



An Efficient Algorithm for Generating Diverse Microstructure Sets and Delineating Properties Closures

Johnson, O.K., Kurniawan, C.

Oliver Johnson
ojohnson@byu.edu

Original Publication Citation

An Efficient Algorithm for Generating Diverse Microstructure Sets and Delineating Properties Closures, Johnson, O.K., Kurniawan, C., *Acta Materialia*. vol. 147, April, pp. 313-321 (2018). doi: [10.1016/j.actamat.2018.01.004](https://doi.org/10.1016/j.actamat.2018.01.004)

An Efficient Algorithm for Generating Diverse Microstructure Sets and Delineating Properties Closures

Oliver K. Johnson^{a,b,*}, Christian Kurniawan^a

^aDepartment of Mechanical Engineering, Brigham Young University, Provo, UT 84602

^bDepartment of Materials Science and Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

Abstract

Properties closures delineate the theoretical objective space for materials design problems, allowing designers to make informed trade-offs between competing constraints and target properties. We present a new algorithm called hierarchical simplex sampling (HSS) that approximates properties closures more efficiently and faithfully than traditional optimization based approaches. By construction, HSS generates samples of microstructure statistics that span the corresponding microstructure hull. As a result, we also find that HSS can be coupled with synthetic polycrystal generation software to generate diverse sets of microstructures for subsequent mesoscale simulations. By more broadly sampling the space of possible microstructures, it is anticipated that such diverse microstructure sets will expand our understanding of the influence of microstructure on macroscale effective properties and inform the construction of higher-fidelity mesoscale structure-property models.

Keywords: Microstructure, Microstructure Hull, Properties Closure, Microstructure Design

1. Properties Closures

The last decade and a half has seen the development of a suite of theoretical tools to design and optimize the microstructure of materials for specific engineering applications. These developments were formalized in a design methodology called *microstructure sensitive design for performance optimization* (MSDPO) [1]. One of the strengths of MSDPO is the ability to consider the complete envelope of all physically possible properties combinations for a given design problem. This space is referred to as a *properties closure*, \mathcal{P} , and it permits materials designers to visualize the full design space, observe complex interactions between properties, and make design trade-offs as necessary (see Fig. 1).

A properties closure can be thought of loosely as a theoretical analog of the familiar Ashby plots [2]. However, the primary purposes of these two design tools are fundamentally different. Ashby plots, being compiled from experimental data, summarize

the possible properties combinations of materials that already exist, and therefore serve as a valuable tool for *materials selection*. In contrast, properties closures, being derived from constitutive models, predict the properties combinations of all possible microstructural configurations (including composition and other state variables) for a particular material system whether or not they have yet been synthesized, and therefore serve as a valuable tool for *materials development and design*.

For example, Fig. 1, illustrates a properties closure for the combination of yield strength (σ_{y1}) and elastic compliance (S_{1111}) for polycrystalline α -Ti [10]. This properties closure was used to design an optimal crystallographic texture for a cantilever beam application, in which the design objective was maximization of deflection without yield. Rather than adapting the geometry of the design to the limitations of existing microstructures, this procedure enables adaptation of the microstructure to suit the engineering design (note the optimal texture shown in the inset).

Existing algorithms for delineating properties closures can be computationally expensive (particularly as the number of properties of interest increase), can get stuck in local minima, and are sen-

*Corresponding Author: Tel. +1-801-422-0972

Email address: o.johnson@byu.edu (Oliver K. Johnson)

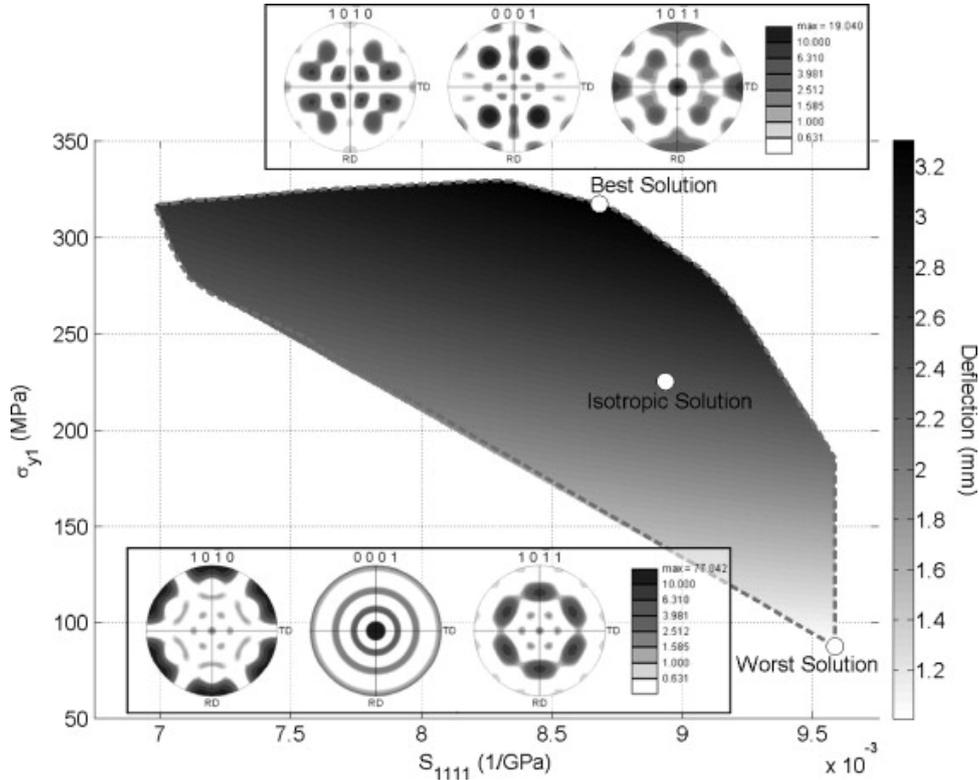


Figure 1: The relevant property closure for a cantilever compliant beam made of high-purity polycrystalline α -Ti with hexagonal-orthorhombic textures. The textures predicted to provide the best and the worst performances for this case study are also shown. Reprinted from [10], with permission from Elsevier.

sitive to a variety of algorithmic parameters. In the present work we develop a new, more efficient algorithm for delineating properties closures. We also find that this algorithm has utility for generating diverse sets of synthetic microstructures that span a properties closure, facilitating the investigation of complex structure-property correlations across a wide spectrum of possible microstructures. This approach may be useful in conjunction with recently developed synthetic microstructure reconstruction software packages like DREAM.3D [14] and Neper [23].

2. Previous Algorithms

All methods for generating properties closures involve exercising structure-property models over a *microstructure hull*, which we denote by the symbol \mathcal{M}_H . A microstructure hull is the space of all possible microstructures, as described by some parameterization relevant to the physical model under consideration [1]. For instance, in a two phase composite model for elastic modulus, a microstructure could be parameterized by the volume fraction of one of the phases, *e.g.* f_α , and the microstructure hull would simply be $\mathcal{M}_H = \{f_\alpha \mid f_\alpha \in [0, 1]\}$. For more complex constitutive models that depend on higher-order microstructural features like the dis-

tribution of crystal orientations, or grain boundary (GB) characters, the number of parameters can be large or even infinite and \mathcal{M}_H is significantly more complex [1, 17]. For example in the case of orientation distribution functions (ODFs) expressed as a generalized Fourier series, there are technically an infinite number of coefficients that constitute the set of state variables (or microstructural parameters). In practice the series is truncated and a finite, but potentially large, number of coefficients are used.¹

Once the relevant \mathcal{M}_H is defined, algorithms for delineating the corresponding properties closure involve various methods of exploring \mathcal{M}_H in search of microstructures with extremal properties.

Analytical methods include bounding theories such as the Voigt [28] and Reuss [24] bounds, or the Hashin-Shtrikman bounds [16]. Points exterior to the bounds for \mathcal{P} predicted by these theories are guaranteed to be non-physical, however these bounds are not tight: there are also points interior to the predicted bounds that are non-physical. This complicates optimization as mathematically feasible solutions may be obtained that do not correspond to physically realizable microstructures. Furthermore, they provide bounds only for a sin-

¹In such cases the methods presented in this work are still compatible.

gle property rather than simultaneous combinations of properties (Cartesian products of such bounds are not appropriate because materials properties are generally not independent).

Of the computational methods, the simplest approach to construct \mathcal{P} involves uniformly sampling \mathcal{M}_H [1]. While rapid, it provides a very poor approximation of \mathcal{P} that explores only a very small, centrally located portion of the full closure. More accurate estimates of \mathcal{P} have been obtained by Pareto Front techniques using various algorithms including Generalized Weighted Sum (GWS), Adaptive Weighted Sum (AWS), and Adaptive Normal Boundary Intersection (ANBI) [1] (see [10, 22] for a related approach). For some problems, GWS can employ simple quadratic programming. However, it obtains boundary points that are tightly clustered around regions of high curvature. More importantly, GWS can only identify points on convex surfaces and \mathcal{P} is generally not convex. The other algorithms can find boundary points on concave and convex portions of \mathcal{P} , but at the cost of introducing nonlinear constraints requiring more computationally expensive optimization routines like sequential quadratic programming (SQP). A hybrid approach can also be employed with GWS being used to obtain an initial set of boundary points and, e.g., ANBI being used to delineate the concave portion of \mathcal{P} [1].

As an alternative to gradient based methods, genetic algorithms (GA) have also been used [1, 29]. These are generally less accurate than the deterministic approaches, but tend to spread out the boundary points more uniformly, even along concave regions.

With the exception of uniform sampling, all of these methods are time intensive and involve a large number of search directions, particularly as the number of properties of interest grows. In this paper we propose a new algorithm for approximating a properties closure that does not require the use of optimization techniques. In addition to aiding in the delineation of properties closures, this method is extremely useful as a means to generate synthetic microstructures for subsequent simulations. Our algorithm is a refinement of the uniform sampling approach, and, as such, retains the benefits of simplicity and efficiency, but is significantly more robust.

3. A New Algorithm: Hierarchical Simplex Sampling

To construct \mathcal{P} , we take a stochastic approach and sample many microstructures from \mathcal{M}_H simultaneously, compute the relevant material properties of each, and define \mathcal{P} as the region in properties space that bounds all of the resulting points. We wish to sample microstructures in such a way that the resulting properties closure is maximal. In other words, we want to choose microstructures that will explore as much as possible of the true properties closure.

The methods explained here are very general and apply to arbitrary microstructure hulls and any properties closure provided that appropriate constitutive equations are available. Without loss of generality, we will make the ensuing discussion concrete by considering a specific design problem in which the microstructure is described by its orientation distribution function (ODF) and the texture sensitive properties of interest are a single component of the elastic compliance tensor, \overline{S}_{1111} , and the effective diffusivity of the GB network, \overline{D} . The overline indicates that these are the homogenized effective properties of a polycrystal.

An arbitrary ODF can be expressed as a series expansion in the basis of Dirac delta functions [1, 10, 18] according to:

$$f(q) \approx \sum_{j=1}^J p'_j \delta(q, {}^j q) \quad (1)$$

where $q \in \text{SO}(3)$ is an orientation and ${}^j q$ is one of a set of *fundamental orientations* that form a discretization of $\text{SO}(3)$. The approximation in Eq. 1 becomes exact as $J \rightarrow \infty$. The texture coefficient, p'_j , can be interpreted as representing the probability of observing an orientation corresponding to a bin centered at ${}^j q$. The set of coefficients for a given ODF can be written as a vector, $\mathbf{p}' = (p'_1, p'_2, \dots, p'_J)$, representing a point in a J -dimensional vector space. Different textures have different coefficients and consequently are represented as different points in this space. For a single crystal with orientation ${}^i q$ the coefficients of the corresponding ODF are just $p'_j = \delta_{ij}$. The set of coefficient vectors for all of the fundamental orientations is called the *texture set* and is defined by

$$\mathcal{M}_S^{(1)} = \left\{ {}^j \mathbf{p}' \mid {}^j \mathbf{p}' = ({}^j p'_1, {}^j p'_2, \dots, {}^j p'_J), \right. \\ \left. {}^j p'_i = \delta_{ij}, j \in [1, J] \right\} \quad (2)$$

$M_S^{(1)}$ forms a microstructural basis and any ODF can be written as a convex combination of the basis vectors of $M_S^{(1)}$. This allows us to define the microstructure hull for crystallographic texture, which we call the *texture hull* and denote $M_H^{(1)}$, as the convex hull of the texture set:

$$M_H^{(1)} = \left\{ \mathbf{p}' \mid \mathbf{p}' = (p'_1, p'_2, \dots, p'_J), \right. \\ \left. 0 \leq p'_j, \sum_{j=1}^J p'_j = 1 \right\} \quad (3)$$

As can be discerned from Eq. 3, in the Dirac basis, the texture hull is a standard $(J - 1)$ -simplex, whose vertices are the elements of $M_S^{(1)}$. If we had expressed the ODF in a generalized Fourier basis—such as the generalized spherical harmonics or hyperspherical harmonics—the construction of the texture hull would have required the explicit computation of a high-dimensional convex hull, which is generally not practical. The key to our new algorithm lies in exploiting the simple geometric structure of a microstructure hull expressed in the Dirac basis. This geometric structure obviates the need to explicitly compute a convex hull, and also makes the procedure for sampling far simpler.

3.1. Uniform Simplex Sampling

As mentioned in Section 2, uniformly sampling a microstructure hull leads to a poor approximation of the properties closure, \mathcal{P} . Notwithstanding, since our method is related to this process, it will be important to understand how this might be performed in the present context. Sampling uniformly from a standard $(J - 1)$ -simplex can be accomplished in a number of ways including rejection sampling [5, 15], gaps between sorted samples from $U(0, 1)$ [7, 25], and various Markov Chain Monte Carlo (MCMC) methods such as the class of hit-and-run algorithms [27]. The particular choice of a uniform simplex sampling approach is not of central importance to our proposed algorithm, and any technique for generating uniform samples from the unit simplex can be substituted. In our tests, we have employed the following procedure, which is adapted from a combination of the inverse-transform sampling method and Theorems 2.1-2.2 of [7] (see also [25]):

1. Generate J uniformly distributed random variables X_1, X_2, \dots, X_J

2. Transform them to exponentially distributed random variables *via* $Y_i = -\ln X_i$
3. Perform the transformation $P'_i = Y_i / \sum_{j=1}^J Y_j$

The vector $(P'_1, P'_2, \dots, P'_J)$ will then be uniformly distributed over the standard $(J - 1)$ -simplex. We refer to this as the uniform simplex sampling (USS) algorithm. While this sampling method is computationally inexpensive, it does not result in points that adequately explore the properties closure. The reason for this is that the boundary of the simplex (faces, edges, etc.) has measure 0 in J dimensions, and, consequently, is never sampled. Because all of the points come from the interior of the simplex they will be a mixture of all J elements of $M_S^{(1)}$, and thus do not differ sufficiently from the uniform ODF, which is located at the barycenter of the simplex. However, this observation suggests an alternative approach.

3.2. Hierarchical Simplex Sampling

Recognizing that the faces, edges, etc. of a $(J - 1)$ -simplex are just lower-dimensional simplices themselves, the simplex sampling approach may be performed in a hierarchical fashion, starting with the vertices (0-faces), then the edges (1-faces), faces (2-faces), and continuing with the hyper-faces of increasing dimension. Let N_S be the desired number of samples. Our *hierarchical simplex sampling* (HSS) algorithm is summarized in the following steps:

1. Take all J of the vertices (0-faces)
2. Take $\max(1, \lfloor \beta \times N_S \rfloor)$ points sampled uniformly from the $(J - 1)$ -face (the simplex interior)
3. Beginning with $j = 2$ and proceeding to increasing j , take $\lfloor N_r (J - j) / \sum_{j=2}^{J-1} j \rfloor$ points sampled uniformly from the standard $(j - 1)$ -simplex and assign them uniformly across all of the $(j - 1)$ -faces. This can be accomplished simply by applying a different random permutation to the coordinates of each sample.

In this procedure, $\lfloor x \rfloor$ is the rounding operation, and N_r is the remaining number of samples, yet to be taken. In Step 2, the term $\max(1, \lfloor \beta \times N_S \rfloor)$ ensures that at least 1 point is taken from the interior. We employ $\beta = 10^{-4}$; however, we have found that as long as N_S is reasonably large the value of β has no impact on the algorithm because interior points are never found on the boundary of \mathcal{P} , which is instead composed of more extreme microstructures

that reside on lower-dimensional hyperfaces of the simplex. In fact, our experience suggests that Step 2 can be entirely omitted, though we report it for the sake of completeness. In Step 3 we use the heuristic of taking a number of samples equal to $\lceil N_r(J-j)/\sum_{j=2}^{J-1} j \rceil$ for the $(j-1)$ -simplices. This heuristic was optimized empirically. Our HSS algorithm retains much of the computational efficiency of the uniform simplex sampling approach, but probes far more of the properties closure, as will be demonstrated.

4. Results & Discussion

For the present design problem we use this HSS algorithm to sample textures from $M_S^{(1)}$. We will denote this set of sampled textures

$$\mathcal{S} = \left\{ \mathbf{p}'_s \mid \mathbf{p}'_s = \{p'_{j_s}\}_s, j \in [1, J], s \in [1, N_S] \right\} \quad (4)$$

where \mathbf{p}'_s is the s -th sample, and can be interpreted as a vector of coefficients $\{p'_{j_s}\}_s = (p'_{1_s}, p'_{2_s}, \dots, p'_{J_s})^T$. In this way \mathcal{S} , which is a set of sets, can be represented as a matrix where \mathcal{S}_{js} is the j -th coefficient of the s -th sample.

To demonstrate the effectiveness of our new HSS algorithm, we map the microstructures sampled from $M_H^{(1)}$ (i.e. \mathcal{S}) to the properties space. This requires the use of relevant constitutive equations expressed in spectral form. For the present work we employ polycrystal homogenization models for the effective elastic compliance (\overline{S}_{1111}) and effective grain boundary network diffusivity (\overline{D}) of polycrystalline Al that are adapted from [18] to accommodate fully three-dimensional ODFs. The derivation and application of these models are discussed in depth in [19]. We mention them only briefly here.

The model for \overline{S}_{1111} is derived from the Voigt isostrain model [4, 28], which provides a simple upper-bound and can be expressed as:

$$\overline{S}_{1111} = \sum_{n,l,m} c_{l,m}^n [s_{1111}]_{l,m}^n \quad (5)$$

with

$$[s_{1111}]_{l,m}^n = \int_{S^3} Z_{l,m}^{n*}(q) S_{1111}(q) dq \quad (6)$$

In Eq. 5, $c_{l,m}^n$ are the coefficients of an ODF expressed in the basis of hyperspherical harmonics, $Z_{l,m}^n(q)$, and $*$ is the complex conjugate operation.

The function $S_{1111}(q)$ provides the single-crystal orientation dependence of S_{1111} . The transformation required to relate the hyperspherical harmonic coefficients to the Dirac coefficients of Eq. 1 is provided in [19]. Formally, the hyperspherical harmonic representation consists of an infinite series, but in all practical applications truncation is applied. In this work we include all terms through $n = 20$.

For \overline{D} we employ the generalized effective medium (GEM) model [6]:

$$p_1 \frac{D_1^{1/s} - (2\overline{D})^{1/s}}{D_1^{1/s} + (p_{c,2}^{-1} - 1)(2\overline{D})^{1/s}} + p_2 \frac{D_2^{1/t} - (2\overline{D})^{1/t}}{D_2^{1/t} + (p_{c,2}^{-1} - 1)(2\overline{D})^{1/t}} = 0 \quad (7)$$

where D_1 and D_2 are the diffusivities of the low- and high-angle grain boundaries, p_1 and p_2 are their number fractions, $p_{c,2}$ is the percolation threshold for the high-angle grain boundaries, and s and t are the critical exponents that are assumed to be fixed for a given universality class. In our implementation, p_1 , p_2 , and $p_{c,2}$ are predicted from the triple junction fractions [11–13, 20, 26], which, in turn, are predicted from the ODF coefficients, p'_j , as explained in [19].

4.1. A Note on Discretization

These structure-property models are built from the same spectral approach embodied in Eq. 1, and consequently require a discretization of the relevant space of state variables. For texture sensitive properties like \overline{S}_{1111} that space is $SO(3)$ and the discretization produces the *fundamental orientations* already discussed. For such models a uniform or equispaced discretization is a logical choice. However, grain boundary network sensitive properties like \overline{D} depend on grain boundary *misorientations* and their interactions at triple junctions, and consequently on the spacing between fundamental orientations in $SO(3)$. This results in special considerations regarding the choice of discretization.

In particular, the model for \overline{D} depends on the triple junction fractions, and the types of triple junctions that can be represented are limited by the unique triplets of fundamental orientations (modulo crystal and triple junction symmetries). The triple junction fractions (J_0 , J_1 , J_2 , and J_3) characterize the fraction of triple junctions coordinated

J_0	J_1	J_2	J_3
0.6480	0.3329	0.0139	0.0052

Table 1: Fraction of each triple junction type appearing in the set of fundamental triple junctions resulting from the multi-level pseudo-grid discretization of orientation space.

by 0, 1, 2, or 3 low-angle grain boundaries, respectively. Importantly, a uniform grid of fundamental orientations does not produce a uniform distribution of triple junction types. The mapping between the spaces is such that certain types of triple junctions, J_2 in particular, are relatively rare, but are experimentally observed [12], and strongly influence grain boundary network connectivity as the *only* types of triple junctions at which non-closed connected paths of high-angle (typically high-diffusivity) grain boundaries can terminate. Thus, the absence of J_2 type triple junctions would restrict the grain boundary network configurations that could be considered. The resolution of a regular discretization of $SO(3)$ necessary to produce triple junctions of type 2 is prohibitively high, therefore, we employ an alternative discretization of $SO(3)$, as described below, which allows for representation of all 4 triple junction types with fewer discretization points².

First we define a coarse grid of 8 orientations in the fundamental zone ($\phi_1 \in \{45^\circ, 135^\circ, 225^\circ, 315^\circ\} \times \Phi \in \{20^\circ\} \times \phi_2 \in \{0^\circ, 45^\circ\}$). Around each of these orientations we subsequently define a local octahedral subgrid of orientations generated by rotations of of 10° about the positive and negative $\langle 100 \rangle$, $\langle 010 \rangle$, and $\langle 001 \rangle$ directions. The resulting set of fundamental orientations are shown in Fig. 2. Table 1 shows that, while the distribution of triple junction types among the fundamental triple junctions is still not equal, all types are represented.

4.2. Algorithm Performance Comparison

The model for \bar{D} assumes spatially uncorrelated grain orientations, so that the statistics of the GB network structure are completely specified by the ODF. However, \bar{D} has a rather complicated non-linear dependence on the texture coefficients, which results in a non-convex properties closure that is challenging to delineate and thus presents an opportunity to compare the performance of our

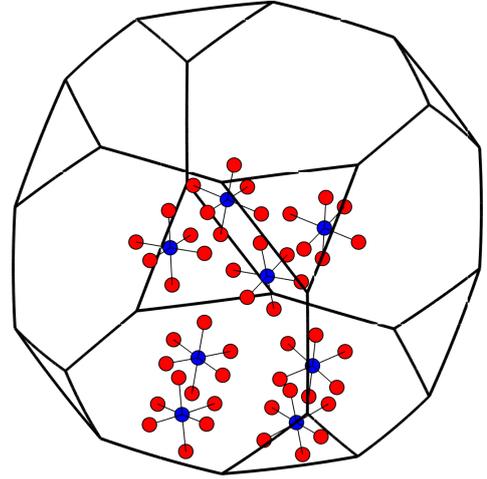


Figure 2: Fundamental orientations plotted in the cubic fundamental zone (FZ) using the quaternion parameterization and an isovolumetric projection [21] to three-dimensions. The blue markers correspond to the original coarse grid of orientations, and the red markers correspond to the respective local octahedral subgrids of orientations. The markers on the bottom of the figure extend below the boundary of the FZ and would normally be replaced by their periodic images inside of the top of the FZ, but they are presented as shown to highlight the structure of the grid and subgrids. The lines between points are for clarity of visualization only.

HSS algorithm with the prior work described in Section 2.

To compare the performance of each algorithm, we evaluated the number of points (solutions) that each algorithm obtained when run for approximately the same duration (approximately 370 sec.). For the GWS, AWS, and ANBI algorithms, all solutions obtained were boundary points of the approximation of \mathcal{P} that the method found (i.e. the solutions were either on the true surface of \mathcal{P} or they converged to local minima as they approached the surface of \mathcal{P}). For the GA, USS, and HSS algorithms, additional points were obtained on the interior of \mathcal{P} , and for these methods we identified the boundary of the resulting point clouds (and the points residing on them) using the built-in boundary() function in MATLAB, with a “shrink factor” equal to 1. This approach identifies the point cloud boundary by computing the respective α -shape [8, 9] with the smallest radius that produces a single region.

The computation times, and number of total points and boundary points obtained per unit time are provided in Table 2. For a fixed time, our HSS algorithm obtained between 8 and 44 times the number of boundary points as the traditional methods (GWS, AWS, ANBI, GA), and because it probed a much greater portion of \mathcal{P} than the USS algorithm (even though it found roughly the same

²See [19] for more discussion related to this phenomenon.

Table 2: Performance comparison between the HSS algorithm from the present work and traditional property closure delineation algorithms.

Method	Time (s)	Points/Time (s ⁻¹)	Boundary Points/Time (s ⁻¹)
GWS	359.10	0.10	0.10
AWS	377.31	0.02	0.02
ANBI	377.82	0.03	0.03
GA	372.10	0.54	0.11
USS	367.50	90.06	0.40
HSS	373.85	88.53	0.87

total number of points) the HSS algorithm found more than twice as many boundary points as the USS algorithm.

In addition to the number of points obtained, the fidelity of the resulting approximation of \mathcal{P} is of great interest. Since an exact analytical expression for \mathcal{P} is not available, the results of each algorithm were compared to a high-resolution approximation generated by the following procedure. A total of 10^8 ODFs were sampled via HSS in groups of 10^5 (i.e. the sampling was parallelized for computational efficiency). Normalized properties space coordinates, $\mathbf{r} = [x(\mathbf{p}'_s), y(\mathbf{p}'_s)]^T$, were then computed for all points according to

$$x(\mathbf{p}'_s) = \frac{2\overline{S_{1111}}(\mathbf{p}'_s) - (\overline{S_{1111}^{\max}} + \overline{S_{1111}^{\min}})}{\overline{S_{1111}^{\max}} - \overline{S_{1111}^{\min}}} \quad (8a)$$

$$y(\mathbf{p}'_s) = \frac{2 \ln \overline{D}(\mathbf{p}'_s) - (\ln \overline{D}^{\max} + \ln \overline{D}^{\min})}{\ln \overline{D}^{\max} - \ln \overline{D}^{\min}} \quad (8b)$$

where \overline{X}^{\max} and \overline{X}^{\min} are the theoretical maximum and minimum for effective property \overline{X} . For each group of 10^5 ODFs, α -shapes with radii of 0.05, 0.1, and ∞ (corresponding to the convex hull) were then computed and the union of the point sets falling on each of these three α -shapes were stored. Because they do not enter into the calculation required to delineate \mathcal{P} , the interior points were not stored. This downselection process preserved a total of 887,814 of the original 10^8 ODFs. Let us denote this set of sampled ODFs by \mathcal{S} and its image in the normalized properties space by \mathcal{R} . The high-resolution approximation of \mathcal{P} was then computed by finding the most distant elements of \mathcal{R} in every direction by a process adapted from the GWS algorithm. For each direction defined by unit vectors $\hat{\mathbf{n}}_i = [\cos \theta_i, \sin \theta_i]^T$, with 256 evenly spaced $\theta_i \in [0, 2\pi]$, the point $\mathbf{r}_s \in \mathcal{R}$ falling approximately on the boundary of \mathcal{P} was chosen to be the one that maximized the projection $\mathbf{r}_s^T \hat{\mathbf{n}}_i$ subject to the

collinearity constraint

$$1 - \frac{\mathbf{r}_s^T \hat{\mathbf{n}}_i}{\|\mathbf{r}_s\|} < \tau \quad (9)$$

where a collinearity tolerance of $\tau = 10^{-4}$ was used, resulting in an angular tolerance of 0.81° . The boundary of the high-resolution approximation to \mathcal{P} resulting from this procedure is shown in Fig. 3. The density of points $\mathbf{r}_s \in \mathcal{R}$ located on the left and right sides of \mathcal{R} is lower than in other regions and so the approximation was corrected by manually excluding a total of 16 points (shown in red) that appear to be interior to the rest of the closure boundary. The fidelity of the approximations to \mathcal{P} obtained via each of the algorithms was evaluated by comparison with this high-resolution approximation to \mathcal{P} .

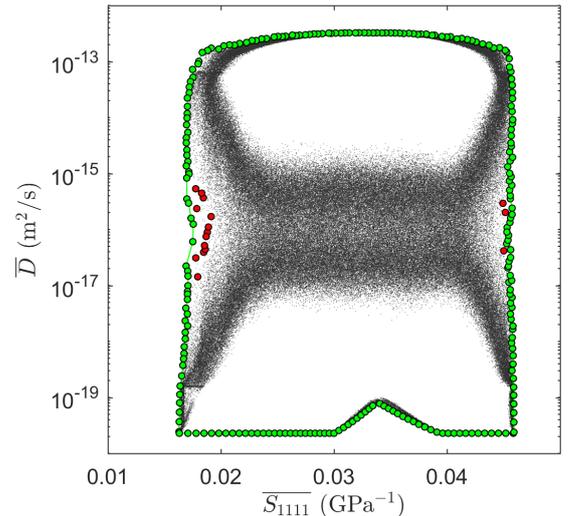


Figure 3: Calculation of the high-resolution approximation to \mathcal{P} . Black markers are the 887,814 points remaining after the downselection process. Colored markers are the subset identified as boundary points. Red markers were manually excluded. Green markers indicate points used to define the high-resolution approximation to \mathcal{P} to which the results of the various algorithms are compared in Fig. 4.

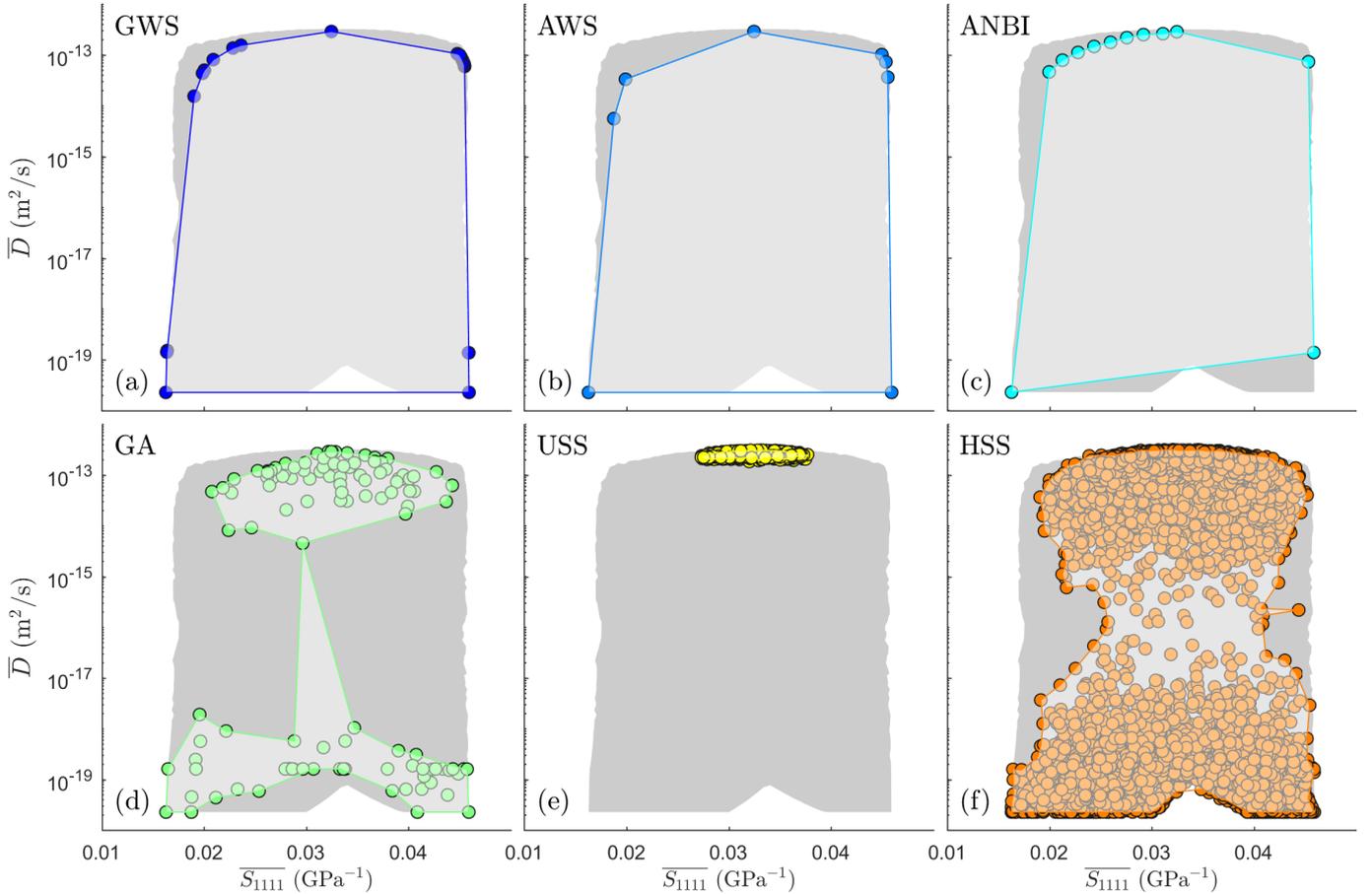


Figure 4: Comparison of points and closures obtained using traditional property closure delineation algorithms and the HSS algorithm from the present work. The dark gray region in each figure is the high-resolution closure from Fig. 3.

As illustrated in Fig. 4, the GWS, AWS, and ANBI methods were unable to find points on the boundaries of \mathcal{P} where those boundaries were nearly flat. In many cases multiple search directions converged to nearly identical solutions. Although AWS and ANBI are theoretically capable of finding points on concave regions, we observed that for this particular problem—for which the mapping between design space and objective space are quite complex—they were unable to do so. The GA respected the concavity of \mathcal{P} but produced very sparse coverage of the left and right sides of \mathcal{P} , resulting in a poor approximation. All of these optimization based algorithms can easily converge to local minima, and are sensitive to the starting location and various other algorithm-specific tuning parameters.

Although reproducible and absent of any adjustable parameters, USS produced points sampled entirely from the interior of $M_H^{(1)}$ which are tightly clustered in the upper portion of \mathcal{P} and result in an extremely uninformative and inaccurate approximation of \mathcal{P} . In contrast, for the same computational time, the solutions found by HSS fill nearly the entirety of \mathcal{P} and strictly respect the concave

region of \mathcal{P} , resulting in the most informative and tight fitting approximation of \mathcal{P} . Although stochastic, HSS produces reliably high-quality results. The only algorithmic parameters are how many samples to take at each level of the hierarchy, but these are fixed if the heuristic described earlier is employed.

We note that with the approximately 370 sec. run time, resulting in 33,096 points, the results of the HSS approach are not perfect, and the locations of remaining error are concentrated in the left and right portions of \mathcal{P} as shown in Fig. 4. As with most of the points on the boundary of \mathcal{P} , these regions correspond to low-dimensional (roughly three- to six-dimensional) hyperfaces of the microstructure hull simplex (see also Fig. 6). There are no immediately obvious features that distinguish points in these regions of \mathcal{P} from other points on its boundary (e.g. they do not reside on particular hyperfaces, but rather are distributed across many different hyperfaces that do not generally even share a vertex). However, as shown in Fig. 5, increasing the number of samples reduces this deviation to arbitrary accuracy. We note also that the computational time to generate more sam-

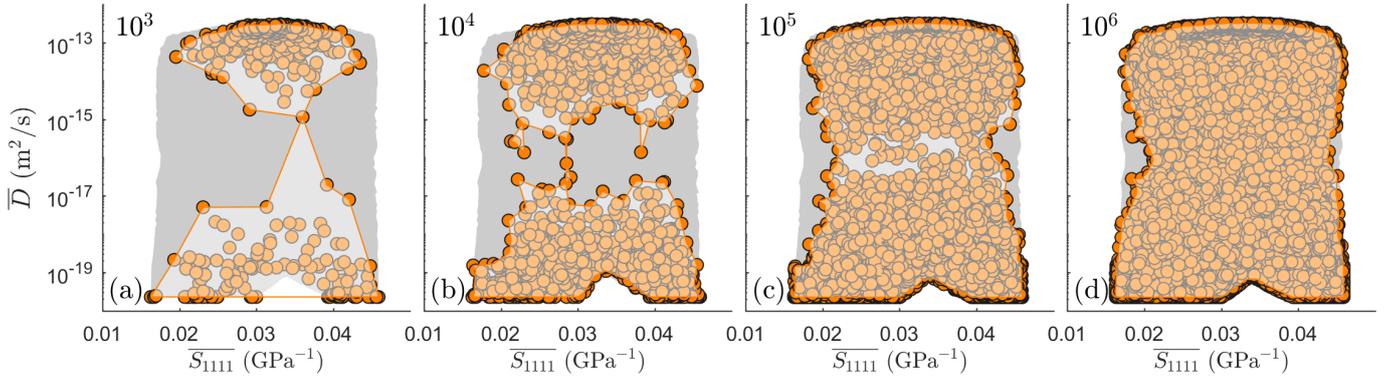


Figure 5: Illustration of the accuracy of the approximation of \mathcal{P} obtained via HSS with increasing number of samples, N_S , as indicated in the upper-left corner of each plot.

ples scales linearly with N_S .

In addition to the isochronal comparison already made, we can also consider an isoperformance comparison. In this context the question becomes “How much time is required to achieve a similar performance?”. To answer this question we must first note that the GWS, AWS, and ANBI methods were unable to respect the concave regions of the closure and for many search directions the algorithms converge to the same point. Consequently, considering more directions and applying more resources/time will not result in a closure approximation that is dramatically improved over what is already shown in Fig. 4. To verify this, we doubled the resolution for the GWS, AWS, and ANBI methods and the only changes that were observed were a few more points found on the top portion of the closures (the left, right, and bottom were unchanged), for the GWS and AWS algorithms, yet the required computation time nearly doubled. The ANBI algorithm found fewer points near the top (reflective of the parameter sensitivity of the algorithm). In contrast, the USS and GA algorithms should converge to the true closure with increasing number of samples, so for these cases a meaningful comparison might be made.

Figure 6 shows 10^6 microstructures sampled via the HSS algorithm plotted in the properties space, with each point colored by the number of nonzero elements in its corresponding Dirac coefficient vector, \mathbf{p}' . This provides an indication of what portions of the microstructure hull map to various regions of the properties closure. If all of the coordinates of \mathbf{p}' are nonzero then a point is interior to the microstructure hull simplex. Microstructures with fewer nonzero elements in \mathbf{p}' inhabit lower-dimensional hyperfaces of the microstructure hull. It is clear from Fig. 6 that the boundary of the properties closure is formed by microstructures in-

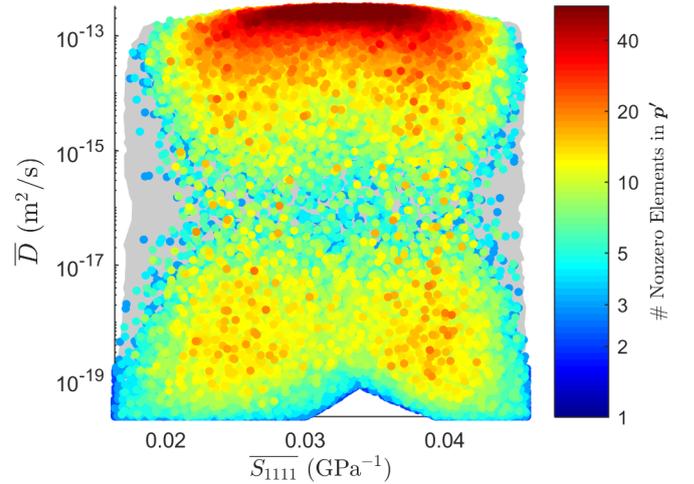


Figure 6: A total of 10^6 microstructures sampled via HSS with markers colored by the number of nonzero elements of the respective Dirac ODF coefficients, \mathbf{p}' . Microstructures with fewer nonzero elements inhabit lower dimensional hyperfaces of the microstructure hull simplex. Microstructures with many nonzero elements are heavily concentrated in the upper region of \mathcal{P} . Most (97%) of the extremal microstructures (which constitute the boundary of the point cloud) have between 2-3 nonzero elements, although finite marker size leads to marker overlap in the upper region making this difficult to see visually.

habiting very low-dimensional hyperfaces (97% of those on the point cloud boundary have between 2 and 3 nonzero elements). Because the USS algorithm samples exclusively from the interior of the microstructure hull the probability of obtaining a point on its surface is exactly zero, so that it appears impossible to find microstructures on the boundary of \mathcal{P} via the USS algorithm. However, we might consider how close of an approximation might be achieved. Using the USS algorithm, if the probability of having a single coordinate within ϵ of zero is $Q \in (0, 1)$ then the probability of sampling a point having $J - j$ of its coordinates within ϵ of zero is Q^{J-j} , which for any $j < J$ goes to zero with increasing dimension. Thus, good approximations to

\mathcal{P} may only be expected for very low-dimensional problems. For the present problem with a modest value of $J = 56$, the probability of obtaining a microstructure with even half of its coordinates close to zero via the USS algorithm would be on the order of 10^{-28} .

For the GA an isoperformance comparison is more practical, and Fig. 7 shows a comparison of the computation times required to generate microstructural samples with similar distributions over \mathcal{P} using the GA and the HSS algorithms, respectively, for two different levels of accuracy. While the GA does show the ability to achieve similar levels of fidelity, it is far less efficient at doing so, with the computation time being more than an order of magnitude greater than the HSS algorithm.

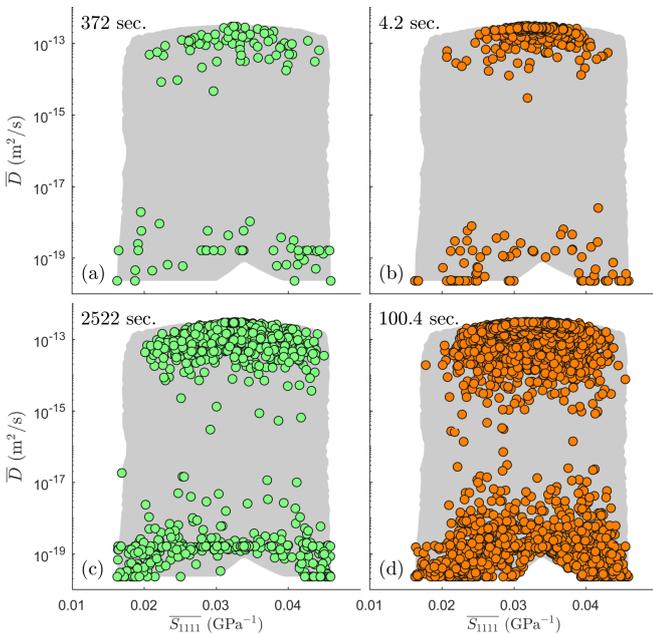


Figure 7: Comparison of computation times to generate microstructures with similar distributions over \mathcal{P} using the GA algorithm (left) and HSS algorithm (right). The top and bottom rows show the comparison for two different levels of accuracy/effort.

5. Generating Diverse Microstructural Sets for Mesoscale Simulations

In addition to the delineation of the boundary of \mathcal{P} , which provides an atlas of the space of all theoretically feasible property combinations, we have found the HSS algorithm useful in generating diverse sets of microstructures. It is becoming increasingly common to perform mesoscale simulations on synthetic polycrystals to investigate the influence of microstructure on, e.g., mechanical properties, phase transitions, or diffusion. Tools

such as Neper [23] and DREAM.3D [14] facilitate the generation of volumetric tessellations and meshes representing two- and three-dimensional polycrystals that can subsequently be used for such mesoscale simulations (e.g. via the finite element method (FEM)). While simulation of an individual microstructure can be instructive, it is generally desirable to perform simulations on a large set of microstructures so that trends can be observed and more general conclusions can be drawn regarding the underlying physical phenomena that are not masked by the idiosyncrasies of a single microstructural instantiation. The HSS algorithm can be used to generate such microstructural sets that span the microstructure hull of interest and consequently capture the full range of microstructural diversity.

Because the focus of the present work has been on the texture hull, $M_H^{(1)}$, we illustrate the microstructural diversity of a set of ODFs sampled from $M_H^{(1)}$ via the HSS algorithm. After obtaining the Dirac ODF coefficients, p'_s , for each sample via HSS, continuous ODFs were generated according to the kernel density estimator:

$$\hat{f}(q) = \sum_{j=1}^J p'_j K(q, {}^j q) \quad (10)$$

where $K(\cdot)$ was chosen to be the de la Valle-Poussin kernel with a 15° half-width. This process was carried out using the MTEX toolbox [3] for MATLAB. The resulting smooth and continuous ODFs are presented in (001) pole figures colored by multiples of the random distribution (MRD) in the top left of each of the subfigures in Fig. 8. The fundamental orientations for this application were defined by a regular grid in Euler angle space with a resolution of 20° , resulting in a total of 150 fundamental orientations.

Having sampled ODFs from $M_H^{(1)}$, it is possible to supply these to synthetic polycrystal generation software packages to specify a microstructure in a format suitable for mesoscale simulations (e.g. FEM). Figure 8 shows synthetic polycrystals generated and meshed using Neper [23], with respective grain orientations sampled from the ODFs we obtained via HSS and assigned randomly. The sampled orientations for each microstructure are presented in (001) pole figures in the top right of each of the subfigures in Fig. 8 together with the corresponding meshed polycrystalline microstructures. We note that this method of assigning grain orientations satisfies the target ODF only on a number

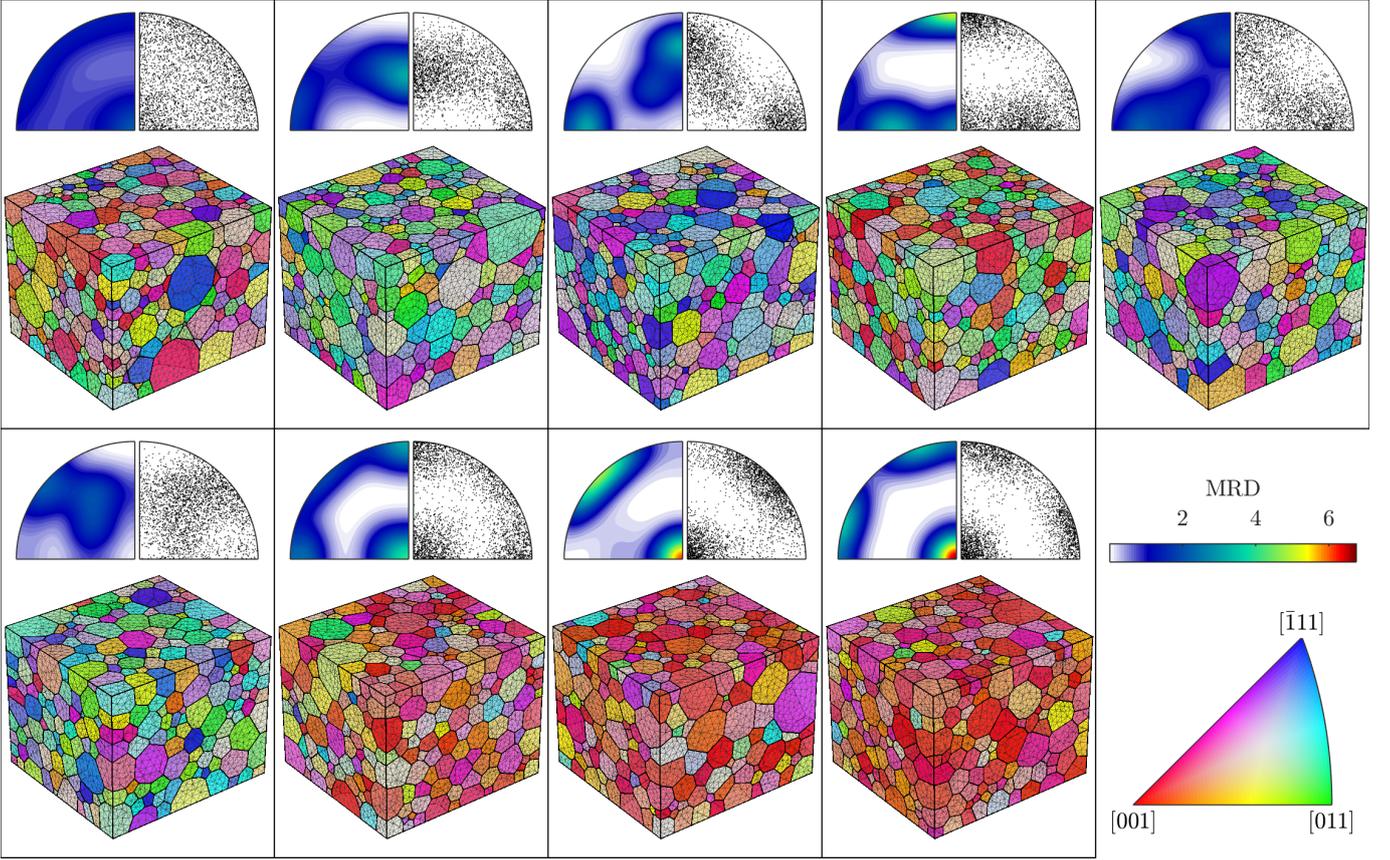


Figure 8: ODFs sampled from $M_H^{(1)}$ via HSS and their use in conjunction with synthetic polycrystal generation software. In each subfigure the kernel density estimator for the ODFs, as defined by Eq. 10, is plotted in the upper left (001) pole figure (see MRD colorbar in bottom right of figure). The orientations sampled from each ODF are shown in the upper right (001) pole figure. The meshed synthetic polycrystalline microstructures are also shown with grains colored by their orientation according to the inverse pole figure coloring scheme (legend shown in bottom right of figure).

fraction basis and a more sophisticated approach to assigning grain orientations, such as is implemented in the DREAM.3D [14] software package, would be necessary to match the ODF statistics on a volume fraction basis. Nonetheless, this simple approach suffices to illustrate the potential for coupling HSS with synthetic microstructure generation software to produce diverse microstructural sets for subsequent mesoscale simulation and analysis.

Other microstructural statistics, such as grain boundary character distribution (GBCD) functions, or triple junction distribution (TJD) functions, can also be sampled via HSS from their respective microstructure hulls ($M_H^{(2)}$ and $M_H^{(3)}$ respectively). Reconstruction of microstructures that satisfy such statistics can be accomplished via Monte Carlo methods (see [30]).

6. Conclusions

We have described a new algorithm called *hierarchical simplex sampling* (HSS) for the efficient delineation of materials properties closures and the generation of diverse microstructural sets.

We compared the performance of HSS to traditional algorithms for property closure delineation and found that HSS obtained more boundary points than traditional algorithms for a fixed run time by a factor of between 8 and 44. In addition, the properties closure resulting from the HSS method was a more faithful representation of the true properties closure, with points more evenly distributed across the closure and strictly respecting concave regions.

We also demonstrated the use of HSS to generate sets of microstructure statistics (e.g. orientation distribution functions) that span their respective microstructure hulls. The resulting microstructure sets collectively encode the theoretical range of microstructural diversity contained within the corresponding microstructure hull. As a practical use case, we illustrated the potential of coupling HSS with existing synthetic polycrystal generation software to produce diverse sets of microstructures that are fully meshed and could subsequently be subjected to mesoscale simulations (e.g. via the finite element method). We anticipate that simulations performed on such diverse microstructure sets

will explore a broader range of material responses and facilitate the derivation of more general conclusions that are not masked by the idiosyncrasies of a single microstructural instantiation.

Acknowledgments

This material is based upon work performed primarily at BYU and supported by the National Science Foundation under Grant No. 1610077. OKJ acknowledges prior funding while at MIT by the US Department of Energy (DOE), Office of Basic Energy Sciences under award no. DE-SC0008926 with C.A. Schuh as PI, and by the Department of Defense (DoD) through the National Defense Science & Engineering Graduate Fellowship (NDSEG) Program. We are also grateful for interactions with Dr. Michael Spivey that inspired the use of the particular uniform simplex sampling method employed in our tests.

References

- [1] B. L. Adams, S. R. Kalidindi, and D. T. Fullwood. *Microstructure-Sensitive Design for Performance Optimization*. Butterworth-Heinemann, 1 edition, 2012.
- [2] M. F. Ashby. *Materials Selection in Mechanical Design*. Butterworth-Heinemann, Cambridge, MA, 5 edition, 2016.
- [3] F. Bachmann, R. Hielscher, and H. Schaeben. Texture Analysis with MTEX Free and Open Source Software Toolbox. *Solid State Phenomena*, 160:63–68, feb 2010.
- [4] H. J. Bunge. *Texture Analysis in Materials Science: Mathematical Methods*. Cuvillier, 3 edition, 1993.
- [5] G. Casella, C. P. Robert, and M. T. Wells. Generalized Accept-Reject sampling schemes. pages 342–347. Institute of Mathematical Statistics, 2004.
- [6] Y. Chen and C. A. Schuh. Diffusion on grain boundary networks: Percolation theory and effective medium approximations. *Acta Materialia*, 54(18):4709–4720, oct 2006.
- [7] L. Devroye. *Non-Uniform Random Variate Generation*. Springer New York, New York, NY, 1986.
- [8] H. Edelsbrunner, D. Kirkpatrick, and R. Seidel. On the shape of a set of points in the plane. *IEEE Transactions on Information Theory*, 29(4):551–559, 1983.
- [9] H. Edelsbrunner and E. Mücke. Three-dimensional alpha shapes. *ACM Transactions on Graphics*, 13(1):32, 1994.
- [10] T. Fast, M. Knezevic, and S. R. Kalidindi. Application of microstructure sensitive design to structural components produced from hexagonal polycrystalline metals. *Computational Materials Science*, 43(2):374–383, aug 2008.
- [11] P. Fortier, K. Aust, and W. Miller. Effects of symmetry, texture and topology on triple junction character distribution in polycrystalline materials. *Acta Metallurgica et Materialia*, 43(1):339–349, jan 1995.
- [12] M. E. Frary and C. A. Schuh. Percolation and statistical properties of low- and high-angle interface networks in polycrystalline ensembles. *Physical Review B*, 69(13):134115, apr 2004.
- [13] V. Y. Gertsman and K. Tangri. Computer simulation study of grain boundary and triple junction distributions in microstructures formed by multiple twinning. *Acta Metallurgica et Materialia*, 43(6):2317–2324, jun 1995.
- [14] M. A. Groeber and M. A. Jackson. DREAM.3D: A Digital Representation Environment for the Analysis of Microstructure in 3D. *Integrating Materials and Manufacturing Innovation*, 3(1):5, dec 2014.
- [15] J. M. Hammersley and D. C. Handscomb. *Monte Carlo Methods*. Springer Netherlands, Dordrecht, 1964.
- [16] Z. Hashin and S. Shtrikman. A Variational Approach to the Theory of the Effective Magnetic Permeability of Multiphase Materials. *Journal of Applied Physics*, 33(10):3125–3131, jun 1962.
- [17] O. K. Johnson and C. A. Schuh. The triple junction hull: Tools for grain boundary network design. *Journal of the Mechanics and Physics of Solids*, 69:2–13, sep 2014.
- [18] O. K. Johnson and C. A. Schuh. Texture mediated grain boundary network design in two dimensions. *Journal of Materials Research*, 31(09):1171–1184, may 2016.
- [19] O. K. Johnson and C. A. Schuh. Texture Mediated Grain Boundary Network Design in Three Dimensions. (*Under Review*), 2017.
- [20] M. Kumar, W. E. King, and A. J. Schwartz. Modifications to the microstructural topology in f.c.c. materials through thermomechanical processing. *Acta Materialia*, 48(9):2081–2091, may 2000.
- [21] J. K. Mason and C. A. Schuh. Hyperspherical harmonics for the representation of crystallographic texture. *Acta Materialia*, 56(20):6141–6155, dec 2008.
- [22] G. Proust and S. R. Kalidindi. Procedures for construction of anisotropic elastic-plastic property closures for face-centered cubic polycrystals using first-order bounding relations. *Journal of the Mechanics and Physics of Solids*, 54(8):1744–1762, aug 2006.
- [23] R. Quey, P. Dawson, and F. Barbe. Large-scale 3D random polycrystals for the finite element method: Generation, meshing and remeshing. *Computer Methods in Applied Mechanics and Engineering*, 200(17-20):1729–1745, apr 2011.
- [24] A. Reuss. Berechnung der Fließgrenze von Mischkristallen auf Grund der Plastizitätsbedingung für Einkristalle. *Zeitschrift für Angewandte Mathematik und Mechanik*, 9(1):49–58, 1929.
- [25] R. Rubinstein. Generating random vectors uniformly distributed inside and on the surface of different regions. *European Journal of Operational Research*, 10(2):205–209, jun 1982.
- [26] C. A. Schuh, M. Kumar, and W. E. King. Universal features of grain boundary networks in FCC materials. *Journal of Materials Science*, 40(4):847–852, feb 2005.
- [27] R. L. Smith. The hit-and-run sampler: a globally reaching markov chain sampler for generating arbitrary multivariate distributions. In *Proceedings Winter Simulation Conference*, pages 260–264, Coronado, CA, USA, dec 1996. IEEE.
- [28] W. Voigt. *Lehrbuch der Kristallphysik*. B.G. Teubner Verlag, Leipzig, 1928.
- [29] X. Wu, G. Proust, M. Knezevic, and S. R. Kalidindi.

Elasticplastic property closures for hexagonal close-packed polycrystalline metals using first-order bounding theories. *Acta Materialia*, 55(8):2729–2737, may 2007.

- [30] C. Yeong and S. Torquato. Reconstructing random media. *Physical Review E*, 57(1):495–506, jan 1998.