibly distant point in the VFZ that in reality is highly correlated.

Figure 2.3: Hexagonally binned parity plots of pairwise distances of 388 Ni bicrystals [4]. Euclidean distance approximation is converted to octonions \( (x_{i,j,k} = 2(\frac{180}{\pi})|\hat{o}_{i,k}^{\text{sym}} - \hat{o}_{j,k}^{\text{sym}}|) \) for comparison with the traditional octonion metric [5]. The minimum distance among an ensemble of VFZO sets \( \min_{k \in [1,k_{\max}]} x_{i,j,k} \) is used for (a) 1, (b) 2, (c) 10, and (d) 20 VFZO sets. As the number of VFZO sets increases, the correlation between the Euclidean distance and the traditional octonion distance improves.

This is a limitation of the VFZO framework, which generates a VFZ with low-symmetry GBs at the borders in contrast to typical FZs [57, 58]. While defining a FZ with high-symmetry GBs at the borders (especially mirror-symmetry GBs) will certainly increase interpolation accu-
racy, the favorable interpolation results presented in this work are obtained because overestimation is infrequent within a small correlation length (e.g. $10^\circ$ [6], which many NNs fall within for a 50000 VFZO set, see Figure 2.2b), and underestimation is non-existent within numerical precision. Naturally, smaller dataset pairwise distance matrices will exhibit more frequent distance overestimations.

Overestimation imposes a “sparseness” of data within a local region of influence common to the interpolation methods in this work, whereas underestimation would give erroneous high correlations between uncorrelated GBs. Because only overestimation relative to traditional octonion distances exist in this work (as shown in Figure 2.3), we expect that large errors will occur infrequently (Section 3.1).

While distance calculations are subject to these infrequent overestimates, they are largely immaterial for interpolation. This is because all of the interpolation methods in this work involve a region of influence that is small, so that if the distance to a NN is overestimated it simply does not contribute to the interpolation (the “sparseness” referred to earlier). Consequently the accuracy of the interpolation is not significantly impacted by infrequent distance overestimates, and excellent results can be achieved without addressing this limitation. However, if even greater accuracy is desired it can be obtained for a relatively minor cost by considering multiple VFZs.

We find that taking the minimum distance among several VFZO sets defined by separate reference octonions leads to better correlation between the Euclidean approximation and the traditional octonion metric as shown in Figure 2.3. Additionally, Figure 2.4 shows that the error between scaled Euclidean distance and the traditional octonion metric decreases rapidly as the number of ensemble VFZO components increases. This confirms that employing a small ensemble of VFZO sets results in significant improvement to the Euclidean distance approximation (Figures 2.3 and 2.4) of the traditional octonion metric. However, as already mentioned, improvements to interpolation results are expected to be less significant since they are already robust to occasional distance overestimates. In terms of computational runtime, use of an ensemble of 10 VFZOs will increase runtime by a factor of $\sim 10$ via a loop-based implementation. For a symmetrized $50000 \times 50000$ pairwise distance matrix, this results in a runtime of approximately 1 CPU hour instead of $\sim 7$ CPU minutes for a single VFZ. However, this is still much faster than the original.
Figure 2.4: RMSE and MAE of pairwise distance errors for 388 Ni bicrystals [6] of scaled Euclidean distance approximation relative to the traditional octonion metric [5] (compare with Figure 2.3). The minimum distance among an ensemble of VFZO sets (min\(\forall k\in[1,k_{max}]) x_{i,j,k}, \text{ where } x_{i,j,k} \text{ is the scaled Euclidean distance}) is taken, iteratively adding consecutive sets up to \(k_{max} = 20\). As the number of VFZO sets increases, RMSE and MAE between the scaled Euclidean distance approximation and the traditional octonion distance decreases.

octonion approach used in [5], which would take an estimated 6.6 CPU years (or 153 CPU days if one GB in the GB pair is fixed according to the assumption in [53]).

VFZO Euclidean, hyperspherical arc length, and octonion distances are computed via VFZO repository function GBdist4.m which is used in the symmetrization function get_octpairs.m and an example of ensemble VFZO distance calculations is given in plotting.m.

In addition to their use for distance calculations alone, ensembles of VFZO sets can be employed with interpolation methods to increase overall interpolation accuracy, but there is a computational cost (e.g. approximately 10× using an ensemble of 10 VFZO sets). For 50000 input points, use of an ensemble with 10 VFZO sets decreases RMSE and MAE from 0.0241 J m\(^{-2}\) and 0.0160 J m\(^{-2}\) to 0.0187 J m\(^{-2}\) and 0.0116 J m\(^{-2}\), respectively (single trial run) We expect these overall accuracy improvements occur because GBE predictions near the exterior of the VFZ where
data may be sparse are improved. Ensemble interpolation results as a function of ensemble size and parity plots for mean, median, minimum, and maximum functions applied to the ensemble are shown in Figure 4.2 and Figure 4.3, respectively. Further details of ensemble interpolation are given in Section 4.2.

2.2 Generating Random Voronoi Fundamental Zone Octonions

In addition to the 3 core operations of the VFZO framework described in Section 2.1, it will be necessary for our tests, and useful for other applications, to be able to generate random GBOs from 5DOF representations. We briefly explain here our process for accomplishing this.

First, random GBOs are formed by taking random misorientation quaternion (qm) and boundary plane (BP) normal (nA) pairs. Random misorientation quaternions are obtained via cubochoric sampling [59] (get_cubo.m) and random BP vectors are sampled from a multivariate Gaussian distribution (µ = 0, σ = 1) in R³ and normalized⁵. After this, they are converted to GBOs via VFZO repository function five2oct.m. The VFZO repository function get_five.m returns the result of these several operations. These (qm, nA) pairs are then converted to an octonion representation, o, using VFZO repository function o=five2oct(qm, nA) (see also VFZO repository function get_ocubo.m for generating random GBOs directly).

The GBOs are then symmetrized (i.e. they become VFZOs) via osym=get_octpairs(o). A default reference octonion⁶ is used for these calculations, unless specified by the user. We use the active convention for qm, nA, and o (see Appendix A for further details of conventions).

For the present work we use this procedure to randomly generate VFZO sets containing between 100 to 50000 VFZOs where each trial run has its own unique set of GBs. We use these to perform the validation and performance evaluation tests described later. For reference, we note that the average NN distance (over approximately 70 trials) of such sets ranges between (10.7175 ± 0.3684)° and (2.6479 ± 0.2254)°, respectively.

---

⁵Several methods for uniform sampling of points on a sphere, including the one mentioned here, are described in https://mathworld.wolfram.com/SpherePointPicking.html.

⁶This is generated by get_ocubo.m using a random number generator seed of 10. We expect that five2oct.m combined with get_five.m will generate near identical statistical properties to get_ocubo.m which is supported by a visual comparison of pairwise distance histograms (not shown in this work), and indirectly by an assertion in Section 5.3 of [53].
Figure 2.5: NN VFZO (ω\textsubscript{NN}) distances (°) versus VFZO set size out of 70-80 random VFZO sets per set size.

Figure 2.5 illustrates how the VFZO average NN distance varies with the cardinality of the set (i.e. number of random VFZOs in the set). For a specific 50000 VFZO set, the NN octonion distance is (2.87090 ± 0.69112)° (Figure 2.2a) while the average 100-th NN distance is within 10° (Figure 2.2b). This indicates that, on average, prediction VFZOs fall within a typical GB correlation length (10° [6]) of input VFZOs in large set sizes.

2.3 Interpolation in the Voronoi Fundamental Zone Octonion Framework

With the VFZO framework established, it is possible to define interpolation schemes over the VFZ to predict the properties of new GBs from the known properties of other GBs. For one application of interest to us, it is necessary to evaluate multiple different functions over a fixed set of input and prediction GBOs. In this section we first present a barycentric interpolation method that we have developed to efficiently accomplish this specialized task by pre-computing the interpolation weights (which remain fixed when only the function being evaluated changes) We then present adaptations of three other interpolation methods—GPR (Section 2.3.2), IDW (Section 2.3.3), NN (Section 2.3.4)—that are useful for general applications (an additional interpola-
tion method—GPRM—which we developed specifically for a non-uniformly distributed, noisy, simulation dataset is described in Section 2.5.3).

2.3.1 Barycentric Interpolation

Barycentric coordinates are a type of homogeneous coordinate system that reference a prediction point within a simplex [60] or convex polytope [60–62] based on “masses” or weights at the vertices, which can be negative. The prediction point is assumed to be the barycenter (center of mass) of the simplex or convex polytope, and weights at the vertices necessary to make this assumption true are determined. We utilize rigid SVD transformations and a standard triangulation algorithm (quickhull [54] via delaunayn() in VFZO repository function sphconvhulln.m) to define a simplicial mesh (Appendix B.1). We then use barycentric weights (i.e., coordinates) for computing intersections of a point within a simplicial facet (Appendix B.2) and for interpolation (Appendix B.3) [60]. A detailed explanation of the process is provided in Appendix B. The barycentric interpolation method is invoked in interp5DOF.m by setting the method argument to ‘pbary’.

2.3.2 Gaussian Process Regression

GPR or Kriging uses the notion of similarity between points to fit Gaussian processes (random variables) to data based on prior information and provides uncertainty information in addition to interpolated or inferred values. For a general treatment of GPR, see [63]. We use MATLAB’s built-in function, fitrgp(), with all default parameters\(^7\) except that PredictMethod = ‘fic’ was used regardless of the number of input points, and we assume a Euclidean approximation of the VFZ (see Section 2.1.3 and Figure 4.1). A slower, more accurate, and more memory-intensive PredictMethod = ‘exact’ parameter choice is also available (Section 3.2). The GPR interpolation method is invoked in interp5DOF.m by setting the method argument to ‘gpr’.

\(^7\)MATLAB R2020b was used for the Fe simulation dataset, all other results employed MATLAB R2019b, the latest installed version on the computing cluster.
2.3.3 Inverse-distance Weighting Interpolation

IDW interpolation applies a weighted average to points within a neighborhood of a query point to obtain an interpolated value. interp5DOF.m implements a simple IDW approach based on [64]. A default radius of influence of \( r = \sqrt{2} \mu \) is used, where \( \mu \) represents the mean NN distance, and where octonion distance is approximated by the Euclidean distance or 2-norm (see Section 2.1.3, and Figure 4.1). NN interpolation (Section 2.3.4) is used for a given query point when there are no input points in the radius of influence. The IDW interpolation method is invoked in interp5DOF.m by setting the method argument to 'idw'.

2.3.4 Nearest Neighbor Interpolation

NN interpolation takes the nearest input point relative to a query point and assigns the value of the NN input point to the query point. This is implemented via the built-in MATLAB function dsearchn() using a Euclidean approximation of octonion distance (see Section 2.1.3, and Figure 4.1). The NN interpolation method is invoked in interp5DOF.m by setting the method argument to 'nn'.

2.4 Use of Interpolation Function

To facilitate easy application of the presented methods, a vectorized, parallelized, MATLAB implementation, interp5DOF.m, is made available in the VFZO repository [65] with similar input/output structure to that of built-in MATLAB interpolation functions (e.g. scatteredInterpolant(), griddatan()). A typical function call is as follows:

\[
\text{ypred} = \text{interp5DOF}(q_m, nA, y, q_m2, nA2, \text{method})
\]

The argument \( y \) is a vector of known property values corresponding to the GBs defined by \((q_m, nA)\), which respectively denote pairs of GB misorientation quaternions and BP normals. The result, \( \text{ypred} \), is a vector of predicted/interpolated property values corresponding to the prediction GBs defined by \((q_m2, nA2)\).

Internally, these are converted to octonions and interpolation is performed using the selected method. For the validation function, these can be compared to the true GBEs \( y_{true} \). The methods used in this work are 'pbary', 'gpr', 'idw', and 'nn', corresponding to planar barycentric, GPR, IDW, and NN interpolation, respectively. A placeholder template with in-
structions for implementing additional interpolation schemes is also provided in \texttt{interp5DOF.m}. See [7] and \texttt{five2oct.m} [65] treatments of conversions to octonion coordinates in the passive and active senses, respectively (Appendix A).

## 2.5 Literature Datasets

In addition to performing validation tests of the VFZO framework, we will also describe results in which we apply it to actual GB property data available from literature sources. Here we briefly mention details related to the retrieval and processing of two MS simulation datasets from the literature: We describe GPR applied to Fe (Section 2.5.1) and Ni (Section 2.5.2) simulations, as well as a specialized GPRM model applied to Fe to address non-uniformity and noise concerns (Section 2.5.3).

### 2.5.1 Gaussian Process Regression for Fe Simulation Dataset

The Fe simulation data is obtained from [8] rather than [45] due to a mistake in the earlier dataset file\(^8\). GBs with a GBE less than 0.01 J m\(^{-2}\) are removed to get rid of “no-boundary” GBs. Repeated GBs are then identified and removed by converting all GBs into a VFZO set (see \texttt{Kim2oct.m}) and applying \texttt{avgrepeats.m} with \texttt{avgfn}='\texttt{min}' to sort the repeated GBs into “degenerate sets”\(^9\), and only the average GBE (and a single GB) within each degenerate set was retained. We estimate the intrinsic RMSE and MAE of the Fe simulation dataset to be 0.06529 J m\(^{-2}\) and 0.06190 J m\(^{-2}\), respectively. Minimum and maximum error was −0.2625 J m\(^{-2}\) and 0.2625 J m\(^{-2}\), respectively. See Section 4.4.2 for further details on methods used to estimate intrinsic error of the Fe simulation dataset.

### 2.5.2 Gaussian Process Regression for Ni Simulation Dataset

We use the GBOs representations [5] of GBs from [6] contained in \texttt{‘olm_octonion_list.txt’} [66], importing and converting them to the active sense by

\(^8\)We were informed of the error during an email discussion with the corresponding author of [8].

\(^9\)A degenerate “set” is distinct from a VFZOs “set”, the former of which is discussed in greater detail in Supplementary Information Section 4.4.2.
taking the quaternion inverse of each of the octonions’ quaternions We take GBE values from the first column of 'olm_properties.txt' ([66]), and use a GPR model (Section 2.3.2)

2.5.3 Gaussian Process Regression Mixture for Fe Simulation Dataset

Separate from the four main methods analyzed in this work, a GPRM model is developed to better predict low GBE using the non-uniformly distributed, noisy, Fe simulation dataset described in Section 2.5.1. An exponential rather than a squared exponential kernel was used for the subset GPR model (Section 4.4) to accommodate sharper transitions to better approximate low GBEs. Further details of the GPRM model are given in Supplementary Information (Section 4.4).
CHAPTER 3. RESULTS AND DISCUSSION

To illustrate the utility of the VFZO framework for one application, namely interpolation, we compare the (i) accuracy (Section 3.1), and (ii) efficiency (Section 3.2) of the four previously described interpolation methods implemented over the VFZ with each other and with existing methods from the literature (see Chapter 1). For these tests, we use the 5DOF GB energy function by [1] (trained on Ni bicrystal simulation data [6]) as a validation function which we refer to as the BRK function.

For a given trial run, an input VFZO set, \(o\), and a prediction VFZO set, \(o_2\), are randomly generated according to Section 2.2. The BRK validation function is evaluated at each of these points and the values are stored in the vectors \(y\) and \(y_{\text{true}}\), respectively. \(o\) and \(o_2\) are SVD transformed together into a 7D Cartesian representation, which is an optional step for GPR, NN, and IDW, but is required for barycentric interpolation. Finally, we use interp5D0F.m [65] to predict the function value at the prediction points, which are stored in the vector \(y_{\text{pred}}\) We repeat this process for each of the interpolation methods approximately 10 times per set size, and compare the predictions, \(y_{\text{pred}}\), to the true values, \(y_{\text{true}}\).

Following this validation study, we also demonstrate VFZO GPR interpolation applied to a large, noisy, MS Fe bicrystal simulation dataset [8] and a small, low-noise, MS Ni bicrystal simulation dataset [6] (Section 3.3.1), to evaluate performance on real GB property data.

3.1 Interpolation Accuracy

Figure 3.1 provides hexagonally binned parity plots (via VFZO repository function parityplot.m which depends on a modified version of [67]) for each of the four interpolation methods.

All of the methods permit successful interpolation, and the highest density region in all cases falls squarely on the parity line. The GPR and barycentric results show a slight asymmetry.
Figure 3.1: Hexagonally binned parity plots for 50000 input and 10000 prediction octonions formed via pairs of a random cubochorically sampled quaternion and a spherically sampled random boundary plane normal. Interpolation via (a) GPR, (b) IDW, (c) NN, and (d) barycentric coordinates. BRK GBE function for FCC Ni [1] was used as the test function.

such that low energy values are overpredicted more often than they are underpredicted. The width of the point clouds provides a qualitative indication of the dispersion in the prediction errors, and the logarithmically scaled color indicates the frequency of errors of a given magnitude. As can be seen, the vast majority of errors are very small (the highest density—yellow region—is concentrated on the line of parity). Quantitative measures of the overall accuracy are presented for RMSE (Table 3.1) and MAE (Table 3.2), and will be discussed in detail below. Error metrics are obtained via VFZO repository function get_errmetrics.m.
Table 3.1: Comparison of average interpolation RMSE (approximately 10 trial runs) for each interpolation method in the present work, using 50000 points in the definition of the VFZ and GBEs obtained by evaluating the BRK validation function ([1]) at these points. A constant model (Cst, Avg RMSE), whose value was chosen to be the mean of the input GBE was used as a control. The last two columns represent the reduction (↓) in RMSE in absolute units of Jm$^{-2}$ and % relative to the control model, respectively.

<table>
<thead>
<tr>
<th>Method</th>
<th>Distance</th>
<th>Dataset</th>
<th># GBs</th>
<th>RMSE (Jm$^{-2}$)</th>
<th>Cst, Avg RMSE (Jm$^{-2}$)</th>
<th>RMSE ↓ (Jm$^{-2}$)</th>
<th>RMSE ↓ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPR</td>
<td>VFZ</td>
<td>BRK</td>
<td>50000</td>
<td>0.0218</td>
<td>0.1283</td>
<td>0.1065</td>
<td>83</td>
</tr>
<tr>
<td>Barycentric</td>
<td>VFZ</td>
<td>BRK</td>
<td>50000</td>
<td>0.0238</td>
<td>0.1283</td>
<td>0.1045</td>
<td>81.4</td>
</tr>
<tr>
<td>IDW</td>
<td>VFZ</td>
<td>BRK</td>
<td>50000</td>
<td>0.0356</td>
<td>0.1283</td>
<td>0.0927</td>
<td>72.3</td>
</tr>
<tr>
<td>NN</td>
<td>VFZ</td>
<td>BRK</td>
<td>50000</td>
<td>0.0445</td>
<td>0.1283</td>
<td>0.0838</td>
<td>65.3</td>
</tr>
</tbody>
</table>

Table 3.2: Comparison of average interpolation MAE (approximately 10 trial runs) for each interpolation method in the present work, using 50000 points in the definition of the VFZ and GBEs obtained by evaluating the BRK validation function ([1]) at these points. A constant model (Cst, Avg MAE), whose value was chosen to be the mean of the input GBE was used as a control. The last two columns represent the reduction (↓) in MAE in absolute units of Jm$^{-2}$ and % relative to the control model, respectively.

<table>
<thead>
<tr>
<th>Method</th>
<th>Distance</th>
<th>Dataset</th>
<th># GBs</th>
<th>MAE (Jm$^{-2}$)</th>
<th>Cst, Avg MAE (Jm$^{-2}$)</th>
<th>MAE ↓ (Jm$^{-2}$)</th>
<th>MAE ↓ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPR</td>
<td>VFZ</td>
<td>BRK</td>
<td>50000</td>
<td>0.0145</td>
<td>0.0955</td>
<td>0.081</td>
<td>84.8</td>
</tr>
<tr>
<td>Barycentric</td>
<td>VFZ</td>
<td>BRK</td>
<td>50000</td>
<td>0.0145</td>
<td>0.0955</td>
<td>0.081</td>
<td>84.8</td>
</tr>
<tr>
<td>IDW</td>
<td>VFZ</td>
<td>BRK</td>
<td>50000</td>
<td>0.0225</td>
<td>0.0955</td>
<td>0.073</td>
<td>76.4</td>
</tr>
<tr>
<td>NN</td>
<td>VFZ</td>
<td>BRK</td>
<td>50000</td>
<td>0.0307</td>
<td>0.0955</td>
<td>0.0648</td>
<td>67.9</td>
</tr>
</tbody>
</table>

3.1.1 Constant-Valued Control Models

To aid in objective interpretation of the error metrics, comparison is made to a constant valued control model, whose value is chosen to be the average of $y$ (approximately 1.16 Jm$^{-2}$ in the limit of $n_{\text{inputpts}} \to \infty$) resulting in RMSE and MAE values of approximately 0.1283 and 0.0955 Jm$^{-2}$. This comparison with the relevant constant-valued function gives a sense of the complexity and variability of the validation function and allows for a more objective comparison between differing works. For example, the RMSE for the relevant constant function compared
to the validation function employed for the ANN interpolation method in [2] is 0.0854 J m$^{-2}$; in contrast, the RMSE for the relevant constant function compared to the BRK validation function used in this work is 0.1302 J m$^{-2}$ (see Table 3.1). This suggests that the BRK validation function is more complex and therefore less well approximated by a constant than the validation function used to test the ANN interpolation method in [2]. Consequently, the improved performance of the present methods (see Section 3.3.1 and Tables 3.5 and 3.6) is even more notable in that the validation function employed here is more difficult to interpolate.

### 3.1.2 Experimental and Simulation Error

To give further context to the results of this and prior works, it is useful to consider what the intrinsic error is for typical GB property data. This provides an idea of the minimum possible interpolation error, since one cannot reliably detect lower error in the interpolation than already exists in the observed data itself.

One such estimate for error is furnished by the work of [68], who introduced a non-discretizing approach to extract relative GB energies from polycrystalline samples using the locally optimal block preconditioned conjugate gradient (LOBPCG) method. Their approach utilizes regularization imposed on triple junction (TJ) equilibrium equations and k-nearest neighbor (kNN) distances. Using 60000 TJs (180000 GBs) and a custom, non-smooth validation function they obtained GBE RMSE values of 0.0076 J m$^{-2}$ and 0.0277 J m$^{-2}$ for GBE values greater than 0.9 J m$^{-2}$ and less than 0.9 J m$^{-2}$, respectively. This suggests that an optimistic estimate for the error in noise-free$^1$ experimental GBE data obtained using such a method is on the order of 0.0076 J m$^{-2}$ to 0.0277 J m$^{-2}$, which also serves as an estimate of the minimum achievable noise-free experimental interpolation error for any of the interpolation methods described here. Similar analysis for noisy 0 K MS simulation data is provided in Section 3.3 and Section 4.4.2 giving a RMSE and MAE of 0.06529 J m$^{-2}$ and 0.06190 J m$^{-2}$, respectively.

$^1$These errors are based on Figure 8 from [68], which employed synthetic triple junctions with a custom validation function, rather than experimental data. While the authors did also consider the addition of noise, we use the noise-free results as an estimate of the best-case scenario.
Again comparing the relevant constant-valued control model\(^2\) to the validation function employed by [68], we calculate a RMSE and MAE of 0.0976 J m\(^{-2}\) and 0.0466 J m\(^{-2}\), respectively. This implies that the validation model used by [68] is also simpler\(^3\) than the BRK validation model employed in the present work.

### 3.1.3 Our Four Interpolation Methods

As shown in Tables 3.1 and 3.2, of the four interpolation methods from this work, GPR has the lowest error, both in terms of RMSE and MAE, while NN has the highest error. Compared to a constant valued control model, GPR interpolation reduced the prediction RMSE by 83\%, which outperforms all of the interpolation methods in this work with respect to accuracy, as well as those considered from the literature. After GPR the next most accurate methods are barycentric, IDW, and NN. It is worth noting that the RMSE interpolation error for the GPR and barycentric methods is comparable to the minimum achievable noise-free experimental interpolation error (the estimated error in experimental data obtained via the LOBPCG analysis mentioned previously). This is even more significant because the BRK validation function used in the present work is more complex and difficult to interpolate than that used in [68].

The accuracy of the predictions made using the VFZ methods depends on the VFZO set size and distribution. Figure 3.2 compares the prediction accuracy for each of the 4 methods to the constant valued control model, as a function of the number of input VFZOs (\(n_{\text{inputpts}}\)). As expected, higher density VFZO sets result in lower error, but eventually give diminishing returns. Moreover, the standard deviations produced via multiple runs are tightly constrained and generally shrink as the VFZO set size increases.

GPR consistently gives lower error than the other three interpolation methods for all VFZO set sizes. NN interpolation produces the worst error of the four methods, but is better than a constant valued control model (i.e. average of the input GBEs) so long as \(n_{\text{inputpts}}\) exceeds a few hundred input points.

\(^2\)We use the mean of the true GBEs from their validation function to define the constant-valued control model instead of the mean of the input GBEs because the latter does not exist for polycrystalline data.

\(^3\) [68] used 8 cusps of varying depths and widths based on the Read-Shockley model and unity GBE everywhere else.
Figure 3.2: (a) Average RMSE and (b) average MAE vs. number of input points for (planar) barycentric (blue), GPR (orange), IDW (yellow), and NN (purple) interpolation for approximately 10 random runs with different input and prediction points. Standard deviations of approximately 10 runs are also included. Compare with approximately 0.1283 Jm$^{-2}$ and 0.0955 Jm$^{-2}$ RMSE and MAE, respectively, for a constant, average model (green) using the average of the input properties (approximately 1.16 Jm$^{-2}$).

It is worthwhile to note that both GPR and IDW are kernel-based in that a model parameter controls the size of the region that can influence the interpolation results. In the GPR case, this is automatically calculated via an internal fitting routine of fitrgp(). NN distance distributions (Figure 2.2) can lead to insight about correlation lengths in a given VFZO set and are used in the IDW implementation. For IDW, the radius of influence is set to $r = \sqrt{2} \mu$, where $\mu$ is the mean NN distance. It is likely that better tuning of the kernel parameters in these two methods (such as use of built-in hyperparameter optimization in the case of fitrgp()) could further decrease their interpolation errors. Additionally, for GPR, use of the ‘exact’ predictMethod or a larger 'fic' set size will also likely reduce interpolation error.

By contrast, barycentric interpolation automatically adjusts its effective region of influence because the size of the simplices in the mesh decreases as the number of vertices increases. More uniformly distributed meshes (such as obtained via constrained optimization [69, 70]) will likely result in lower, more uniform interpolation error, especially for this simplex-based approach which
Table 3.3: Approximate coordinates of VFZOs A and B used for the interpolation in Figure 3.3. Individual quaternions of each octonion are given in the active sense and in the laboratory reference frame with an assumed GB normal pointing in the +z direction, also in the laboratory reference frame.

<table>
<thead>
<tr>
<th>Octonion</th>
<th>o(1)</th>
<th>o(2)</th>
<th>o(3)</th>
<th>o(4)</th>
<th>o(5)</th>
<th>o(6)</th>
<th>o(7)</th>
<th>o(8)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.8658</td>
<td>-0.4269</td>
<td>-0.1270</td>
<td>0.2280</td>
<td>0.2810</td>
<td>0.8390</td>
<td>-0.3852</td>
<td>0.2622</td>
</tr>
<tr>
<td>B</td>
<td>0.4684</td>
<td>-0.7657</td>
<td>-0.4100</td>
<td>-0.1617</td>
<td>-0.1483</td>
<td>0.8204</td>
<td>-0.3588</td>
<td>0.4198</td>
</tr>
</tbody>
</table>

can exhibit high-aspect ratio facets and non-intersections outside the bounds of the mesh (Figure 4.4). While the barycentric interpolation error is always higher than GPR for the considered set sizes, at 50000 VFZOs, the errors of GPR and barycentric interpolation are nearly identical.

Figure 3.3: Predictions of GPR (blue circles), barycentric (red circles), NN (magenta circles), and IDW (green circles) as a function of distance along a 1D arc ($\overline{AB}$) between two VFZOs ($A$ and $B$). The true, underlying BRK function is also shown (black line). 50000 random input VFZOs were generated and used for each of the models. 150 equally spaced points between $A$ and $B$ obtained via oSLERP [7] were used as prediction points. GPR uncertainty standard deviation is plotted as shaded error band.
3.1.4 Plotting Grain Boundary Energy between Arbitrary GBs

To provide a visual illustration of the property predictions, Figure 3.3 shows the predicted GBE for each of the four interpolation methods as a function of distance along a 1D arc (\(AB\)) between two VFZOs, \(A\) and \(B\). Approximate coordinates for \(A\) and \(B\) are given in Table 3.3, and each intermediate point between \(A\) and \(B\) resides on the surface of a hypersphere. The 150 intermediate points were obtained using oSLERP \([7]\). Each model used its own set of 50000 random input VFZOs with GBE sampled via the BRK validation function. The two VFZOs were chosen by taking the furthest apart pair out of 20000 VFZOs which thus approximates the largest dimension of the VFZ where each endpoint is close to the true VFZ exterior.

Comparison of the predictions from the four interpolation methods with the true values of the BRK validation function along this 1D path shows that all methods yield reasonable agreement with the true model. The GPR and barycentric methods appear to agree most with the true model, followed by IDW and NN. The NN method shows the piecewise-constant (stair-step) artifact typical of NN methods. We also note that while the fidelity of the predictions is quite good for all methods in the interior of the VFZ, the performance does degrade at the extreme limits of the VFZ (note the deviations at the left and right limits of Figure 3.3). This effect seems to be particularly pronounced for the barycentric method, and much less so for the GPR method.

We believe this is the first\(^4\) plot of a GB property continuously interpolated between two arbitrary GBs (i.e. neither residing entirely in a single misorientation fundamental zone (MFZ) nor a single boundary plane fundamental zone (BPFZ)). Such visualizations can naturally be extended to 2D and 3D by plotting colored points in a triangle or tetrahedron, respectively, all of which (1D, 2D, and 3D) represent small “slices” of the GBC space.

Additionally, such visualizations may suggest the ability to estimate numerical derivatives or gradients of GB properties without being restricted to a GB subspace (e.g. MFZ or BPFZ) which can be a useful mathematical construct for the GB community. For example, steepest descent paths and all local GBE minima can be estimated and used in grain growth simulations. In such contexts, use of ensembled VFZO interpolation may be necessary to prevent discontinuity artifacts when

\(^4\) OSLERP results from [7] plots GB structure continuously between two GBs, and [5] performs cross-validation on the simulated Olmsted Ni GBs. The results in both [7] and [5] are distinct from what is presented here: a plot of continuously interpolated GBEs between two arbitrary GBs.
Table 3.4: Comparison of average runtime (s) for 10 trials for barycentric, GPR, IDW, and NN interpolation methods for various input VFZO set sizes using 12 cores and evaluated on 10000 prediction VFZOs. Because GPR, IDW, and NN method defaults do not use parfor loops but may have internal multi-core vectorization, it is unclear to what extent the number of cores affects the runtime of methods other than barycentric interpolation. VFZO symmetrization runtime was not included; however, symmetrization of 50000 GBOs takes approximately 76 seconds on 6 cores (Intel i7-10750H, 2.6 GHz) and is a common step in every interpolation method (i.e. it is fundamental to the VFZO framework) We used the BRK validation function for GBE [1].

<table>
<thead>
<tr>
<th>VFZO Set Size</th>
<th>Barycentric</th>
<th>GPR</th>
<th>IDW</th>
<th>NN</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>191.8 ± 19.57</td>
<td>0.4187 ± 0.4342</td>
<td>0.034 ± 0</td>
<td>0.0367 ± 0.0041</td>
</tr>
<tr>
<td>388</td>
<td>388.4 ± 18.84</td>
<td>0.943 ± 0.3481</td>
<td>0.0904 ± 0.0224</td>
<td>0.0705 ± 0.0129</td>
</tr>
<tr>
<td>500</td>
<td>455.7 ± 55.28</td>
<td>0.6104 ± 0.3138</td>
<td>0.1352 ± 0.0364</td>
<td>0.0724 ± 0.0051</td>
</tr>
<tr>
<td>1000</td>
<td>536.5 ± 35.26</td>
<td>1.743 ± 0.9464</td>
<td>0.1948 ± 0.0395</td>
<td>0.1203 ± 0.0184</td>
</tr>
<tr>
<td>5000</td>
<td>998.9 ± 54.48</td>
<td>5.216 ± 0.4816</td>
<td>0.8726 ± 0.1529</td>
<td>0.9277 ± 0.2418</td>
</tr>
<tr>
<td>10000</td>
<td>1516 ± 56.59</td>
<td>5.609 ± 0.8756</td>
<td>1.631 ± 0.3915</td>
<td>0.8938 ± 0.1717</td>
</tr>
<tr>
<td>20000</td>
<td>2526 ± 119.5</td>
<td>11.45 ± 3.29</td>
<td>3.191 ± 0.4752</td>
<td>1.275 ± 0.3423</td>
</tr>
<tr>
<td>50000</td>
<td>5743 ± 361.3</td>
<td>13.69 ± 4.05</td>
<td>7.635 ± 1.872</td>
<td>3.817 ± 0.5884</td>
</tr>
</tbody>
</table>

crossing the exterior of a VFZ as discussed in Section 2.1.3 which we plan to explore in future work.

3.2 Interpolation Efficiency

Below, we present interpolation efficiency results in terms of computational runtime and memory for the four interpolation schemes used in this work (Section 3.2.1). Additionally, an in-depth treatment of the improved symmetrization runtime (separate from interpolation runtime) relative to the original octonion metric is also given (Section 3.2.2).

3.2.1 Comparison of Methods

We discuss runtime and memory requirements for barycentric, GPR, IDW, and NN interpolation methods. Computational runtimes of the various methods are shown in Table 3.4.
Barycentric interpolation takes the longest, in spite of the fact that it is the only parallelized method by default (not accounted for in Table 3.4). In other words, since 12 cores were used to obtain these runtime results, the total runtime across all cores is much higher compared with the other methods; however, it is possible that other methods used multi-threading via built-in vectorized functions. The long computation times of barycentric interpolation result primarily from the large number of facets present in a high-dimensional mesh triangulation and the interconnectedness of facets with respect to each other.

GPR is fast and has lower error compared to barycentric interpolation; however, the entire process has to be reevaluated (in the current implementation) if the input points (i.e. VFZOs) or input property values (i.e. GBEs) change (typically referred to as predictors/features and responses, respectively, in the machine learning community). On the other hand, barycentric interpolation is fast if the triangulation and intersections are pre-computed and only input property values change (see interp_bary_fast.m), but slow if the input or prediction points change, which requires recomputing the triangulation and intersections. Additionally, GPR is the second-longest in terms of runtime, but is more accurate than any of the other three methods.

NN and IDW interpolation have vectorized implementations and are much simpler than the barycentric and GPR methods. Consistent with expectations, NN and IDW exhibit almost negligible runtimes; however, this is at the expense of increased error, as discussed earlier (Section 3.1). It should also be noted that barycentric interpolation has much higher memory requirements than GPR, NN, and IDW due to the need to store large matrices. If PredictMethod = 'exact' in fitrgp(), then GPR also has high memory requirements for large VFZO sets. For 50000 input points with sufficient RAM (e.g. ~32 GB) and 12 cores available, the 'exact' method runtime is (535.1 ± 392.6) seconds. However, because the 'fic' approximation is always used in this work, memory requirements are similar to NN and IDW.

Because the default implementation of IDW uses a radius cut-off, the distance and weight matrices can be stored as sparse objects, dramatically reducing both the final memory storage requirements and computational complexity of this method We expect that a kNN approach would produce similar results both in terms of runtime and error when a relatively uniform sampling of GBC is obtained.
3.2.2 Symmetrization Runtime Comparison with Traditional Octonion Metric

In addition to the interpolation runtime of the methods just presented, it is valuable to consider the runtime of the VFZ symmetrization step (not included in Table 3.4). The symmetrization step is at the core of the VFZO framework and is a key to its overall performance. It precedes and is a common step for both (i) distance calculations and (ii) all of the interpolation methods presented here.

Directly computed, scaled Euclidean and arc length distances in the VFZO framework approximate the original octonion distance by [7], and the calculation speed is even higher than explicit GBO distance calculations using the original octonion distance. For example, 50000 GBOs can be symmetrized into VFZOs in approximately 76 seconds using 6 cores (e.g. via VFZO repository function `get_octpairs.m`), and the corresponding 50000 × 50000 pairwise-distance matrix can be computed in approximately 10 seconds using the built-in MATLAB Statistics and Machine Learning Toolbox function `pdist.m`, giving a total runtime of approximately 86 seconds (466 total CPU seconds). Compared to the original octonion metric distance calculations [5] in the Fortran-based EMSoft package [71], this represents an improvement in computational speed by ∼5 orders of magnitude using our MATLAB implementation in the VFZO repository [65].

Improvement per distance calculation per core of the VFZO repository is about $4 \times 10^5$ relative to the EMSoft [71] metric of 26 minutes using 8 cores for a 388 × 388 pairwise distance matrix. This EMSoft timing information is directly reported in [5]. In other words, computation of a 50000 × 50000 using the traditional octonion metric and EMSoft implementation would take approximately 6.6 CPU years (or 153 CPU days if one GB in the GB pair is fixed according to the assumption in [53]). Since most interpolation methods will depend on computing new distances, probing the model at new GBs will also be expensive. For example, it would take at minimum ∼60 CPU days to perform property interpolation for 10000 prediction GBs assuming the pairwise-distance matrix relative to 50000 input GBOs need to be computed.

This significant speed up stems from the fact that in the VFZO framework SEOs only need to be considered once per GB, $O(L)$, rather than once per distance calculation, $O(L^2)$, and that SEOs only need to be considered once in a GB pair, $O(N_p^2)$, rather than for every combination between the two GBs, $O(N_p^4)$. The SEO computation complexity is thus $O(N_p^2 L^2)$, a significant improvement compared with the original SEO complexity of $O(N_p^4 L^2)$ [5], where $N_p$ is the cardi-
nality of the crystallographic point group \( N_p = 24 \) for \( m\bar{3}m \) FCC point group) and \( L \) is the number of GBs.

Empirically, to compute a pairwise-distance matrix for \( L = 50000 \) GBs using the VFZO repository [65], the full \( O(N_p^2 L) \) symmetrization operations take about 76 seconds \( \times \) 6 cores = 456 seconds of CPU time, whereas the subsequent pairwise-distance computation is \( O_{pd}(L^2) \) and takes approximately 10 seconds for a 50000 \( \times \) 50000 matrix. Even though \( O(N_p^2 L) \ll O_{pd}(L^2) \), the symmetrization step takes far more time than the pairwise distance calculation (even for large \( L \)) because of the cost of generating SEOs. Because Euclidean distances—which can be computed faster than trigonometric inverse functions—are employed, and built-in, vectorized MATLAB functions are utilized, there is a further speed enhancement in the VFZO approach.

3.3 Literature Datasets

In addition to the validation results just provided, we also apply the VFZO framework to real GB property data from several sources in the literature. This allows comparison to previous methods as well as demonstration of the performance of the the VFZO framework for typical MS data. Specifically, we present GPR interpolation results for MS Fe and Ni simulation datasets and compare them with prior work (Section 3.3.1). Finally, because GPR overestimates the low GBE for the non-uniformly distributed, noisy Fe simulation dataset, we also provide results for an adaptation called the GPRM model that compensates for this effect.

3.3.1 Comparison with Prior Work

The GPR interpolation method of the present work was used with the same number of input GBs as was supplied in [2] for Fe (17176) and [5] for Ni (388) to provide a more consistent comparison with prior work. For Fe, the remainder of the simulation data was used for testing, consistent with [2], except that zero-energy GBs and degenerate GBs were treated differently as described in Section 2.5.1. For Ni, a leave-one-out cross validation (LOOCV) scheme was used, consistent with [5].

Hexagonally binned parity plots for the Fe and Ni simulation datasets are shown in Figure 4.5d and Figure 4.9, respectively. RMSE and MAE comparisons along with improvement
Table 3.5: Comparison of interpolation MAE (1 trial run) for 0 K molecular statics (MS) datasets. A constant model (Cst, Avg MAE), whose value was chosen to be the mean of the input GBE was used as a control. The last two columns, MAE $\downarrow$ (J m$^{-2}$ and MAE $\downarrow$ (%)), represent the reduction in MAE in units of J m$^{-2}$ and % relative to the control model, respectively. Non-sym refers to distances calculated in [2] without regard for crystal symmetries.

<table>
<thead>
<tr>
<th>Method</th>
<th>Distance</th>
<th>Dataset</th>
<th># GBs</th>
<th>MAE (J m$^{-2}$)</th>
<th>Cst, Avg MAE (J m$^{-2}$)</th>
<th>MAE $\downarrow$ (J m$^{-2}$)</th>
<th>MAE $\downarrow$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPR</td>
<td>VFZ</td>
<td>MS Fe</td>
<td>17176</td>
<td>0.0405</td>
<td>0.0617</td>
<td>0.0212</td>
<td>34.4</td>
</tr>
<tr>
<td>ANN [2]</td>
<td>Non-sym</td>
<td>MS Fe</td>
<td>17176</td>
<td>0.0486</td>
<td>0.0617</td>
<td>0.0131</td>
<td>21.2</td>
</tr>
<tr>
<td>LKR [5]</td>
<td>GBO</td>
<td>MS Ni</td>
<td>388</td>
<td>—</td>
<td>0.1752</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

Table 3.6: Comparison of interpolation RMSE (1 trial run) for 0 K molecular statics (MS) datasets. A constant model (Cst, Avg RMSE), whose value was chosen to be the mean of the input GBE was used as a control. The last two columns, RMSE $\downarrow$ (J m$^{-2}$ and RMSE $\downarrow$ (%)), represent the reduction in RMSE in units of J m$^{-2}$ and % relative to the control model, respectively. Non-sym refers to distances calculated in [2] without regard for crystal symmetries.

<table>
<thead>
<tr>
<th>Method</th>
<th>Distance</th>
<th>Dataset</th>
<th># GBs</th>
<th>RMSE (J m$^{-2}$)</th>
<th>Cst, Avg RMSE (J m$^{-2}$)</th>
<th>RMSE $\downarrow$ (J m$^{-2}$)</th>
<th>RMSE $\downarrow$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANN [2]</td>
<td>Non-sym</td>
<td>MS Fe</td>
<td>17176</td>
<td>—</td>
<td>0.0854</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>GPR</td>
<td>VFZ</td>
<td>MS Ni</td>
<td>388</td>
<td>0.0951</td>
<td>0.2243</td>
<td>0.1292</td>
<td>57.6</td>
</tr>
<tr>
<td>LKR [5]</td>
<td>GBO</td>
<td>MS Ni</td>
<td>388</td>
<td>0.0977</td>
<td>0.2243</td>
<td>0.1266</td>
<td>56.4</td>
</tr>
</tbody>
</table>

relative to a constant, average model are given in Table 3.5 and Table 3.6, respectively. For the Fe case, we see a larger improvement than prior work likely due to the incorporation of GB symmetry, which was not considered in [2]. For the Ni case, there is a slight improvement relative to prior work, indicating that accuracy is similar to the original octonion metric while maintaining the significant computational benefits of the VFZO framework.

3.3.2 Gaussian Process Regression Mixture Applied to Fe Simulation Data

In addition to GPR, a GPRM model (Figure 4.5) based on a sigmoid mixing function (Figure 4.6) is used to better predict low GBE values of the non-uniformly distributed, noisy Fe dataset (Section 2.5.3). GPRM interpolation results for the Fe GBE simulations [8] are shown in Fig-
Figure 3.4: Interpolation results for a large Fe simulation database [8] using 46,883 input GBs and 11,721 prediction GBs in an 80%/20% split and a GPRM model to better approximate low GBEs. Use of a GPRM model predicts low GBE better than the standard GPR model (compare with Figure 4.5d). (a) Hexagonally binned parity plot of the GPR mixing model with RMSE and MAE of 0.055035 Jm$^{-2}$ and 0.039185 Jm$^{-2}$, respectively, relative to typical, constant average models of 0.0854 Jm$^{-2}$ and 0.0617 Jm$^{-2}$, respectively. (b) Predictions of GPRM model (blue circles) as a function of distance along a 1D arc ($\overline{AB}$) between two VFZOs ($A$ and $B$).

Figure 3.4, where approximate coordinates for the octonions $A$ and $B$ in Figure 3.4b are given in Table 3.7. We find that:

- the model error is on par with the intrinsic error of the data
- the predictions likely exhibit overprediction bias relative to the true minimum for a given GB
- future availability of multiple metastable state GBEs is anticipated to greatly improve the model performance

We now elaborate each of these points.

First, because only a single metastable state was used for each GBE simulation, both the training and validation data are subject to noise, consistent with a wide lateral spread of predictions in both Figure 3.4 and the intrinsic error estimation (Figure 4.8). The Fe simulation dataset GPRM model gives lower RMSE (0.055035 Jm$^{-2}$) and MAE (0.039185 Jm$^{-2}$) than the intrinsic
error estimates. This indicates that the intrinsic error itself is somewhat overestimated\(^5\). The fact that both model and intrinsic error metrics are relatively close and the prediction and intrinsic error parity plots (Figure 3.4 and Figure 4.8b, respectively) are similar suggests that the model is performing well. It also suggests that further improvements in the model relative to the “true” values will be “hidden”, i.e. they will probably not manifest as lower RMSE or MAE nor as more tightly distributed parity plots, etc.

Next, given the theoretical existence of a true minimum GBE for a given GB, the predictions which were based on metastable GBEs can be assumed to have an overprediction bias relative to the true minimum. On average, we expect this overprediction bias relative to the true minimum GBE (rather than the most likely metastable state) may be on the order of a few hundred mJ m\(^{-2}\) and may vary as a function of true minimum GBE. In other words, the model obtained is probably an estimate of the most likely metastable GBE rather than the true minimum GBE. This is akin to saying that we obtain from this data a model that approximates the non-equilibrium, Stillinger quenched red curve of Figure 4(c1) in [33], not the minimum GBE blue curve of the same chart. See [33] for an in-depth treatment of equilibrium and metastable GBE.

Finally, datasets where multiple metastable GBEs (e.g. 3-10 repeats) are provided for each GB will likely greatly improve the performance of the GPR model in predicting either the most likely metastable GBE (when all GBEs are considered) or the true minimum GBE (when only the minimum GBE is considered for each GB) and may even negate the need for a GPRM approach. Thus, it is suggested that, where feasible, future large-scale GB bicrystal simulation studies will report all property data for repeated trial runs rather than a single trial run or a single value from

\(^{5}\)The prediction error of a model typically cannot be less than the noise of the prediction data of a model even if the model is estimating the true prediction values with better accuracy than the noise (which is very possible and even expected with GPR models when the noise in the input data is approximately Gaussian).
a set of trial runs. Ideally, data for the three additional microscopic DOFs for GBs (which falls into the category of epistemic uncertainty in this work) would also be included. We believe it is likely that minimum energy paths (i.e. paths of steepest descent) in the GBE landscape depend on both macroscopic and microscopic DOFs (in total, 8DOF) and could offer a more holistic view of GB behavior that better mimics and explains experimental grain growth observations. Indeed, it has been experimentally observed that at least some GB migration mechanisms involve structural transformations between equilibrium GBs via metastable states [34].
CHAPTER 4. SUPPLEMENTAL INFORMATION

4.1 Euclidean and Arc Length Distances

The close correlation between Euclidean and arc length distances in the VFZO sense is shown in Figure 4.1 using pairwise distances of 10000 VFZOs. This justifies the use of Euclidean distance as an approximation of hyperspherical arc length (and by extension, that a scaled Euclidean distance approximates a non-symmetrized octonion distance, see Eqs. (2.1)–(2.3) of the main paper). However, comparison with the original octonion metric [7] gives overestimation for some boundaries. This is an inherent feature of the VFZO framework that can be addressed via use of the ensemble methods described in Section 2.1.3 (see also Figures 2.3 and 2.4).

4.2 Ensemble Interpolation Results

Ensemble interpolation is a classic technique that can be used to enhance predictive performance of models. Here we describe the methods (Section 4.2.1), results (Section 4.2.2), and the potential of integrating ensemble interpolation with a GPRM scheme (Section 4.2.3).

4.2.1 Methods

VFZO ensemble\(^1\) interpolation occurs by:

1. generating multiple reference octonions to define multiple VFZs

2. obtaining multiple VFZO representations for a set of GBs based on the various reference octonions

3. performing an interpolation (e.g. GPR) for each of the representations

\(^1\)Ours is a “bagging”-esque ensemble scheme because the same interpolation method (GPR) is used but with different representations for the input data.
Figure 4.1: Parity plot of 8D Cartesian hyperspherical arc length vs. 8D Cartesian Euclidean distance for pairwise distances in a \( (m^3m) \) symmetrized set of 10000 randomly sampled VFZOs. The max arc length is approximately 0.58 rad, indicating a max octonion distance of approximately 1.16 rad or 66.5° between any two points in the VFZ. The close correlation between arc length and Euclidean distance supports the validity of using Euclidean distance instead of arc length in the interpolation methods. This is separate from the correlation between VFZO Euclidean or arc length distances with the traditional octonion distance [5].

4. homogenizing the ensemble of models (e.g. by taking the mean or median of the various models)

4.2.2 Results

Use of an ensemble interpolation scheme decreases interpolation error for a GPR model with 50000 input and 10000 prediction VFZOs. By using an ensemble size of 10 (i.e. 10 GPR models each with different reference octonions and therefore different VFZs), RMSE and MAE decreased from 0.0241 J m\(^{-2}\) and 0.0160 J m\(^{-2}\) to 0.0187 J m\(^{-2}\) and 0.0116 J m\(^{-2}\), respectively, using the median homogenization function (Figure 4.2).

Figure 4.3 shows the hexagonally binned parity plots for predictions made using the mean, median, minimum, and maximum predicted values over an ensemble of 10 VFZs. Qualitatively,
the ensemble mean and ensemble median parity plots look similar to those from the main text (Figure 3.1), though the distributions of the ensemble scheme are somewhat tighter. The ensemble minimum produces better predictions of low GBE than any of the other models, but underestimates high GBE as expected. Naturally, the ensemble maximum overestimates in general. Diminishing returns manifest in Figure 4.2 for mean and median homogenizations. This is to be expected because the original octonion distances [7] are well-approximated using an ensemble size of 10 (Figure 2.3c and Figure 2.4).

4.2.3 Possibility: Combining Ensemble with Gaussian Process Regression Mixture

A scheme which preferentially favors the ensemble minimum for low GBE predictions and defaults to ensemble mean or median for all other GBEs may produce even better results across the full range of GBEs. For example, this could be accomplished by combining the ensemble scheme described here with the GPR mixture model described in Section 4.4.1.
**4.3 Barycentric Interpolation**

**4.3.1 High-Aspect Ratios**

An artifact of the barycentric interpolation method which occurs due to the presence of high-aspect ratio facets is shown in Figure 4.4. As the dimensionality increases for a constant number of points and from the numerical tests, the rate of missed facet intersections increases. This artifact and our method for addressing it are discussed in Appendix B.2 of the main text.
4.4 Kim Interpolation

A GPR mixing model is developed to accommodate the non-uniformly distributed, noisy Fe simulation data [8] and better predict low GBE. Details of the method (Section 4.4.1) and an analysis of the input data quality (Section 4.4.2) are given. The code implementation is given in gprmix.m and gprmix_test.m of the VFZO repository [65].

4.4.1 Details of Gaussian Process Regression Mixture

As shown in Figure 4.5a, prediction using the standard approach of the main document (termed the $\epsilon_1$ model) overestimates low GBEs for this dataset. By training the model on only GBs with a GBE less than $1.2 \text{Jm}^{-2}$ (termed the $\epsilon_2$ model) and by using an exponential
Figure 4.5: (a) Hexagonally binned parity plot of the standard GPR model. (b) All prediction GBs based on the model using only training GBs with a GBE less than 1.2 Jm$^{-2}$. (c) Combined disjoint model as explained in the text. (d) Hexagonally binned parity plots of the final GPR mixing model. Points in (c) are produced by splitting the prediction data into less than and greater than 1.2 Jm$^{-2}$. A sigmoid mixing function (Figure 4.6) is then applied where the predicted GBEs shown in (c) determines the mixing fraction ($f$) to produce a weighted average of models (a) and (b). A large Fe simulation database [8] using 46883 training datapoints and 11721 validation datapoints in an 80%/20% split. The GPR mixture model decreases error for low GBE and changes overall RMSE and MAE from 0.057932 Jm$^{-2}$ and 0.039857 Jm$^{-2}$ in the original model (shown in (a)) to 0.057502 Jm$^{-2}$ and 0.041272 Jm$^{-2}$ (shown in (d)), respectively.

(KernelFunction='exponential') rather than a squared exponential kernel, prediction of low GBEs improves, but naturally underestimation occurs for higher GBEs (Figure 4.5b).
A combined, disjoint model (Figure 4.5c) is taken ($\varepsilon_3$) by replacing $\varepsilon_1$ GBE predictions for GBs with GBE less than 1.2 J m$^{-2}$ with the corresponding $\varepsilon_2$ predictions. Finally, a weighted average (Eq. (4.1)) is taken according to:

$$\varepsilon_{\text{mix}} = f\varepsilon_1 + (f - 1)\varepsilon_2$$

where $\varepsilon_1$ and $\varepsilon_2$ represent the standard GPR model and the GPR model trained on the subset of GBs with a GBE less than 1.2 J m$^{-2}$, respectively, and $f$ is the sigmoid mixing fraction given by:

$$f = \frac{1}{e^{-m(\varepsilon_3-b)} + 1}$$

and shown in Figure 4.6 with $m = 30$ and $b = 1.1$ J m$^{-2}$, as used in this work. Larger values of $m$ yield a steeper sigmoid function and larger values of $b$ shift the sigmoid function further to the right. Specific values for $m$ and $b$ were chosen by visual inspection and trial and error. This results in a GPR mixing model which better predicts low GBEs while retaining overall predictive accuracy (Figure 4.5d).
Uncertainty of the GPR mixing model is similarly obtained by taking a weighted average of the uncertainties of each model according to:

$$\sigma_{\text{mix}} = f \sigma_1 + (f - 1) \sigma_2$$  \hspace{1cm} (4.3)$$

where $\sigma_1$ and $\sigma_2$ are the corresponding uncertainties of $\varepsilon_1$ and $\varepsilon_2$, respectively, and $f$ is given by Eq. (4.2).

### 4.4.2 Input Data Quality

Of the $\sim 60000^2$ GBs in [8], $\sim 10000$ GBs were repeats that were identified by converting to VFZOIs and applying VFZO repository function \texttt{avg\_repeats.m}. In [8], mechanically selected GBs were those which involved sampling in equally spaced increments\(^3\) for each 5DOF parameter, and a few thousand intentionally selected GBs (i.e. special GBs) were also considered. Of mechanically and intentionally selected GBs, 9170 and 112 are repeats, respectively, with a total of 2496 degenerate sets\(^4\) (see Figure 4.7 for a degeneracy histogram). Thus, on average there is a degeneracy of approximately four per set of degenerate GBs.

By comparing GBE values of (unintentionally\(^5\)) repeated GBs in the Fe simulation dataset [8], we can estimate the intrinsic error of the input data. For example, minimum and maximum deviations from the average value of a degenerate set are $-0.2625 \text{ Jm}^{-2}$ and $0.2625 \text{ Jm}^{-2}$, respectively, indicating that a repeated Fe GB simulation from [8] can vary by as much as $0.525 \text{ Jm}^{-2}$, though rare. Additionally, RMSE and MAE values can be obtained within each degenerate set by comparing against the set mean. Overall RMSE and MAE are then obtained by averaging and weighting by the number of GBs in each degenerate set. Following this procedure, we obtain an average set-wise RMSE and MAE of $0.06529 \text{ Jm}^{-2}$ and $0.06190 \text{ Jm}^{-2}$, respectively, which is an approximate measure of the intrinsic error of the data. Figure 4.8 shows histograms and parity plots of the intrinsic error. The overestimation of intrinsic error mentioned in the main text (Section 3.3)

---

\(^2\)The “no-boundary” GBs (i.e. GBs with close to $0 \text{ Jm}^{-2}$ GBE) were removed before testing for degeneracy.

\(^3\)In some cases, this was equally spaced increments of the argument of a trigonometric function.

\(^4\)A degenerate “set” is distinct from a VFZO “set”, the latter of which is often used in the main text.

\(^5\)To our knowledge, the presence of repeat GBs were not mentioned in [8] or [45]
could stem from bias as to what type of GBs exhibit repeats based on the sampling scheme used in [8] and/or that many of the degenerate sets contain a low number of repeats (Figure 4.7). Next, we see that by binning GBs into degenerate sets, most degenerate sets have a degeneracy of fewer than 5 Figure 4.7 We split the repeated data into sets with a degeneracy of fewer than 5 and greater than or equal to 5 and plot the errors (relative to the respective set mean) in both histogram form (Figure 4.8a and Figure 4.8c, respectively) and as hexagonally-binned parity plots (Figure 4.8b and Figure 4.8d, respectively). While heavily repeated GBs tend to give similar results, occasionally repeated GBs often have larger GBE variability. This could have physical meaning: Certain types of (e.g. high-symmetry) GBs tend to have less variation (i.e. fewer and/or more tightly distributed metastable states). However, it could also be an artifact of the simulation setup that produced this data (e.g. deterministic simulation output for certain types of GBs).
Figure 4.8: Degenerate GBs sets are split into those with a degeneracy of fewer than 5 and greater than or equal to 5 and plotted as (a) and (c), respectively) error histograms and (b) and (d), respectively) hexagonally-binned parity plots. Large degenerate sets tend to have very low error, whereas small degenerate sets tend to have higher error. In other words, GBs that are more likely to be repeated many times based on the sampling scheme in [8] tend to give similar results, whereas GBs that are less likely to be repeated often have larger variability in the simulation output. We do not know if this has physical meaning or is an artifact of the simulation setup.
Figure 4.9: (left) Hexagonally binned parity plot for Ni simulation grain boundary energy (GBE) interpolation using LOOCV. (right) Parity plot for leave-one-out cross validation (LOOCV) interpolation results reproduced from Figure 6a of [5] under CC-BY Creative Commons license.

4.5 Olmsted Interpolation

As illustrated in Figure 4.9, LOOCV interpolation results for 0 K MS low-noise Ni simulations using the GPR method are similar to LKR results reported in Figure 6a of [5] (reproduced on the right of Figure 4.9 for convenience).
CHAPTER 5. CONCLUSION

In this work, we presented the VFZO framework for (i) computing distances between GBs and (ii) predicting the properties of GBs from existing measurements. We found that distance calculations in the VFZO framework (via VFZO repository function GBdist4.m) are dramatically more computationally efficient ($O(N_p^2 L)$) than traditional methods ($O(N_p^4 L^2)$) at the expense of infrequent, large distance overestimation which can be addressed through ensemble techniques at a small computational cost (relative to traditional methods) as discussed in Section 2.1.3.

We also developed and tested a barycentric interpolation method, and adapted three other interpolation methods for use in the VFZO framework. We provide an easy-to-use, versatile implementation of these methods through an interpolation function interp5D0F.m written in MATLAB (https://github.com/sgbaird-5dof/interp, [65]) and many companion functions in the VFZO repository. This approach is general and applies to any crystal system (any of the 32 crystallographic point groups can be selected by the parameter pgnum). We also developed a GPRM model specifically for better low GBE prediction using a non-uniformly distributed, noisy dataset.

Of the interpolation methods that we present in this work, GPR provided the highest accuracy predictions. It also provided higher accuracy predictions than any of the methods in the literature. The GPR interpolation errors (50000 VFZOs) for the BRK validation model are about 2.4 times the intrinsic error that would be expected from reconstruction of noise-free, experimental polycrystalline data via LOBPCG [68] (180000 GBs) with their simpler validation model. Moreover, the interpolation errors for a Fe simulation dataset are on par with the intrinsic errors of the dataset itself (Section 4.4.2). While IDW and NN interpolation have the fastest computation times, they also have higher interpolation error. Consequently, we recommend the GPR interpolation method for the VFZO framework for most applications because it provides the best combination of accuracy and speed and handles input noise; however, the other methods can meet niche needs.
For example, barycentric interpolation enables rapid and accurate predictions when the function to be evaluated changes, but the input and prediction GBs remain fixed.

We anticipate that the VFZO framework and corresponding implementation will benefit numerous applications related to GB structure and properties, including facilitating GB structure-property model development, enabling efficient surrogate modeling of GB properties, and larger scale iterative simulations that require repetitive evaluation of computationally expensive structure-property models.
Bibliography


[23] Huang, W., Shishehbor, M., Guarín-Zapata, N., Kirchhofer, N. D., Li, J., Cruz, L., Wang, T., Bhowmick, S., Stauffer, D., Manimunda, P., Bozhilov, K. N., Caldwell, R., Zavattieri, P., and


49


51


[66] Chesser, I., 2019. GB Octonion Code. 16, 17


[71] De Graef, M., 2020. EMSoft. 28


APPENDIX A. ACTIVE VS. PASSIVE CONVENTION

Misorientation quaternions are represented in the active sense\(^1\):

\[
q_m = q_A^{-1} q_B
\]  

(A.1)

where \(q_m\), \(q_A\), and \(q_B\) represent the misorientation quaternion, orientation quaternion of grain A in the sample frame, and orientation quaternion of grain B in the sample frame, respectively. The \(^{-1}\) operator denotes a unit quaternion inverse (identical to conjugation of a unit quaternion). Quaternion multiplication is given by equation 23 of [72]

\[
pq \equiv (p_0q_0 - \mathbf{p} \cdot \mathbf{q}, q_0p + p_0q + P\mathbf{p} \times \mathbf{q})
\]  

(A.2)

where \(q_0\) and \(p_0\) are scalar components of the quaternions, and \(\mathbf{q}\) and \(\mathbf{p}\) are the vector components. In this work, we use the convention that \(P = 1\) throughout the various operations in the VFZO repository (\(P \equiv \epsilon_{ijk}\)) and highly encourage interested readers to refer to [72] to understand the redefined versions of quaternion multiplication, quaternion rotation, nuances associated with use of active vs. passive conventions, etc. BP unit normals are expressed pointing away from grain A and in the reference frame of grain A (i.e. the outward-pointing normal convention).

\(^1\)The passive convention is used in [7]
we describe barycentric interpolation applied in the VFZO framework in more detail. This includes:

1. triangulation of a VFZ mesh (Appendix B.1)

2. finding intersections between arbitrary VFZOs and the VFZ mesh (i.e. finding intersecting facets) (Appendix B.2)

3. calculating interpolated values of an arbitrary VFZO property using the intersecting facet (Appendix B.3)

B.1 Triangulating a Voronoi Fundamental Zone Mesh

Creation of a simplicial mesh is necessary to perform barycentric interpolation. Due to the difficulty of visualizing a 7-sphere, we provide visual illustrations of the process as applied to lower-dimensional analogues. After GBOs have been symmetrized into a VFZ (Section 2.1.1), the triangulation process occurs by:

1.1 applying a SVD transformation to remove the U(1)-symmetry degeneracy inherent in the VFZO coordinates (Appendix B.1.1)

1.2 linearly projecting VFZOs onto a hyperplane that is tangent to the vector between the origin and the mean of the input VFZOs to reduce computational burden of the triangulation

1.3 performing a second SVD transformation (Appendix B.1.3)

1.4 computing the triangulation according to the quickhull algorithm [54] using built-in methods

In the explanation of each of these steps that follows, we make reference to lower-dimensional visual analogues of the VFZO triangulation procedure, which are given in Figure B.1,
Figure B.2, and Figure B.3. We note that 3D Cartesian coordinates in Figure B.1 correspond to 8D Cartesian coordinates, whereas 3D Cartesian coordinates in Figure B.2 and Figure B.3 correspond to 7D Cartesian coordinates. This is intentional for two reasons:

- Figure B.1 illustrates that unsymmetrized 8D Cartesian GBOs are analogous to a point cloud on the 2-sphere (Figure B.1a) and that an 8D Cartesian VFZO set, which has already been symmetrized, is analogous to a geodesic arc on the 2-sphere (Figure B.1b). A VFZO set has a degenerate dimension that can then be removed by a rigid SVD transformation to 7D Cartesian coordinates (analogous to 2D Cartesian coordinates in Figure B.1c). This sequence would be more difficult to visualize if Figure B.1a was meant to represent a point cloud on the 3-sphere (4D Cartesian coordinates), etc.

- Figure B.2 illustrates a second transformation from normalized 7D Cartesian coordinates (Figure B.2a) to a hyperplane (Figure B.2b) which is then transformed into 6D Cartesian coordinates via a second SVD. In this case, key issues are retained that would otherwise be lost (Section 4.3.1) if an arc on a circle (1-sphere) to 1D Cartesian coordinates were used instead. Additionally, the use of actual triangles is a more familiar and compelling illustration of triangulation.

While lower dimensional analogues are useful for visualizing and understanding the process of triangulation, a written description is also given in the following sections. As appropriate, we refer back to the teaching figures described in this section.

B.1.1 Singular Value Decomposition Transformation from 8D Cartesian to 7D Cartesian

To reduce the computational complexity of triangulating a high-dimensional mesh [54], some simplifications are made. First, the degenerate octonion dimension obtained from analytically minimizing $U(1)$ symmetry [7] is removed via a rigid (i.e. distance- and angle-preserving) SVD transformation, analogous to a Cartesian rotation and translation (see 3D to 2D SVD transformation from Figure B.1b to Figure B.1c).

1Non-intersection issues due to high-aspect ratios and consideration of facets connected up to $\max\text{NNs}$ do not manifest in triangulations on the surface of a 1-sphere because one of the two facets (i.e. line segments) connected to the first NN mesh vertex relative to the prediction point is guaranteed to have an intersection.
B.1.2 Linearly Project onto Hyperplane

Next, the resulting 7D Cartesian representation of each VFZO is projected onto a hyperplane that is tangent to the centroid (i.e., mean) of the VFZO set\(^2\) (Figure B.2a). By performing this linear projection, one of the dimensions becomes degenerate.

\(^2\)This is not a rigid transformation; however, it approximates one with sufficient accuracy to produce a high-quality triangulation in a VFZ.
B.1.3 Singular Value Decomposition Transformation from 7D Cartesian to 6D Cartesian

This additional degeneracy is removed via a second SVD transformation, this time to 6D Cartesian coordinates (see 3D to 2D projection in Figure B.2a-b). Finally, the resulting points can be triangulated via the quickhull algorithm [54] (see VFZO repository function sphconvhulln.m and built-in MATLAB function delaunayn()), which relies on Euclidean distances. Because the simplicial mesh is defined by a list of edges between vertices for each simplicial facet, this list applies immediately to the VFZO set in its 7D Cartesian coordinates (i.e. no reverse transformation is necessary to use the mesh on the 6-sphere in 7D).

B.2 Intersections in a Voronoi Fundamental Zone Mesh

Once the triangulation has been determined, we need to find which facet each prediction point intersects (i.e. find the intersecting facet). There are two sub-steps:

2.1 applying the same rigid transformation to the prediction points as was applied to the input points (otherwise the prediction points won’t line up properly with the mesh) (Appendix B.2.1)

2.2 identifying facets nearby a prediction point and testing for intersection (Appendix B.2.2).

B.2.1 Apply Same Singular Value Decomposition to Input and Prediction Points

The positions of the prediction points need to be fixed relative to the mesh even after the rigid SVD transformation. This is accomplished by:

2.1a concatenating both input and prediction points

2.1b using the interp5DOF.m sub-routine proj_down.m (which depends on MATLAB’s built-in SVD implementation svd()) to perform the transformation

2.1c subsequently separating the transformed input and prediction points (reverse of concatenation step)

While the triangulation algorithm used in this work relies on Euclidean distances (the use of which is possible via the VFZO framework), other distance metrics that are non-Euclidean [53] could potentially be incorporated into the barycentric approach such as by doing an edge-length based simplex reconstruction [73, 74] using the VFZ triangulation edge lengths.
To map new points onto the mesh, the usv structure output from \texttt{proj\_down.m} needs to be stored and supplied in future calls to \texttt{proj\_down.m}. Likewise, usv need to be supplied to \texttt{proj\_up.m} to perform the reverse SVD transformation.

### B.2.2 Testing Nearby Facets for Intersections

Once the prediction points are lined up properly with the mesh, the facet containing the prediction point (i.e. intersecting facet) is found. We define the intersecting facet as the one for which a point's barycentric coordinates are positive within a given tolerance. Consequently, we determine facet affiliation by:

1. **2.2a** linearly projecting the prediction point onto the hyperplane defined by a mesh facet’s vertices (Figure B.3)

2. **2.2b** computing the point’s barycentric coordinates within the facet [75,76] (see VFZO repository function \texttt{projray2hypersphere.m})

3. **2.2c** testing that all coordinates are positive [60] within a tolerance\(^4\)

4. **2.2d** repeating steps 2.2a-2.2c until an intersection is found or a stop condition is reached (see \texttt{nnMax} below).

For further information on barycentric coordinates and its applications and generalizations, see [60–62, 76–89].

Due to the large number of facets per point of a high-dimensional triangulation (approximately 2000 facets per vertex for a 50000 point VFZ triangulation, or \(1 \times 10^8\) total facets), some simplifications are made in order to determine intersections of prediction points with the mesh. If every edge length of every facet were equal, only facets connected to the first NN would need to be considered to find a proper intersection. However, since the VFZOs are randomly sampled, edge lengths of facets are non-uniform, and non-unity aspect-ratio facets exist (Figure B.2, Figure 4.4). If the facets have high-aspect ratios, the intersecting facets of prediction points can be far from the

---

\(^4\)Two tolerances are used: one for the initial computation of barycentric coordinates by projecting onto the hypersphere to determine facet affiliation (\texttt{projtol=1e-4}) and a larger tolerance (\texttt{inttol=1e-2}) for computation of barycentric coordinates to determine interpolated values (Appendix B.3).
Figure B.3: A ray (red line) is linearly projected from the 2-sphere onto the hyperplane of a mesh facet (transparent black), shown as a red asterisk. The barycentric coordinates are computed as $\lambda_i \in [1,3] = \frac{1}{3}$. Because all barycentric coordinates are positive, it is determined that the projected point is an intersection with the mesh. Given vertex values of 8.183, 3.446, and 3.188 for vertices 1, 2, and 3, respectively, the interpolated value is calculated as 4.94 via Eq. (B.1).

NNs mesh points relative to the prediction points (see Figure 4.4 inset), especially near the perimeter of a hyperspherical surface mesh. Rather than loop through every facet to find an intersection ($\sim 1 \times 10^8$ facets in a 50000 VFZO mesh), the prediction point intersections are calculated by considering facets connected to up to some number of NN mesh vertices ($\text{nnMax}$) relative to each prediction point (in this work, $\text{nnMax}=10$). The NN mesh vertices relative to a prediction point are computed via the MATLAB built-in function $\text{dsearchn}$ as in the NN approach (Section 2.3.4). The facet IDs of facets connected to these NNs are computed by calling built-in MATLAB function $\text{find}()$, as in $\text{find}(K==\text{nn})$, where $K$ is the triangulation from VFZO repository function $\text{sphconvhulln.m}$ and $\text{nn}$ is the ID of one of the NN mesh vertices.

Some prediction points will have no intersecting facet found. From my numerical testing, we determine that this non-intersection phenomenon occurs in two situations:

- high-aspect ratio facets (described above)
- prediction points that are positioned just outside the bounds of the mesh but within the bounds of the VFZ, due to the fact that the mesh is a piecewise linear approximation of a surface with a curved perimeter and that randomly sampled points typically do not fall on the true perimeter

In the first case, barycentric interpolation within high-aspect ratio facets may actually lead to worse interpolation error than a NN interpolation strategy due to influence by GBs far from the prediction point. In the second case, there is no true intersection between the prediction point and the mesh. Both issues can be addressed with the same strategy: we apply a NN approach (Section 2.3.4) when
an intersecting facet is not found within nnMax NNs. In numerical tests, VFZ meshes composed of 388 and 50000 vertices produced non-intersection rates of (12.07 ± 1.02) % and (0.68 ± 0.11) %, respectively, over approximately 10 trials and using 10000 prediction points for each trial.

Testing intersections for nearby facets is handled in the VFZO repository function intersect_facet.m and depends on the barycentric coordinate computations in projray2hypersphere.m.

B.3 Interpolation via Barycentric Coordinates

Once a mesh triangulation has been determined (Appendix B.1), barycentric coordinates are recomputed for a prediction point within the input mesh (Appendix B.2) using a somewhat larger tolerance; the interpolated value is found by taking the dot product of the prediction point’s barycentric coordinates and the properties of the corresponding vertices of the intersecting facet via

\[ v_{m,q} = \sum_{i=1}^{N} \lambda_{m,i} v_{m,i} \]  

(B.1)

where \( \lambda_{m,i} \), \( v_{m,q} \), \( v_{m,i} \) and \( N \), are the barycentric coordinates of the m-th prediction point, interpolated property at the m-th prediction point, property of the i-th vertex of the intersecting facet for the m-th prediction point, and number of vertices in a given facet (\( N = 7 \) for facets of the simplicial mesh on the degeneracy-free 6-sphere), respectively. Interpolation of many prediction points simultaneously can be accomplished by a simple, vectorized approach via MATLAB built-in function dot() as used in VFZO repository function interp_bary_fast.m. This function assumes triangulation and weights have been precomputed. In other words, both input and prediction coordinates remain fixed, and only input property values change. If this is the case, barycentric interpolation of new points is incredibly fast. By contrast, if input coordinates change, the triangulation must be recomputed, and if prediction coordinates change, the intersecting facets must be recomputed. Both triangulation and finding intersecting facets are computationally demanding with respect to memory and runtime (Section 3.2).