Quantification of 3D spatial correlations between state variables and distances to the grain boundary network in full-field crystal plasticity spectral method simulations

Quantification of 3D spatial correlations between state variables and distances to the grain boundary network in full-field crystal plasticity spectral method simulations

Markus Kühbach, Franz Roters

Max-Planck-Institut für Eisenforschung GmbH,
Max-Planck-Straße 1, D-40237 Düsseldorf

E-mail: m.kuehbach@mpie.de

Abstract. Deformation microstructure heterogeneities play a pivotal role during dislocation patterning and interface network restructuring. Thereby, they affect indirectly how the microstructure recrystallizes. Given this relevance, it has become common practice to study the evolution of deformation microstructure heterogeneities with 3D experiments and full-field crystal plasticity computer simulations including tools such as the spectral method.

Quantifying material point to grain or phase boundary distances, though, is a practical challenge with spectral method crystal plasticity models because these discretize the material volume rather than mesh explicitly the grain and phase boundary interface network. This limitation calls for specific data post-processing methods to quantify the spatial correlations between state variable values at each material point and the points' corresponding distance to the closest grain or phase boundary.

This work contributes to the development of advanced such post-processing routines. Specifically, two grain reconstruction and three distancing methods are developed for solving above challenge. The individual strengths and limitations of these methods surplus the efficiency of their parallel implementation is assessed with an exemplary DAMASK large scale crystal plasticity study. We apply the new tool to assess the evolution of subtle stress and disorientation gradients towards grain boundaries.

Keywords: Crystal plasticity, gradients, interfaces, slip transfer, data-mining, parallelization, DAMASK

1. Introduction

Heterogeneities of constitutive and microstructural state variables, such as stress, strain, orientation, or dislocation density, play a pivotal role during metal forming and downstream annealing treatment [1–3]. These heterogeneities form as a consequence of

local differences during dislocation self-organization [1, 4–7] or geometrical constraints which the crystal interfaces imprint on the individual deformation mechanisms [8]. The accumulation of misorientation and dislocation density at such interfaces [9–12], in turn, generates local conditions and gradients which facilitate the formation of dynamic or static recrystallization nuclei [13–17]. Therefore, considerable experimental effort [8,18] has been devoted to the characterization of these deformation microstructure heterogeneities, both in 2D [19–24] and recently also 3D [25,26]. Nowadays, it is common practice to investigate the evolution of the deforming microstructure with full-field 3D crystal plasticity representative volume element (RVE) computer modeling [27–29] and compare explicitly such results to experiments [18, 30–38].

Most commonly, such computer simulation studies report descriptive statistics of tensorial state variable values. These are either presented as histograms of state variable values or colorful renditions of the RVE domain or specific RVE sections. For characterizing how state variable values are distributed in the RVE volume, though, one should instead quantify the distance of each RVE material point to its closest interface point and evaluate the distributions of state variable values as a function of these distances.

We acknowledge that such quantification task faces challenges when one aims in addition for highest possible statistical significance and uses full-field spectral method models [38–45]: firstly, simulations for heterogeneity build-up demand for substantially finer spatially resolved RVE domains than are typically used for flow curve predictions. This increases numerical costs because statistical significance for texture and gradients demands to include not only as many grains but also an as fine as possible spatial resolution for each of them; all to reflect accurately that dislocation density and orientation gradients build-up in 3D as a function of grain orientation and neighborhood [11]. For such costly simulations the key advantage of spectral method models with periodic boundary conditions are lower numerical costs than compared to similarly resolved finite element crystal plasticity models [44]. Nevertheless, this advantage comes with a second challenge: instead of probing explicit interface networks, now additional post-processing costs accrue to recover and characterize the interface network.

The requirement to analyze distances in the deformed configuration, i.e. distances as they are measured in an experiment, defines a third challenge: the successive shape distortion of the RVE domain has to be accounted for. This is challenging for periodic boundary conditions: ideally, the grains in contact with the RVE domain should get analyzed as well as to not make compromises in statistics. The grains are, though, frequently not only physically fragmented but also logically represented by a composite of fragments on opposite sides of the RVE domain. Another challenge to master during any quantification of spatial correlations is to collect statistics with strongest significance. For simulations with several million material points and eventually dozens of strain steps this calls for efficiently parallelized post-processing tools to enable terabyte volume analysis productivity. As a fifth challenge, any post-processing of state-variable-to-interface-distance correlations for spectral method models calls for grain

reconstruction methods because as the grains get deformed they accumulate internal misorientation.

At least for the grain reconstruction challenge, advanced post-processing tools were developed in the SEM/EBSD community [46–51]. Consequently, we assessed these [49, 52–54] for their potential and application to solve above reconstruction and distancing task. Not a single tool, though, combined grain reconstruction functionality for three-dimensionally distorted surplus periodically constrained point cloud data with documented application for multi-hundred million points.

This gap motivated our research which contributes methods that cope with all above listed challenges. Specifically, we implement statistical quantification methods which characterize how state variable values are distributed as a function of material point distance to grain boundaries. We implement two grain reconstruction methods and apply these in combination with three different distancing methods to compile a comparative assessment of long range spatial distributions for state variable values. The case study in this work focuses on single-material-point-resolved quantification of the Cauchy stresses and disorientation with respect to the mean orientation of the grain. These are exemplar quantities of frequent interest and controversy with respect to their tendency to form spatial gradients at interfaces.

Furthermore, we report how such methods can be implemented into an efficient strong and weak scaling tool supplementing the Düsseldorf Advanced Material Simulation Kit (DAMASK [38]). Open source tools like this should encourage the crystal plasticity community to explore also their 3D RVE data in further quantitative detail.

2. Methods

2.1. Defining the data analysis task

This work presents a specific analysis workflow to post-process DAMASK [38, 43, 44, 55] simulation results. Two grain reconstruction, three material-point-to-interface distancing methods, and additional post-processing algorithms were developed. The workflow enables the quantification of spatial correlations between specific material point state variable values and each material points' closest projected distance to an interface. The individual steps of the workflow are detailed in the appendix. The workflow was implemented into a parallelized original C/C++ program. Given that the entire source code is provided as open source, it suffices to report the key steps of the workflow and the strategies to their efficient implementation.

The workflow answers the following data mining question: how are state variable values distributed spatially as a function of the projected distance to the nearest interface for a given collection of strain steps? We define a strain step as a state variable value dataset which is probed from an RVE crystal plasticity simulation at a specific true strain value. From now on, it is assumed that this dataset contains results for every

material point which supports the RVE domain.

Interfaces are either grain boundaries or phase boundaries. In this paper, we focus on single phase microstructures. The methods are formulated general enough to be applicable for phase boundaries, provided their geometry is sampled by material points. We report 3D applications of the methods. The simplification to 2D is straightforward.

Under above provision the input data for each strain step consists of all N material points. Identified as p_i, these define positions x_i in the deformed configuration. The results are defined and analyzed in the laboratory Cartesian coordinate system $\in \mathcal{R}^3$. As such, all local displacements of a material point versus its initial position in the unloaded case are accounted for [43]. This makes the results comparable to the situation in experiments where a plastically deformed microstructure is probed typically by a posteriori sample preparation. For this coordinate system convention the initial grid of integration points becomes successively and irregularly distorted (Fig. 3a). This calls for the development of point cloud processing methods which back out the geometry of each grain and consider the fact that typical DAMASK simulations use full three-dimensional periodic boundary conditions.

Each material point i has an associated set of state variable values $\{\boldsymbol{s_i}\}$. Members of this state variable set are for instance the local deformation gradient $\boldsymbol{F_i}$ or the orientation $\boldsymbol{q_i}$. Although not all these variables are state variables in the microstructural sense, we have opted, for simplicity, to refer to these computed quantities as state variables.

Different strategies are in place in the crystal plasticity community to define an initial grid of integration points for sampling the microstructure in the RVE volume. Cubic grids are the most commonly employed strategy. Also full-field spectral method crystal plasticity simulation tools like DAMASK [38, 44] use such grid. This avoids explicit bookkeeping of the interface network geometry and thus cuts numerical costs. As a disadvantage, though, the interfaces have to be reconstructed through post-processing to enable a quantification of above distance correlations. Below workflow presents an original solution to solve this task; implemented as a supplemental DAMASK post-processing data mining tool, called damaskpdt for short. Here, the acronym pdt stands for post data treatment.

2.2. Pre-processing stage

The processing of the strain step data is distributed across the nodes of a cluster computer. Each process handles complete strain steps (at least one). The processes read independently all material point and respective state variable values from the simulation results file. In this work, data are parsed from the traditional binary container format file. An extension of the tool to use the HDF5 based container format in DAMASK [38, 45] remains as a task for the future.

After loading the strain step data, each process executes a pre-processing stage. This includes the calculation of the material point positions x_i in the deformed configuration, the equivalent stress and strain tensors for each material point, and an

RVE-averaging of these quantities using methods previously described [38, 43, 44, 55]. Next, three point clouds are defined for each strain step: the first point cloud is $\mathcal{P}_n^0$. It specifies the ensemble of the original N material points in the deformed configuration. The second point cloud $\mathcal{P}_n^{0+\epsilon}$ specifies the ensemble $\mathcal{P}_n^0$ surplus all its 26 periodic image points inside a bounding box with thickness ϵ about the bounding box to $\mathcal{P}_n^0$. Here, ϵ defines a small fraction of the largest edge length of the box. The third point cloud $\mathcal{P}_n^1$ specifies all members of $\mathcal{P}_n^0$ surplus all their 26 periodic image points. Subsequent processing involves region queries on all three point clouds. Special data structures were used to accelerate these queries (section 2.5). The purpose of using multiple point clouds instead of always the complete set $\mathcal{P}_n^1$ is to optimize and reduce memory utilization. As an example, it suffices to work with $\mathcal{P}_n^{0+\epsilon}$ instead of $\mathcal{P}_n^1$ when querying neighboring material points at the RVE boundary within radii of at most ϵ during the disorientation based distancing. Further details to the purpose of the point clouds are provided in the algorithms in the appendix.

2.3. Grain reconstruction

Remark on classical grain reconstruction with DAMASK Attempts were made in the past to reconstruct the grains through application of cluster search algorithms on $\mathcal{P}_n^0$: the grains were built as a cluster of material points with the same texture ID within a critical distance and less than an a priori defined disorientation accumulated among each other. Unless a very large critical distance is chosen, though, this method cannot merge all periodic images of a grain, thus reducing the significance of the analysis. Large distances, though, will occasionally merge second- or even higher-order neighboring grains into a single object. Especially so when these neighbors have similar orientations.

Instead, two different approaches are proposed to reconstruct the grains. The methods borrow conceptually from achievements made in the SEM/EBSD community [46, 49, 54] but add support for periodic boundary conditions and hierarchical software parallelization. Identifying the shape of the grains is a key step of every interface network reconstruction. With this process, material points get labeled as grains based on their orientation and point neighborhood. In what follows, two methods are specified for such labeling of the polycrystalline aggregate:

Modified grain reconstruction methods developed in this work The first method relabels the texture ID from the DAMASK microstructure synthesis step as the grain ID. This has been the common strategy so far when studying in-grain orientation gradient build-up: either explicitly [9, 12] or implicitly by reporting orientation spread via pole figures [11]. This grain reconstruction method is tagged TEX in the results section.

The second method addresses the challenge that when grain fragmentation becomes significant, referring still to the initially instantiated grains is no longer correct. Instead, the individual grain fragments should be distinguished. Useful methods for this task meet three requirements: they work for irregular point cloud data, successfully handle

three-dimensional periodic boundary conditions, and are sensitive to gradual build-up of orientation gradients.

Graph clustering is one method which fulfills all above criteria. Application examples for characterizing microstructures with graph clustering methods, though, remain few. One is known in the SEM/EBSD community as Fast Multiscale Clustering (FMC) [56, 57]. Other examples, on which this work settles, are community detection algorithms [58, 59], which find frequent application in analyzing human communication and interaction patterns among individuals on social media platforms. This work uses the community detection method employed in [59]. Specifically, we use the open source implementation of the Louvain community detection method by Blondel et al. [60, 61]. This grain reconstruction method is tagged LOU in the results section.

Extraction of a single periodic image per grain and bounding box ensemble After labeling each material point one can in principle compute the distance correlations. For arbitrarily deformed domains with periodic boundary conditions, though, practical challenges remain: first, macroscopic and microscopic shape distortions of the RVE domain walls have to be accounted for. Second, grains in contact with the RVE walls are fragmented into multiple pieces. Therefore, it is useful to merge the fragments into a single grain with simpler geometry. Placing it in a cuboidal box is even better because it decouples all downstream processing into grain-local operations.

Consequently, a geometry simplification step is executed. Therein, the individual periodic fragments are first fused using a DBScan [62] clustering algorithm. Specifically, all grain fragment neighbors within a threshold radius of $R = \sqrt{3}L_0$ with L_0 specifying the initial point-to-point distance are fused. Next, the closest replicate periodic image to the center of $\mathcal{P}_n^0$ is chosen and the associated material points are packed into a grain-local bounding box. Once boxed up, efficient multithreaded processing methods from the grain growth modeling community [63–65] speed up the grain-local analyses.

After the geometry simplification the global bounding box to all grains' local bounding boxes are computed (Fig. 2b) and define a rediscretization of this global bounding box. The actual rediscretization, though, is applied inside each grain-local box only to improve efficiency. Once rediscretized, the information contained in $\mathcal{P}_n^1$ is used to identify the nearest neighboring material point to each voxel. Cubic voxels of edge length $0.5L_0$ were used.

2.4. Distance quantification

Disorientation based For this distancing method, tagged DIS in the results section, the challenge of grain reconstruction is ignored completely. Instead, position and orientation data for points in $\mathcal{P}_n^{0+\epsilon}$ are used. The quantity ϵ defines a guard zone thickness that was chosen as 0.1 times the initial RVE domain edge length. The distance values are computed by finding for all material points in $\mathcal{P}_n^0$ all their respective neighbors within a region of radius $R = 0.1$, where R is again in multiples of the initial RVE domain edge

length.

Next, the identified neighbors are sorted in increasing order with respect to their Euclidean distance to the inspected material point. Finally, the distance value for the closest, above a critical threshold Θ_c disoriented pairing, is taken as the distancing result. In this work, a threshold of $\Theta_c = 15°$ was used. If no closest neighbor among the candidates was found, the material point made no contribution to the correlation statistics.

In a nutshell, the advantage of the DIS distancing method is that it requires no grain reconstruction as long as no disorientation-to-distance correlations are desired. The independence of each material point is another advantage which makes the calculations trivial parallelizable. Without taking the spatial arrangement of neighboring points into account, though, a disadvantage of DIS distancing is that only scalar distances are accessible.

Signed distance / voxelization based Enabling the identification of normal distances to the grain boundary is the motivation for and key strength of the second distancing method. It is tagged as SDF in the results section. The key idea is to evaluate signed distance level set functions in combination with latest method developments achieved for scalable grain coarsening simulations [65–68].

Level set function methods parameterize the position of interfaces implicitly as the iso-contour of a real-valued level set function $\Phi(x)$ to an a priori defined iso-value. Defining a signed distance level set function (SDF) in combination with each boundary contour of a grain being defined as $\Gamma := \{x \in \Omega \mid \Phi(x) = 0\}$ with $\Gamma \in \Omega \in \mathcal{R}^3$ is a particular useful choice. Namely, this choice, i.e. $\Phi(x) \in \Omega \in \mathcal{R}^3$, has the key advantage that $\|\nabla\Phi(x)\| = 1$ holds for most points $\in \mathcal{R}^3$ one practically encounters inside and outside the contour. Consequently, a consistent outer unit normal vector can be computed to each point on the contour by evaluating $\boldsymbol{n(x)} = \frac{\nabla\Phi(x)}{\|\nabla\Phi(x)\|} = \frac{\nabla\Phi(x)}{1}$ numerically. In effect, this allows calculating arbitrarily projected distance vectors to the contour.

A two step procedure was employed to compute a signed distance function for each grain. The procedure builds on the previously defined discretization of the bounding boxes about each merged grain. First, $\Phi(x)$ was initialized to positive values $(+h)$ for all voxel assigned to the grain under inspection and negative values $(-h)$ for points outside. The scalar h was set equal to the cell width used for the global rediscretization of the domain. In a second step, the distance information was propagated with the fast sweeping method [65, 69].

Tessellation based Also the third method, tagged VOR in the results section, delivers vectorial distances. Furthermore, it explicitly reconstructs a contour hull for each grain. The key idea of the method is to build up each grain from a collection of Voronoi polyhedra. These can be directly obtained for each grain from a Voronoi tessellation of the respective material point cloud. Formally, the VOR method is based on the

approach of Bachmann et al. [54]. We introduce two improvements, though: first, a generalization of the method for applications on distorted point clouds with periodic boundary conditions. Second, we introduce a hybrid-parallelized solution which executes substantially faster and might provide an avenue to explore for MTEX in the future as well. This parallelization alleviates much of the higher numerical costs, which the VOR method has in comparison to the DIS or SDF methods. VOR does not require a rediscretization of the bounding box to each grain k but operates directly on the grain-local subset of material points from $\mathcal{P}_n^1$.

In a first step, this local point cloud is extracted and each Voronoi cell computed. Each Voronoi cell ets labeled with the respective grain ID k and material point ID. Next, the algorithm identifies which of the cell facets have first-order neighboring Voronoi cells with a different associated grain ID than the currently processed. Such difference is a signature that the currently processed Voronoi cell makes contact with the grain boundary.

The resulting collection of all such facets of Voronoi cells about points tagged with the grain ID under inspection is used to construct the shape of the exterior contour hull (Fig. 1b). The contour hull extraction algorithm is executed in parallel for each grain k.

The Voro++ tessellation library [70] was used to process the individual grain-local tessellations. Specifically, we re-implemented the library wrapping approach of Peterka et al. [71] around calls to this library. To avoid truncating Voronoi cells of the inspected grain, the individual grain-local bounding boxes were enlarged by a guard zone of three times the initial point-to-point distance. Furthermore, possible Voronoi cell contact with the local bounding box domain walls was detected using Voro++ library functionalities. These steps cured any possibly truncated Voronoi cell, eventually through a recomputation after fattening the guard zone locally.

Once the exterior contour hull has been defined, the material point to contour hull distances can be computed. Given that projections on the polyhedron facets can be used, the VOR methods computes projected distances. The implementation is tricky, though, in detail because consistence and speed are desired:

Algorithm to compute vectorial distances within VOR Using the edge circulation conventions of the Voro++ library, first a consistent outer unit normal vector for each contour hull facet is computed. Next, the normal and geometry of the facet is evaluated to compute a consistent right-handed local Cartesian coordinate system for each contour hull facet. Such coordinate system can be defined because each Voronoi cell is a convex polyhedron [72]. By virtue of construction each facet is a convex polygon in 3D space that is reduced, thanks to the local coordinate system to two dimensions. These local coordinate systems accelerate point to polygon inclusion tests.

It remains to compute the distances. First, the general approach used is explained. Second, we detail how the processing of the facets gets accelerated with bounded volume hierarchy querying algorithms. The distances are Voronoi cell interior-point-to-exterior-

contour-hull-projected point distances. Distances were evaluated for each interior point. It suffices to sketch the procedure for a single interior point. The main task is to probe which facet of the contour hull lays closest to the point. While probing the facets for the absolute shortest projected distance the algorithm works as follows: firstly, each interior point gets normal projected onto the facet plane.

Secondly, it is probed whether the projected point lays inside the facet polygon or on its contour respectively or outside. The distance value is eventually used to update the currently closest distance.

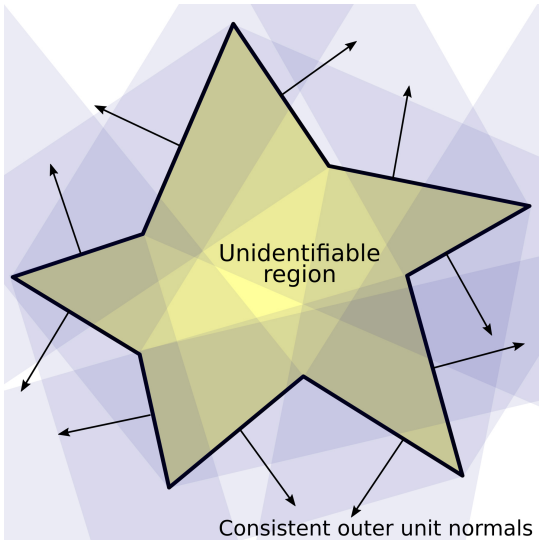
If, though, the projected point is outside the polygon, one proceeds by normal projecting the currently tested interior point on each edge segment of the polygon. If such a projected point lays on the edge, the computed distance eventually updates the current closest distance.

If none of these projections map on a polygon edge segment, we evaluate the Euclidean distance of the currently tested interior point to each vertex of the polygon and update eventually thereby also the current closest distance. Once finished with the vertices, the algorithm proceeds with the next facet of the contour hull until no further facets remain for processing.

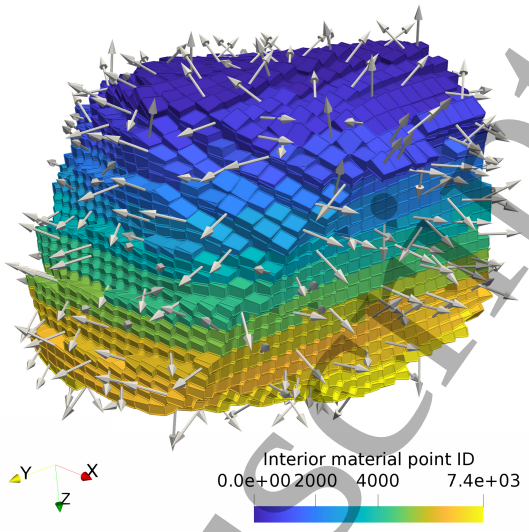
To the best of our knowledge advanced procedures like above are necessary to handle cases where, due to the possible non-convexity of the composed contour hull, a consistent normal distance is not immediately defined. In fact, such cases are frequently possible. One particular simple example is illustrated with the grayed-out region in Fig. 1a. Evidently, there is not necessarily a projected point on a contour facet for every position inside an arbitrarily shaped non-convex polygon (here for the brighest yellowish region). At least not if one projects exclusively parallel to the direction of the facets' outer unit normals.

While getting access to a defined contour hull for each grain surplus projected distances is a clear advantage, the VOR method has also disadvantages: most importantly, higher numerical costs. More subtle, the edge-on compositing of a contour hull from a collection of polygons faces discontinuities of the curvature at the polygon edges. These can be cured, though, with state of the art SEM/EBSD grain reconstruction software [49, 52] which is another possible avenue to explore and use for reconstruction of the interface network in DAMASK in the future.

A strategy for accelerating the distance computations Without additional tricks, the algorithm defines a quadratic time costly computational geometry task – for each interior point each contour facet needs inspection. Therefore, a pruning strategy was designed to reduce the total number of facets tested. The key idea is to use the shorter and shorter identified distance values to occasionally requery which contour hull facets require testing. Therefore, a bounded volume hierarchy (BVH) was built from the contour hull facets [73–76]. This utility data structure organizes the locations of the facet polygons into spatial regions, such that with narrowing down the search fewer and fewer candidates need testing.



(a) Distancing contours with mixed curvature signs



(b) Exemplar grain contour hull

Figure 1: a) Compositing Voronoi cells into a larger polyhedron can result in a polyhedron with concave and convex sections. This demands additional care when computing projected distances to the contour hull facets. b) An exemplar result of the tessellation based contour hull extraction algorithm. A random subset of outer unit normal vectors is shown.

First, we probe an arbitrary candidate and query the BVH for the resulting distance value. This value is used for querying the BVH to narrow down which candidate facets to probe next. Thereafter, the algorithm continues iteratively: each time the distance value gets lower than 0.9 of the last query, the candidate list gets updated until the finding the absolute shortest distance per interior point.

Distilling spatial distributions of disorientation and stress The results of each damaskpdt tool run are two ensembles of Cauchy stress tensor/distance and orientation quaternion/distance value pairs per material point. These data were finally processed with MATLAB (v2017a) / MTEX [77, 78] texture toolbox (v5.0.3) scripts to obtain spatial distributions of stress and disorientation.

A two stepped protocol was executed to quantify the disorientation of each material point to the respective mean orientation of the reconstructed grain: first, a mean orientation was computed for each grain. Documented methods for inferential quaternion statistics [79] were used for this task. Secondly, these mean orientations were evaluated against the individual orientation of each material point.

The so generated value/distance pairs were binned into distance bins on the interval $[0.0, 24.0]$ using 0.2 step. Length units are reported in multiples of the initial [100]-direction point-to-point distance between neighboring material points L_0. Quantile values of the resulting sub-distributions were quantified with MATLAB . Grain size

distributions report how many material points were assigned per grain.

2.5. Parallel implementation and orchestrated hierarchical data placement

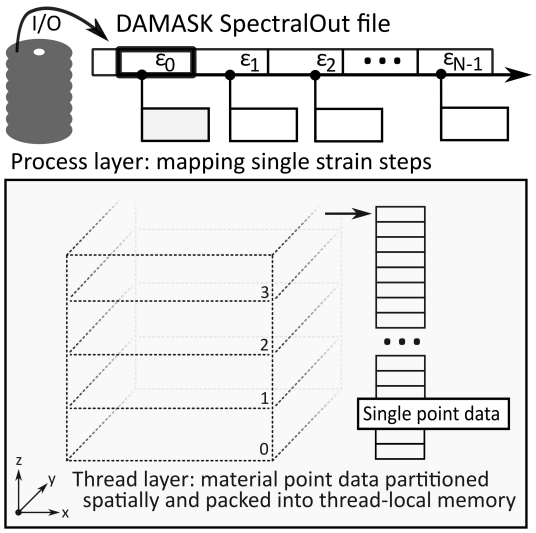
Above workflow was implemented into a hybrid-parallelized tool. Inherent parallelism in the data mining task was exploited: for the reason that each strain step is independent, we distribute the processing of the strain step ensemble via process data parallelism. Specifically, calls to the Message Passing Interface (MPI) library [80, 81] were implemented. Strain steps were distributed across the MPI processes in round-robin fashion. In addition, the processing of each strain step was accelerated via Open Multi-Processing (OpenMP) [65, 82–87]. Instead of using the commonly employed referencing of data items via global arrays, we actively enforce a partitioning of the material point positions and state variable values. Specifically, all derived quantities are split into thread-local data chunks. These chunks are placed preferentially in memory locations with fast connection to the execution core. This facilitates faster reading from memory through closer packing of data and less cache coherency traffic. We admit that our implementation currently uses a static load partitioning scheme whereby load imbalances may be stronger than for dynamic workload distributing. This is a possible disadvantage to investigate in the future.

To efficiently execute all linear algebra operations of the workflow, fast Fourier transformations (FFT), singular value decomposition, and eigenvalue decomposition functionalities of the Intel Math Kernel Library (IMKL) were used.

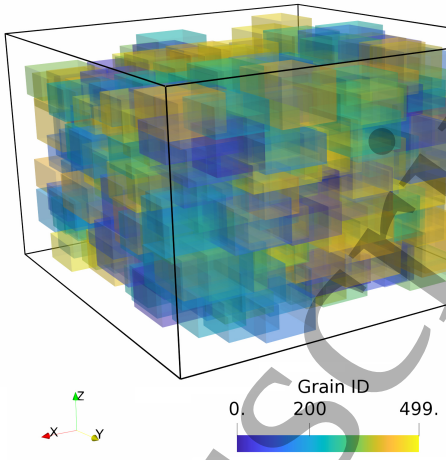
The subsequent grain segmentation and characterization of the spatial descriptive statistics involves range querying tasks. Specifically, they demand the identification of all neighboring points inside a spherical volume of radius R. Using 3D box binning with cubic buckets of width R for $\mathcal{P}_n^{0+\epsilon}$, and $\mathcal{P}_n^1$ enabled constant time complex queries. A disadvantage of this technique compared to other established fast position querying techniques [73,88,89] is the larger memory footprint. This detail remains as future work for a possible code optimization aiming at a lower memory consumption.

2.6. Verification and validation simulation setup

Microstructure instantiation We applied above methods for studying the evolution of the spatial distribution of stresses and disorientation in full-field crystal plasticity simulations. Simulations were executed with the DAMASK spectral solver [38, 44, 55] and used the DAMASK Poisson-Voronoi tessellation microstructure synthesis and orientation sampling routines. Specifically, a 3D face centered cubic single phase polycrystal RVE was instantiated. Full three-dimensional periodic boundary conditions were used. The RVE domain contained 500 grains. These were discretized by 256^3 grid points, i.e. material points. In effect, each grain has an average spherical equivalent radius of 20 material points. Grain orientations were sampled randomly and assigned in a spatially uncorrelated manner.



(a) Hierarchical data partitioning and mapping approach



(b) Bounding box ensemble

Figure 2: a) The processing of the strain step ensemble is partitioned hierarchically: processes are assigned always complete strain steps. These are machined off using OpenMP multithreaded data parallelism. Each process partitions the heavy data further to allow for a mapping of thread-local data into specific memory sections to improve spatial and temporal memory locality.

Distance verification To the best of our knowledge there is no analytical solution which quantifies the distribution of shortest projected distances for an arbitrarily distorted point cloud to a given grain boundary interface network. As such, the verification of above methods is challenging. For Poisson-Voronoi microstructures, though, one can find exact linear algebra solutions for a given point cloud and evaluate them numerically. By virtue of construction the above defined microstructure is a discrete approximation of a specific Poisson-Voronoi tessellation.

Consequently, we verified our implementation as follows. First, the center positions of above grains were taken to define the corresponding fully-periodic Voronoi tessellation in continuous space. Second, the shortest projected distance to all facets of the enclosing Voronoi cell was probed for each material point. The resulting distances were binned using the same protocols.

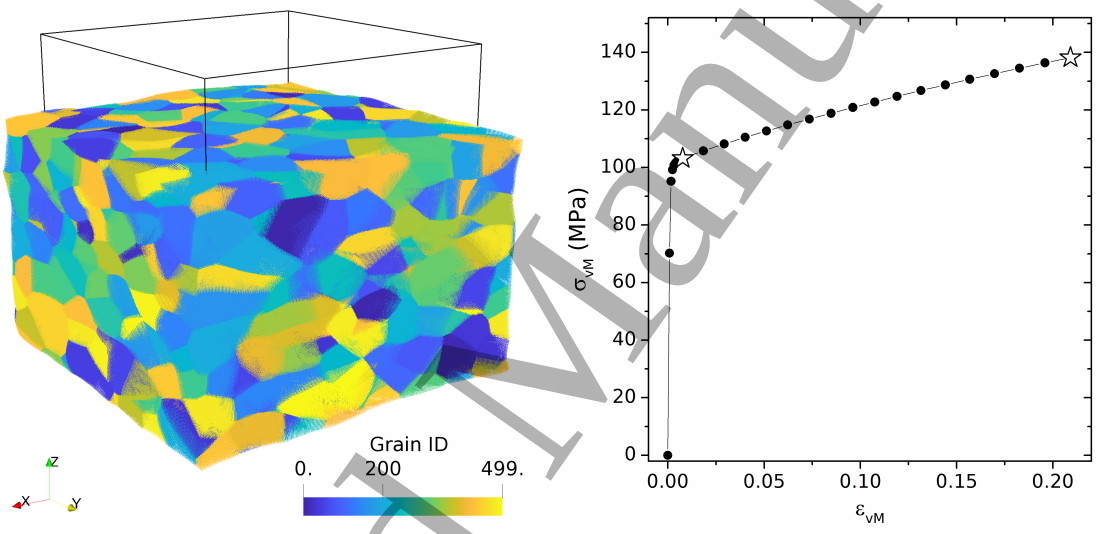
Constitutive model A phenomenological crystal plasticity model [90] was used. The tensor of elastic constants was parameterized based on experimental data for Aluminium [91]. The tensor components were $C_{11} = 106.75\,\mathrm{GPa}$, $C_{12} = 60.41\,\mathrm{GPa}$, and $C_{44} = 28.34\,\mathrm{GPa}$, respectively. Dislocation slip on the 12 primary slip systems was assumed as the exclusive deformation mechanism. The phenomenological model was parameterized with a reference strain rate of $\dot{\gamma}_0 = 0.001\,\mathrm{s}^{-1}$ and a stress exponent of $n_{slip} = 20$. Stress parameters of this constitutive model were set to $\tau_0 = 31\,\mathrm{MPa}$, $\tau_{sat} = 63\,\mathrm{MPa}$, the

hardening parameters were assumed as $a_{slip} = 2.25$ and $h_0 = 75\,\mathrm{MPa}$, respectively. Isothermal conditions and initially stress free grains were assumed.

Load case The RVE was subjected to uni-axial compression along the z-axis, simulated by a deformation gradient rate $\dot{\overline{\boldsymbol{F}}}$ (Eq. 1) with setting Piola-Kirchhoff $\overline{\boldsymbol{P}}$ stress boundary conditions (Eq. 1). Stars identify unconstrained values.

$$\dot{\overline{\boldsymbol{F}}} = \begin{pmatrix} * & 0 & 0 \\ 0 & * & 0 \\ 0 & 0 & -0.1 \end{pmatrix} \mathrm{s}^{-1} \qquad \overline{\boldsymbol{P}} = \begin{pmatrix} 0 & * & * \\ * & 0 & * \\ * & * & * \end{pmatrix} \mathrm{Pa} \tag{1}$$

The resulting RVE domain is visualized in Fig. 3a exemplified for the deformed configuration in the lab coordinate system at a final equivalent von Mises strain $\overline{\epsilon_{vM}} = 0.21$.



(a) Deformed RVE (b) Flow curve prediction with strain steps

Figure 3: The methods were applied to a full-field simulation of a 256^3 material point cube representing a polycrystal with 500 grains. State variable values for every material point were monitored at specific total strain values as shown in b). Strain steps taken at yield ($\overline{\epsilon_{vM}} = 0.01$) and incipient fragmentation ($\overline{\epsilon_{vM}} = 0.21$) (indicated by the stars) are discussed in detail.

Simulation settings and monitoring Default settings of the DAMASK spectral solver were used. At least 4 and at most 40 iterations per strain step were allowed. The simulation was executed on an in-house workstation (Tab. 1) using it exclusively and employing one MPI process with 36 OpenMP threads without pinning the threads. Microstructure snapshots were written at 28 selected strain values which the black dots in Fig. 3b detail. The following quantities were logged for each material point i at each strain step point: the volume, the deformation gradient $\boldsymbol{F}$, its plastic $\boldsymbol{F_p}$ and elastic part

$\boldsymbol{F_e}$, the first Piola-Kirchhoff stress tensor $\boldsymbol{P}$, orientation quaternion $\boldsymbol{q}$, and a reference ID, the so-called texture ID $\mathcal{T}_i$. The latter integer specifies the discrete orientation ID of each material point. The resulting binary spectralOut file occupied 162 GB disk space.

Simulation data post-processing Post-processing was executed on TALOS, a cluster computer (Tab. 1). Different levels of hybrid, i.e. process and thread parallelization were benchmarked. Computing nodes were used exclusively and threads pinned (OMP_PLACES=cores). Explicit calls to the MPI_Wtime and omp_get_wtime functions were used to monitor how much time the individual workflow steps took. I/O and non-I/O operations were distinguished. The main virtual and resident set size memory consumption was probed at the node level by parsing on the fly from the /proc/self/stat system file.

The strong multithreading speed of the tool was benchmarked with runs on a single TALOS node. Specifically, quasi-sequential runs with one process spawning one thread were compared to repetitive runs of the same study with one process spawning 40 threads. The multi node weak scalability was probed by comparing the single node results with runs employing 28 MPI processes each of which spawning 40 threads. I/O operations were performed using a GPFS parallel file system built on top of 12 logical disks. Files were striped across a RAID6 array with 10 disks each (in 8+P+Q [92] configuration). The low level stripe size was 4 MB. Supplemental MTEX post-processing was performed sequentially using a desktop PC. Given that the processing consumed only a few core hours in total, these analyses were not benchmarked.

2.7. Software details

DAMASK (git commit ID: v2.0.1-992-g20d8133) was compiled with GNU v8.2 using O2 optimization. The program was linked against the PETSc v3.11.0 numerical and the MPICH v3.3 MPI libraries. The damaskpdt post-processing tool (git commit ID: caea9c52848809e0635ba7d0afb313a592e3d0dd) was compiled with the Intel Compiler and Performance Suite (v2018.4) using O3 Skylake optimization. The program was linked against the corresponding Intel Math Kernel Library (IMKL) and Intel MPI versions. Additionally, Boost (v1.66) [93] and the Voro++ (v0.4.6) [70] libraries were linked to the tool.

Table 1: Technical details of the computing systems used. C/S means hyperthreading core pairs per socket. Mem stands for main memory in GB.

System	**CPU**	**C/S**	**Mem**	**Operating system**
Workstation	Xeon Gold 6150	18/2	576	Ubuntu 18.04.2 LTS
Cluster	Xeon Gold 6138	20/2	188	SUSE Linux Enterprise Server 15

3. Results

3.1. Verification of the distance computations

First, we verify the distribution of distances through comparison to solutions for the initial configuration. Figure 4a reports the distances in multiples of L_0, i.e. the point-to-point distance in the unloaded configuration. Like at yield, in this configuration, the shape of the RVE is undistorted. Exemplified for the method combination DIS/TEX the histograms show quantization effects inasmuch as all the possible voxel-to-voxel center distances are recovered (L_0, $\sqrt{2}L_0$, and $\sqrt{3}L_0$). This approximates well the analytically accurate distribution of the actual normal distances of the voxel center to the Voronoi facets in continuous space. The figure substantiates that without injecting additional local grain boundary normal information, our methods are voxel accurate only. Given that a fixed number of material points is probed, a larger number of cases with shorter distances implies finding a lower number of the absolute largest distances in the RVE when comparing to the continuous space solution.

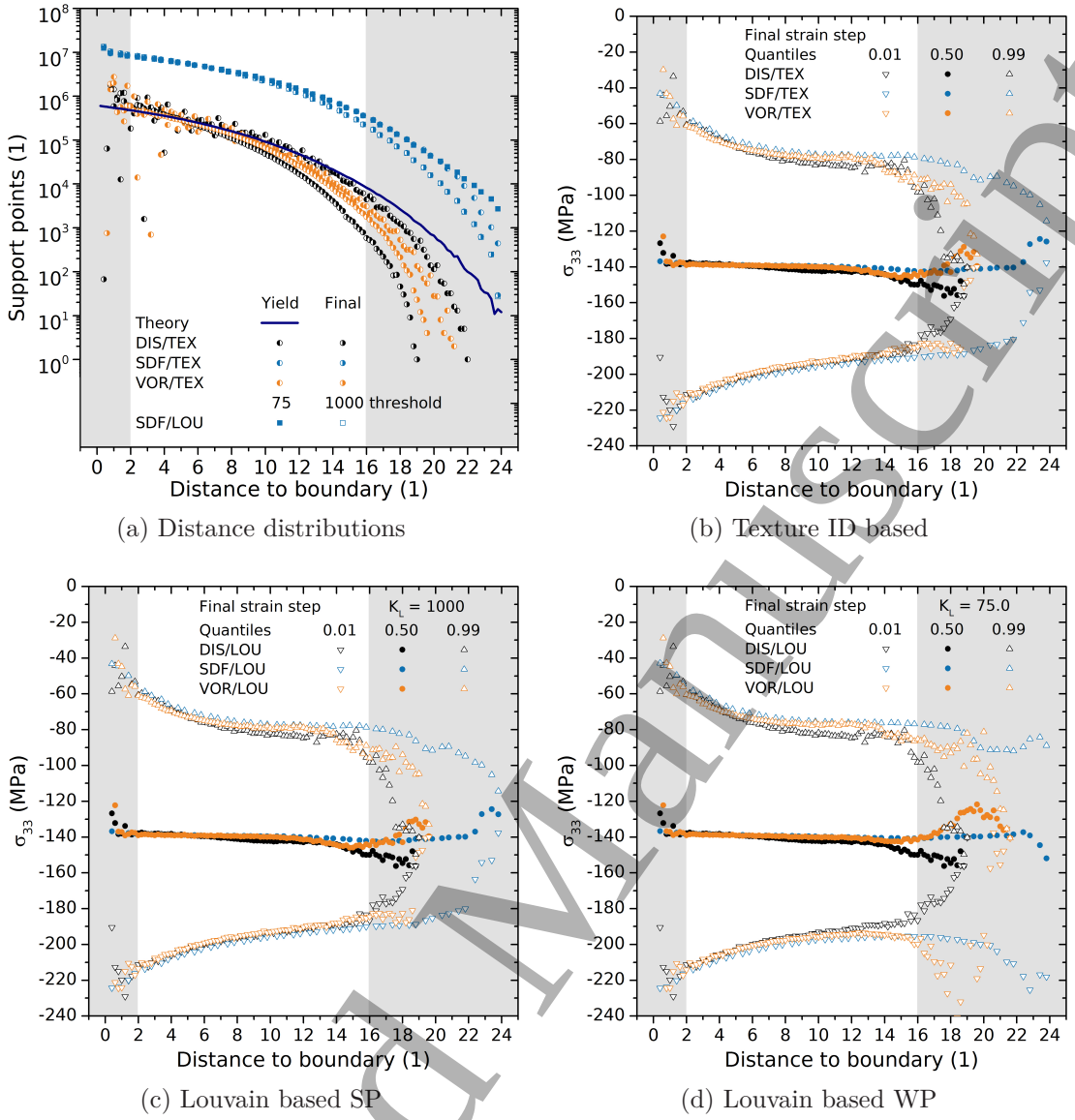
3.2. Quantifying the spatial distribution of stresses towards grain boundaries

Figure 4 summarizes the results from applying above methods to quantify how stresses are distributed in the RVE volume with respect to the distance of each material point to the nearest grain boundary. Using the two developed grain reconstruction methods in combination with the three distancing methods allowed for rigorous and comparative analyses on the same dataset. The results document the statistics at final strain when the stress distributions are broadest and highest in terms of absolute values.

All methods identify with very similar significance and quantile values that the average σ_{33} compressive Cauchy stress is 140 MPa. Only an at most 10 % absolute variation from this average value is observed when probing deeper into the grain interior. Distances larger than approximately 15 distance units from the boundary should not be interpreted because the corresponding numerical support is finite counting limited, i.e. only very few points belong to these distance bins given the average grain radius of 20 voxels.

Comparing the distributions for each distance bin and method combination (Fig. 4a) conveys that all methods capture the flattening of the grains in the compression direction. Consequently, higher counts for the same distance bin are observed for the strain step at yield versus the one at final strain.

The results for the lower distance classes document a rigorous quantification of the numerical effects inherent to spectral based methods in general, and those specific for the DAMASK spectral solver: stresses at material points in the vicinity of numerical discontinuities, here the grain boundaries, show consistently different mean stresses (Figs. 4b/4c) (10 % to 14 %). Finding this difference consistently for material points at the discontinuities suggests that this is attributable to the Gibbs phenomenon [94, 95]. Latter is known as a systematic intensity overshooting in frequency space



(a) Distance distributions

(b) Texture ID based

(c) Louvain based SP

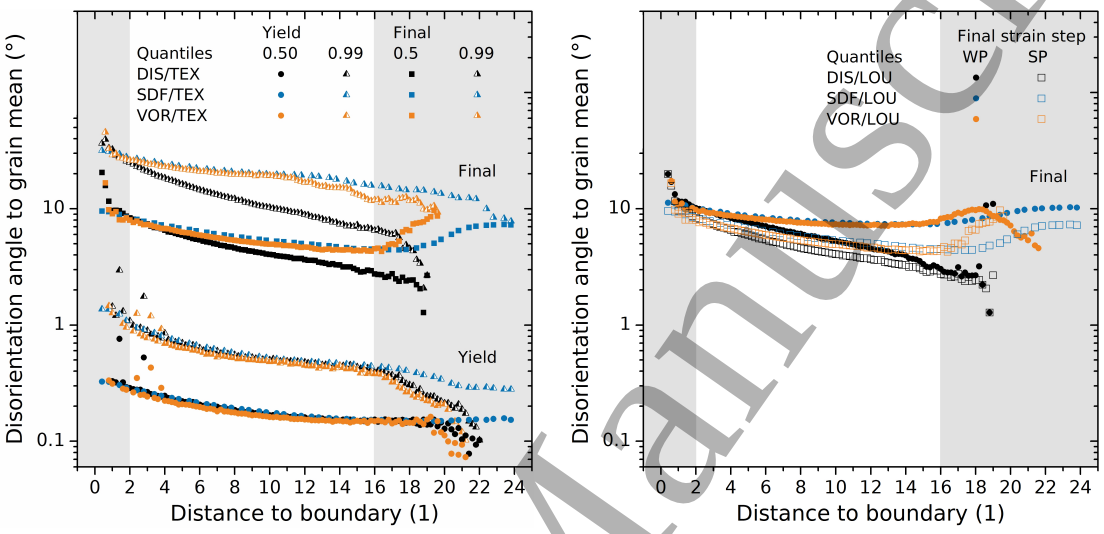
(d) Louvain based WP

Figure 4: Cauchy stress spatial distributions as a function of the distance to the nearest boundary, here exemplified for the largest stress component σ_{33} and comparing specific quantiles of the distance and the state variable distributions. Theory details the results for the analysis of the distances of the voxel center to the Voronoi cell facets in continuous space. Abbreviations discern the distancing methods: disorientation (DIS), signed distance / voxelization (SDF), or tessellation (VOR) based, the grain reconstruction method (texture index TEX or Louvain LOU based), and whether large disorientations within a cluster were strongly (SP, $k = 1000$) or weakly (WP, $k = 75$) penalized. Grayed-out regions detail where numerical effects of the spectral method are expected strongest — either due to numerical effects or finite counting.

when attempting to Fourier expand at discontinuities.

3.3. Quantifying grain orientation spread accumulation towards boundaries

Motivation for this second application exercise comes from frequent literature findings which reported that material volume in the vicinity of grain [96] and phase boundaries [97] is differently disoriented than volume in the grain interior. Possible presence of such localized volume with different orientation has been identified as one reason why static and dynamic recrystallization nuclei [98] originate frequently from the faces or junctions of the grain boundary network [16, 17, 99].



(a) Texture ID based (b) Louvain based WP vs SP

Figure 5: Key quantile values of the distribution of material point disorientation to the mean orientation of the reconstructed grain. Abbreviations discern like in Fig. 4 the distancing methods: disorientation (DIS), signed distance / voxelization (SDF), or tessellation (VOR) based, and how disorientations during grain reconstruction with the Louvain method were penalized (strongly SP or weakly WP). The distribution mean and the upper quantile (0.99) detail the spread of the distributions. In a) distributions are plotted at yield and final strain, while b) pulls focus on the final strain. As in Fig. 4 grayed-out regions detail where either numerical or finite counting effects are strongest.

The results in Fig. 5 deliver quantitative evidence that spatial gradients of disorientation are detectable with all grain reconstruction and distancing methods: material points in the proximity of a grain boundary show on average larger disorientation to the mean orientation of the grain than do points in the grain interior. It can be excluded that these differences are the result of averaging orientations from multiple material points, as it is the case when measuring e.g. kernel average misorientations (KAM) at boundaries using the SEM/EBSD technique. In fact, when quantifying KAM values a kernel with multiple neighboring material points is evaluated. Probing with the kernel would contribute high disorientation values only if insufficiently strong criteria are set with respect to how much disorientation noise is allowed for a

given kernel. In this work, though, no kernel was used. Instead, the disorientation values were computed independently to only one value for each material point - the respective grain mean orientation.

It can also be excluded that the observed gradients are random spatial correlations for the following reasoning: if a grain contains multiple material points in different orientations in the fundamental zone, it is expected that non-directionally correlated point-to-point disorientations are measured if one eventually compares the disorientation for two arbitrarily picked material points. If the orientation variation is strong and the grain small, i.e. the grain has few points support, eventually larger disorientations are measured on average. However, this scatter should also be correlated if there are orientation gradients with a strong component normal to the interface within the grain. Vice versa this scatter should remain practically uncorrelated if such gradients are absent or the grain contains practically uncorrelated small regions with spurious higher disorientation.

Given the strength of the gradient and mean disorientation value, we conclude that the grains are in an incipient stage of fragmentation. They have not yet accumulated localized regions of significant point-to-point disorientation in the high-angle boundary regime (15°), except for a minority population of material points in an at most 2 pixel wide zone at the boundaries. Qualitatively, their higher disorientation is expected as also the absolute Cauchy stress values are slightly higher in this zone (Figs. 4).

There are two additional observations to make with practical significance for characterizing orientation gradients: first, only the methods with normal distancing capability (SDF, VOR) report similar gradient slope ($\approx \frac{4^\circ}{15px}$). Compared to disorientation based distancing, the slope is moderately lower. Second, disorientation gradient characterization with the graph clustering method shows a strong parameter sensitivity (Fig. 5b).

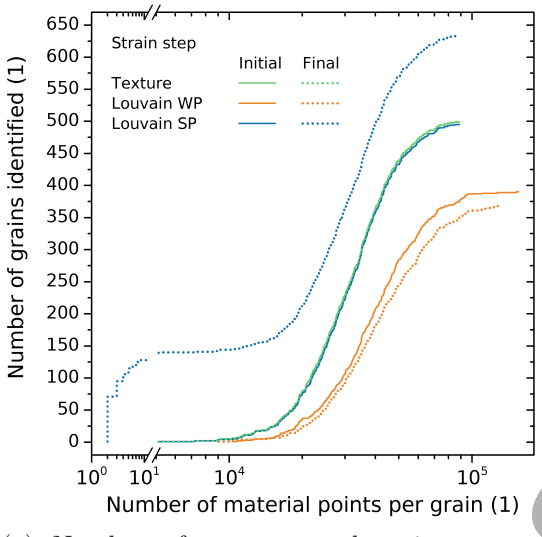
4. Discussion

4.1. Comparing the two grain reconstruction methods

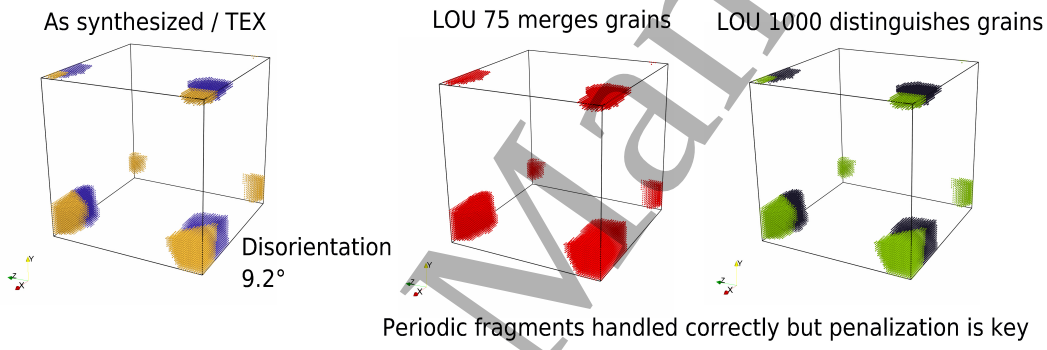
We identified that one practical challenge of using the graph clustering method for grain reconstructing is the strong parameter sensitivity. Therefore, it was worth studying the grain size distributions (Fig. 6a) and consistency with which individual grains were (re-)identifiable (Fig. 6b) as a function of the penalization parameter K_L (algorithm B). Figure 6 summarizes the key findings.

The larger the penalization parameter K_L is set the stronger every accumulation of high disorientation between nodes of the same community gets penalized. In effect, more grains with lower in-grain orientation variation are found on average for a stronger penalization ($K_L = 1000$).

Neither the values for a low nor for a high penalization can avoid systematic challenges when using the reconstructed grain ensemble for characterizing the distance



(a) Number of reconstructed grains versus their size for a weak (WP) versus a strong (SP) penalization



(b) A single grain resolved assessment of the graph clustering method [59] identifies that for a weak penalization $K_L = 75$ preferentially neighbors with low disorientation are merged, thereby reducing the total number of grains.

Figure 6: Texture ID versus graph clustering grain reconstruction: compared in terms of how many grains were reconstructed and how individual grains were (re-)identifiable for different penalization. Initial versus the final strain step results are compared.

correlations. In fact, if the penalization is weak, the algorithm reconstructs different grains and comes not even close to the initially synthesized number. Figure 6b proofs that this is a systematic consequence of the method's tendency to merge neighboring grains with low disorientation at low K_L values.

In effect, the mean orientation of many grain pairs is an average of at least two orientation ensembles with low intra ensemble but possible noticable inter ensemble disorientation. With respect to spatial disorientation gradients this has two consequences: first, such gradients are flatter and second they are shifted to larger disorientation on average. This explains why the gradients for weakly penalized graph

clustering (Fig. 5b) are shallower.

One could avoid the possible bias introduced by the systematic grain merging through applying a stronger penalization, such that eventually the initial grains are re-identified (Fig. 6a). Figure 6a shows the detrimental effect of such procedure when applying it to analyze the (highly) deformed configurations: we observe oversegmentation with an associated qualitative change of the grain size distribution from unimodal to bimodal. A detailed inspection of the respective material points identified that this spreading into eventually bimodality is a consequence of the fact that the strongest disoriented material points at the grain boundary get now segmented into a decoration of the boundaries with very small grains. These findings have a methodological and a practical implication.

The methodological implication is that graph clustering based grain reconstructions should always be backed up by rigorous quantitative parameter sensitivity studies. One strategy could be to report always unnormalized distributions of grain sizes as a function of the segmentation parameter. Otherwise, different physical conclusions might be drawn even though one uses the same method. One example is exemplified in Fig. 5b: disorientation accumulates at boundaries or not.

The practical implication reads as follows: grain fragmentation should be better characterized ideally via the evolution of the grain boundary network rather than to continue insisting logically on the initial grain as the decisive object. In fact, not the homogeneous regions within a deformed grain but the heterogeneities are the key microstructure locations and first descriptors to investigate where and how annealing microstructure evolution initiates.

4.2. Implications of numerics-induced scatter for non-local and coupled crystal plasticity/interface migration spectral method models

Evincing such numerics-induced scatter suggests to apply above quantification protocol in the future regularly. Especially, it should be applied in so-called non-local crystal plasticity models. These predict the local material point values via evaluating a kernel of neighboring material points. Examples are continuum scale crystal plasticity solvers with incrementally and locally coupled dislocation flux sub-models, such as the one from [31, 100] or full-field crystal plasticity models with incremental coupling to phase-field solvers [35].

One should always start inspecting such solvers from an in-depth monitoring of the (state variable) values and the non-local flux terms for each material point as a function of strain. Methods, such as the ones developed in this paper, could be used to assess via such monitoring in how far numerical noise remains uncorrelated and low to avoid a systematic biasing of the results towards larger strains.

Another example in which rigorous quantitative monitoring of state variable values is useful are incrementally coupled grain boundary migration / crystal plasticity models. These typically evolve microstructures along complex transients. As such,

any permeation of correlated numerical noise into the flux or interface migration terms should be reduced as best as possible to keep rigorously controlled conditions, physical accuracy, and precision.

Knowledge in the discrete dislocation dynamics community teaches us that uncontrolled stress spikes should better remain numerically controlled, for instance by supplementing the integration with some time trajectory before making choices [101, 102]. For these future challenges the work delivers an approach based on which to build further tools for assessing the numerical and physical quality of such predictions.

4.3. Comparing computational efficiency

Strong scaling multithreading performance Figure 7 summarizes wall clock timings to document the tools' strong scaling efficient multithreading capabilities. Furthermore, the figure documents key computing time contributions for all combinations of above post-processing methods applied to the same data, using the same computer, and exclusive execution. The left column of each method column pair shows the sequential execution time. The results identify that SDF, in combination with either of the two grain reconstruction methods, is the least costly combination. Results for a single strain step and sequential execution are available after 2 h approximately. Given the capability to identify normal distances is another argument to choose the SDF method over disorientation based distancing.

Figure 7 documents that all methods concentrate at least 87.5 % to 99.3 % of the total execution time in at most three algorithmic code sections. For the disorientation based distancing these steps are the querying of neighboring points and computing point-to-point disorientation angles. For the signed distancing method it is the identification of periodic images, volume rediscretization, and the remapping of every voxel to a closest unique material point. For tessellation based distancing the evaluation of the projected normal distances for each material point demands contour facet querying.

In this work, all these most costly steps are parallelizable. Thus, specific care was taken to implement them in parallel. In addition, data placement was orchestrated such that GB-sized data chunks were placed close in the memory of the executing core taking into account the memory hierarchy of the cluster computer.

In effect, this speeds up the execution of every method combination by 13.6 to 24.4 times the sequential baseline. With 40 OpenMP threads required to achieve this, a strong scaling efficiency between 61 % to 34 % is documented. The main obstacle to achieve higher efficiency in this study was load imbalance. Such was strongest for the tessellation (VOR) method because grains were distributed to the threads statically in round-robin fashion. When building tessellations, however, the grain size critically affects how many facets the contour hull contains; and thus how dissimilar the distancing costs are for different grains. In light of this, the distributing of only 500 grains across 40 threads will equip each thread with possibly too few grains to compensate for the large variety of grain sizes. In effect, the cost and facet count in the contour hull processing

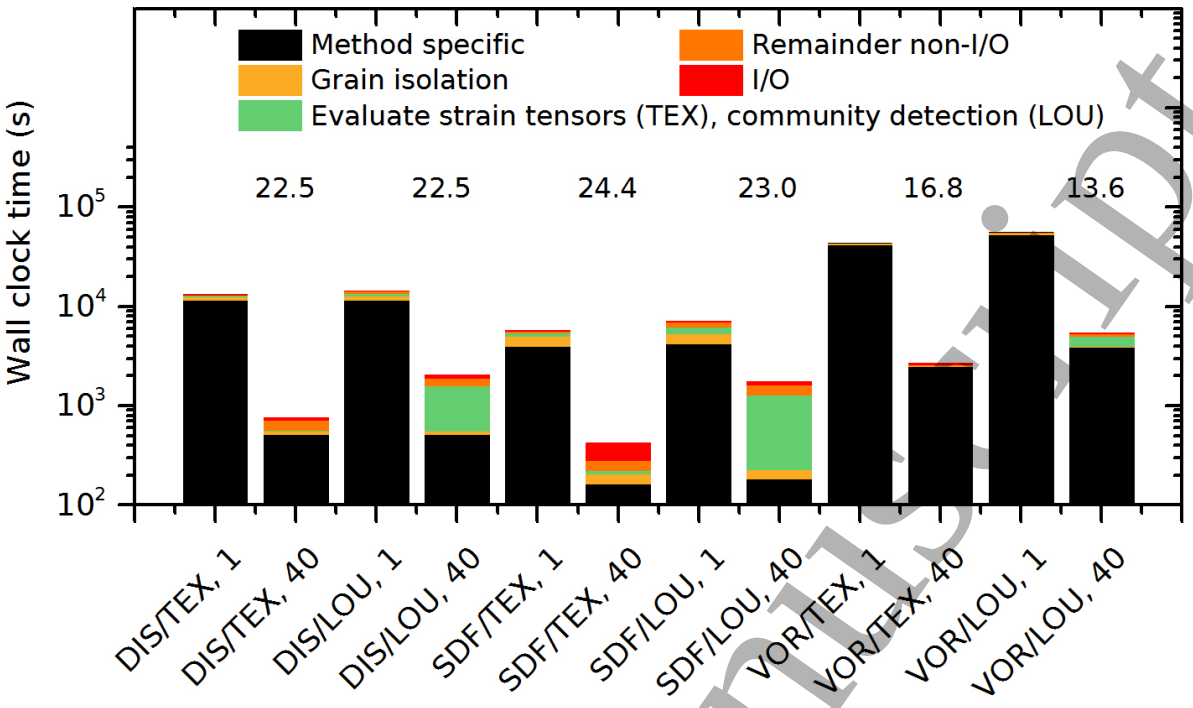


Figure 7: Key results from benchmarking all three methods in combination with both grain reconstruction protocols using the same strain step ($\overline{\epsilon_{vM}} = 0.21$) and the same cluster computer. The numbers above the right column of each column pair denote the respective speedup achieved when executing in parallel for the most costly method specific processing step. Abbreviations identify the disorientation based (DIS), the signed distance/voxelization based (SDF), and the tessellation based distancing (VOR). Grain reconstruction was based either on the initial assignment (i.e. texture ID based TEX) or graph clustering (LOU). Sequential execution is compared with multithreaded execution. Analyses using the LOU method have a different sequential bottleneck than when using TEX (indicated e.g. for DIS/LOU, 40 by the green column portion): the, in this work not parallelized, community detection stage.

stage differs.

A possible improvement for the future is to change the implementation and employ dynamic work scheduling via for instance OpenMP tasking. This, though, demands a compromise: the current round-robin distributing procedure allows to place the grains in controlled memory locations [84]. Dynamic scheduling, in turn, will likely keep the cores more frequently busy but at the cost of typically more memory traffic.

Another potential improvement of this work, with immediate practical benefit, is to parallelize also the graph clustering algorithm for which CPU- and GPU-parallelized community detection solvers have been developed in the past [103–106].

The results document additional practical improvements in which the present work advances the field: the first is improved speed, when compared to hitherto

reported values for DAMASK post-processing [38]. This was achieved with sequential optimization of the file accessing strategy surplus employing parallelized I/O to cut I/O costs by at least an order of magnitude.

As an additional strategy to improve sequential performance, state of the art linear algebra libraries were used to compute point-wise tensor quantities. Such improvement is of practical use not only for executing numerics within DAMASK when solving an RVE but also for post-processing flow curves. As the second improvement, this work details how to combine these strategies with multithreaded execution surplus trivial data parallel processing of strain steps on multiple nodes of a cluster computer.

Having the possibility to construct contour hulls to each grain in the deformed configuration is the third improvement. On the one hand because it allows to quantify a volume for each grain not only by counting the number of material points it contains but by accumulating the volume of its supporting Voronoi cells. The resulting polygon mesh allows for volumetric rendering of the grain as a polyhedron, as exemplified in Fig. 1b. These may be useful to allow further analyses using other microstructure characterization tools like DREAM3D [53] or QUBE [52].

Hybrid execution performance During hybrid execution every MPI process evaluates a different strain step and spawns an own group of threads. Data are read independently from a priori known sections of the DAMASK results file. In effect, this enabled to evaluate all 28 strain steps at once using 1120 cores. Figure 8 summarizes the key results for this hybrid execution mode. Specifically, two sets of common analysis cases were probed: a flow curve extraction versus analyses of aforementioned state variable values to boundary distance correlations, six of them in total.

There are two key findings of practical importance. First, hybrid processing enables to cut wall clock time by an additional order of magnitude. In effect, the evaluation of a flow curve, which took damaskpdt 3.2 h sequentially, was solvable in 76 s when using parallelism. Similarly, distancing results for all strain steps were analyzed after 1.5 h when using tessellation (VOR), the most costly method. Distances were even faster accessible, namely in 8 min, using the signed distance method.

The practical benefit of these findings is put in perspective by the following observation. While the SDF method was already finished for the entire ensemble, the classical DAMASK post-processing routines had not even completed the processing of the flow curve for the first strain step.

The second key finding is that parallel performance gains are limited. Respective arguments for the multithreaded analyses were already identified (Fig. 7). Not parallelizing all post-processing computational steps is another apparent reason for limited scalability [107]. The most significant obstacle to cut execution time further, though, are microstructure-induced differences. Specifically, differences which are caused by interface networks with dissimilar shape and distribution of facet area across the strain step ensemble. Consequently, the individual algorithms show different solving time resulting in workload differences across the strain step ensemble.

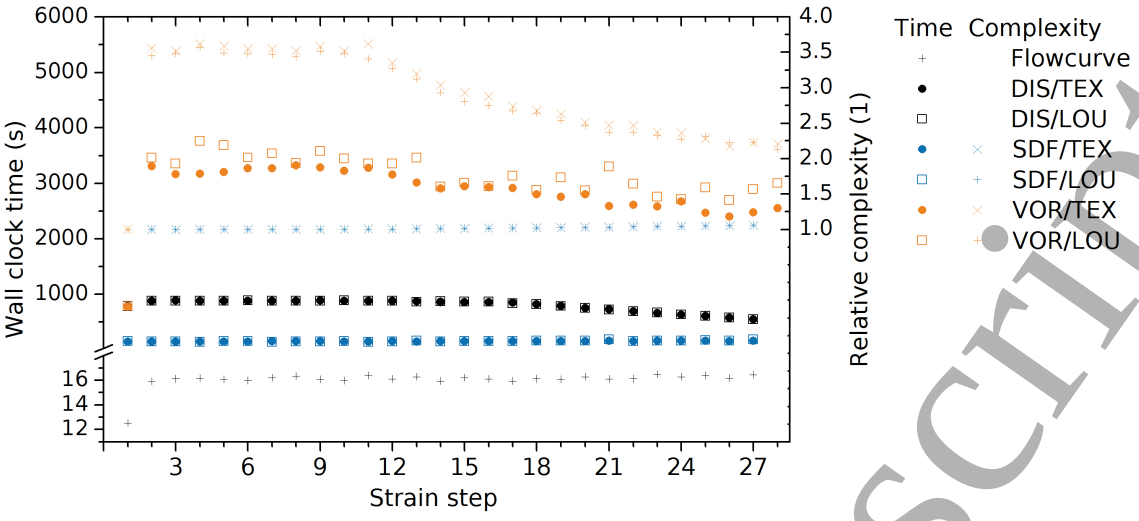


Figure 8: Key results from benchmarking all three methods in combination with both grain reconstruction protocols on the entire strain step ensemble. Different computing costs are quantified via the wall clock time. The results identify that the amount of computational work is different across the strain step ensemble. Taking the first strain step as the reference, relative complexities were established through monitoring e.g. how many voxel were processed for the ensemble of grain bounding boxes and how many Voronoi facet polygon inclusion and edge comparison tests were required to find the closest distances.

Two observations are important to add here. First, these inter strain step work load differences would remain even if the multithreaded execution of each strain step gets better work load partitioned. Second, it is in fact the successive reduction of the relative numerical complexity for a strain step with increasing strain which limits primarily the hybrid performance of MPI parallelization. Provided there were more strain steps to process very likely even more processes could be employed. This is one of the key advantages to our true scientific computing performance solution for post-processing DAMASK simulations.

One example is visible for signed distance based distancing. Given that the RVE shape change results in a moderate increase in the total bounding box volume, more voxel have to be probed for the higher than the lower strain steps. Another, by far stronger, example for microstructure induced work load imbalances is visible for tessellation based distancing (Fig. 8).

When using the strongly penalized grain reconstruction, more grains are detected at higher strains. Therefore, also more boundary area is generated on average, as a consequence of which the material points lay closer to a boundary, providing for faster distancing operations. In addition also different contour hull shape and facet compositions are generated. This modifies the composition of the individual bounded volume hierarchies, and thus their individual response and pruning effectiveness during

querying them.

It is useful to compare the computational costs of post-processing to the actual simulation costs. The DAMASK deformation simulation occupied 36 cores for 354 h wall clock time, i.e. 12744 core hours were spent in total. The post-processing of the entire strain step ensemble using the most costly distancing method kept 1120 cores busy for 4822 s wall clock time, i.e. 1500 core hours in total, or at most 11.8 % of the simulation costs.

Lastly, the memory consumption of the simulation and post-processing should be reported. DAMASK allocated a total of 230 GB virtual memory, i.e. 14.3 kB per material point. Post-processing demands less virtual memory per process when evaluating the flow curve only (21 GB). However, virtual memory consumption peaks at 134 GB for the most costly strain step and using the VOR/LOU distancing method. One reason for the larger memory consumption are the storage demands for position values for a portion of the 26 periodic images to all material points. Another reason is that Voronoi polyhedra and their facets have to be stored for tessellation based grain reconstruction.

Another contribution is due to the conservative strategy we used to implement the querying structures. This could be optimized in the future with potential for a memory footprint reduction of at least a factor two.

5. Conclusions

A set of strong and weak scaling post-processing methods were developed to quantify the accumulation of state variable values at grain boundaries in 3D full-field spectral method crystal plasticity simulations. Exemplified for the DAMASK spectral solver and a phenomenological constitutive crystal plasticity model the results substantiate:

- Voxel accurate methods were developed which reconstruct successfully the grain boundaries and operate on arbitrarily deformed and periodically confined RVE domains with respect to the deformed configuration. This allows for direct comparison to experiments and rigorous numerical assessment of the spectral method.
- Long range gradients of material point disorientation, with higher values to the boundary than in the grain interior, were quantified. They are a signature of incipient grain fragmentation.
- As an alternative method to capture the individual fragments, also a graph clustering grain reconstruction method was assessed. Compared to the classical method of assigning grain IDs based on the initial conditions, though, the quantification of the gradients is very parameter dependent.
- Three methods were detailed to quantify the distance of a material point to the boundary.

- All performance critical computational tasks were implemented distributed and shared memory parallel. Therewith, processing the entire set of 28 strain steps from a 256^3 RVE cube DAMASK simulation, worth 162 GB, was possible in 470 s wall clock time using 28 processes each spawning 40 threads. Compared to exclusive multithreaded (11 063 s) or sequential execution (150 000 s), this hybrid approach allowed for 320 times faster post-processing. The efficiency could be improved further if heterogeneous work load across the individual strain step data, which is method dependent, gets better balanced and remaining sequential code portions parallelized.

Data and code availability

The source code of the tool and its supplementary MATLAB scripts with which we post-processed all results are open source software. They are publicly accessible [108]. All DAMASK and post-processing parameter settings files, including the SLURM batch system submission scripts are available for download [109]. The entire repository of compressed post-processed data occupies 844 GB. It is available from the authors upon serious request.

Acknowledgements

The authors gratefully acknowledge the funding received from the German Research Foundation through project RO 2342/8-1. The work was partially supported by BiGmax, the Max Planck Society's Research Network on Big-Data-Driven Materials-Science. MK acknowledges the discussions with Karo Sedighiani, Muhammad Imran, and Markus Bambach on parameterizing constitutive models. Discussions with Matthew Kasemer on implementing the stress conventions and Martin Diehl on DAMASK and its post-processing scripts are acknowledged.

Work distribution

MK designed and implemented the tool surplus performed and post-processed the simulations. MK and FR wrote the manuscript and discussed the results on a regular basis.

Appendices

A. Detailed specification of the post-processing workflow

Algorithmic details In what follows, the individual steps of the post-processing workflow are detailed. For all steps with a larger than linear time complexity $\mathcal{O}(N_x N_y N_z := N)$, additional details are commented. In every step, N denotes the total number of unique material points supporting the RVE. Indices n specify the strain

step. Three point cloud sets are distinguished: the first contains the N unique material points of the original domain $\mathcal{P}_n^0 \in \mathcal{R}^3$. The second, $\mathcal{P}_n^{0+\epsilon}$, contains $\mathcal{P}_n^0$ surplus all periodic images of $\mathcal{P}_n^0$ which lay inside a cuboidal bounding box about a cuboidal tight bounding box to $\mathcal{P}_n^0$. The outer bounding box is fattened by $\epsilon = 0.1$ in length units of the initial RVE domain. The third point cloud set, $\mathcal{P}_n^1$ contains N_n^1 points, i.e. all members of $\mathcal{P}_n^0$ surplus all their 26 periodic copies. The members i of the material point ensemble $\mathcal{P}_n^0$ have positions $p_{i,n}^0$ which are defined by their initial locations in the RVE in the reference configuration [38, 44] $x_{i,n}^0$ surplus a deformation-induced displacement which maps into the deformed configuration in the laboratory coordinate system $\Delta x_{i,n}^0$. Every material point has associated state variable values $s_{i,n}$. In addition, a reference ID and periodic image variant ID is stored to track which periodic image points are assigned to which unique material point of $\mathcal{P}_n^0$. This allows dereferencing of state variable values $s_{i,n}$ rather than duplicating them. IMKL specifies in which steps the Intel Math Kernel Library is used.

Algorithm 1 Post-processing workflow

procedure DAMASKPDT
  I/O parse parameter and initialize MPI
  I/O parse spectralOut file layout to identify content
  Partition N_n strain increments round robin on MPI processes
  for $\epsilon_n \leftarrow 1, N_n$ **do**
    Process parallelized MPI I/O spectralOut file reading
      ▷ State variable values $s_{i,n}$ per point $\boldsymbol{x}, \boldsymbol{F_T}, \boldsymbol{F_P}, \boldsymbol{P}, \boldsymbol{q}$
    Spatial distributing of material point cloud to OpenMP threads
    Threaded stresses and strains $\forall i \in \mathcal{P}_n^0$ via IMKL
    RVE averaging $\overline{\boldsymbol{F}}, \overline{\boldsymbol{P}}, \overline{\sigma_{vM}}, \overline{\epsilon_{vM}}$
    Threaded displacements $\Delta \boldsymbol{x_{i,n}^0}$ via IMKL ▷ $\mathcal{O}(Nlog(N))$
    Compute periodic images $\mathcal{P}_n^{0+\epsilon}$
    Build a spatial index $\mathcal{B}_n^{0+\epsilon}$ that partitions $\mathcal{P}_n^{0+\epsilon}$
    Threaded grain reconstruction
    if Initial assignment / texture ID based **then**
      RECONSTRUCT_GRAINS_TEXTUREID (see B)
    if Community detection based **then**
      RECONSTRUCT_GRAINS_LOUVAIN (see B)
    GRAIN_GEOMETRY_SIMPLIFICATION (see B)
    Threaded state variable/distance to boundary quantification
    if Disorientation thresholding based scalar distances **then**
      ANALYZE_DIS (see C)
    if Signed distance / voxelization based normal distances **then**
      ANALYZE_SDF (see C)
    if Tessellation based contour hull normal distances **then**
      ANALYZE_VOR (see D)
    MPI parallel I/O binary results $(d, \boldsymbol{\sigma}, \boldsymbol{q})_{i,n} \forall$ voxel or i
    MPI aggregate flow curve data $(\epsilon_{vM}, \sigma_{vM})_n$
  I/O flow curve, internal profiling results, finalize MPI and exit

B. Detailed specification of the grain reconstruction methods

Texture ID based grain reconstruction $\mathcal{T}_i$ specifies the initial texture ID assigned during microstructure synthesis DAMASK [38, 44].

Algorithm 2 Texture based grain reconstruction

```
procedure RECONSTRUCT_GRAINS_TEXTUREID
    for i ← 1, N do
        Assign point i a grain ID G_{i,n} := T_i
```

Graph clustering based grain reconstruction N_{mv} specifies the number of executed node relabeling operations in the current iteration step, Q the Newman-Girvan modularity [58] using a critical quality value of $Q_c = 0.01$. K_L specifies the penalization strength, i.e. whether a weak $K_L = 75$ or a strong $K_L = 1000$ penalization was used. $\|[q_{i,j,n}]_0\|$ denotes the rotation angle argument of the disorientation quaternion. The search radius during graph edge construction was chosen as $R_c^{lv} = \frac{2}{256}$ distance units.

Algorithm 3 Louvain community based grain reconstruction

```
procedure RECONSTRUCT_GRAINS_LOUVAIN
    for i ← 1, N do
        Find all neighboring points p_{j,n} ∈ P_n^{0+ε} | {‖x_{i,n} − x_{j,n}‖ ≤ R_c^{lv}}
        Compute disorientation angle [q_{i,j,n}]_0
        Compute edge weights w_{i,j,n} = exp(K_L · (‖[q_{i,j,n}]_0‖ − 1))
        Add one graph edge with weight w_{i,j,n} for every position pair p_{i,n}, p_{j,n}
    do
        Iterative graph clustering according to [59–61, 110]
    while N_mv > 0 and Q > Q_c
    for i ← 1, N do
        Assign i the final label of the top-level community as grain ID
```

Grain geometry simplification $N_{k,n}^0$ and $N_{k,n}^1$, respectively are the number of material points supporting the k-th grain in $\mathcal{P}_n^0$ and $\mathcal{P}_n^1$ respectively. $A_{k,n}$ specifies a tight axis-aligned bounding box about the positions $N_{k,n}^1$. $\mathcal{A}_n$ is the tight cuboidal global bounding box which contains all $A_{k,n}$. In this work a cell size of half the normalized initial material point point-to-point distance was used, i.e. $d_{cell} := \frac{1}{2 \cdot 256}$.

Algorithm 4 Threaded extraction of a single periodic image per grain

procedure GRAIN_GEOMETRY_SIMPLIFICATION
  Re-organize members of $\mathcal{P}_n^0$ into disjoint sub-sets $\mathcal{P}_{k,n}^0$ per grain k ▷ In what follows, multithreaded processing $\forall k$ grains
  Threaded memory initialization
  Threaded periodic images $\mathcal{P}_{k,n}^1$ and bounding boxes $\mathcal{A}_k$
  Threaded building of sparse spatial index $\mathcal{B}_{k,n}^1$ via $\mathcal{B}_n^1$
  Threaded merging of possible grain fragments via DBScan [62] ▷ $\mathcal{O}(N_k log(N_k))$
  Threaded picking of one representative per grain $\hat{\mathcal{A}}_{k,n}$
  Identify global bounding box $\mathcal{A}_n$ enclosing all $\hat{\mathcal{A}}_{k,n}$
  Define a global voxelization $\mathcal{L}_n$ of volume $\mathcal{A}_n$ with cell size d_{cell}
  Fuse members of $\mathcal{B}_{k,n}^1$ into one global spatial index $\mathcal{B}_n^1$

C. Detailed specification of the distancing methods

Disorientation based distancing A disorientation angle of $\Theta_c = 15°$ was used for thresholding.

Algorithm 5 Disorientation thresholding based scalar distances

procedure ANALYZE_DIS
  for $i \leftarrow 1, N$ **do** ▷ Multithreaded, dynamic scheduling
    Find all N_j neighbors $p_j^{0+\epsilon} \in \mathcal{P}_n^{0+\epsilon} \mid \{\|\boldsymbol{x_{i,n}} - \boldsymbol{x_{j,n}}\| \leq R_c^{lv}\}$
    Sort by distances $d_{i,j,n} := \|\boldsymbol{x_{i,n}} - \boldsymbol{x_{j,n}}\|$ ▷ $\mathcal{O}(N_j log(N_j))$
    for $j \leftarrow 1, N_j$ **do**
      Compute disorientation angle $\Theta_{i,j,n}$
      if $\Theta_{i,j,n} \geq \Theta_c$ **then**
        Report $(d_{i,j,n}, \boldsymbol{s_{i,n}})$
        break
                ▷ ignore i if no pair found

Signed distance / voxelization based normal distances Abbreviations denote a signed distance function (SDF) and the fast sweeping method (FSM) [65, 69].

Algorithm 6 Signed distance / voxelization based normal distances

procedure ANALYZE_SDF
  Threaded voxelizing of k according to $\mathcal{L}_n$, d_{cell} using $\mathcal{B}_n^1$
  Specifically, $\forall$ voxel $\in \mathcal{B}_{k,n}^1$ identify closest member of $\mathcal{P}_n^0$
  Threaded initialization of SDF $\Phi_{k,n}(x)_0$
  Threaded spreading of SDF via FSM $\Phi_{k,n}(x)_1$ as in [65]
  Threaded report $(\Phi_{k,n}(x), \boldsymbol{s_{j,n}})$ $\forall$ voxel with $\Phi_{k,n}(x) \geq 0$

D. Tessellation based contour hull normal distances

$\hat{\mathcal{A}}_{k,n}$ specifies an axis-aligned bounding box about the identified single periodic image of grain k. The box is fattened by a guard zone of thickness $\frac{3}{256}$ of the normalized initial material point-to-point distance. The fattening assures that the volume about each material point belonging to the grain k can be tessellated and no Voronoi cell gets arbitrarily cut off by the bounding box walls. The query structure of the Voro++ library was configured to include five points per spatial bin on average. For each grain k interior points are distinguished from exterior points with respect to their logical identity when building the exterior contour hull of the grain. Specifically, interior points ip originate from a material point in $\mathcal{P}_n^0$ with grain ID k. BVH is short for bounded volume hierarchy. N_f^k denotes the number of exterior facets identified for each grain k. N_{ip}^k is the number of interior points per grain k. Exterior facets to a Voronoi cell of an interior point are all facets to the first order neighboring cells of an exterior point if any such exist.

Algorithm 7 Tessellation based contour hull normal distances

procedure ANALYZE_VOR ▷ $\forall k$
  Threaded pulling of all points $\hat{\mathcal{A}}_{k,n}$ from $\mathcal{B}_n^1$
  Threaded mapping periodic image point to unique material point
  Threaded Voro++ build spatial index to accelerate tessellator
  Threaded Voro++ tessellate all Voronoi cells for points
  Threaded Voro++ exterior facets, consistent outer unit normal
  procedure IDENTIFY_NORMAL_DISTANCES
    Threaded construction of a BVH with all facet polygons
    for $ip \leftarrow 1$, N_{ip}^k **do**
      Pool all nf facets f of the contour as initial candidates
      while $f < nf$ **do**
        Normal projection interior point x_{in} on facet plane x_{proj}^f
        if x_{proj} covered by facet polygon = false **then**
          for $e \leftarrow 1$, N_e^f **do** ▷ Circulate facet edges e of f
            Normal projection x_{in} on e
            if x_{proj}^e on the edge = true **then**
              Eventually update current shortest distance
              Eventually re-query facet candidate pool
          for $v \leftarrow 1$, N_v^f **do** ▷ Circulate facet vertices v of f
            Compute distance $d_{in,v} = \|x_{in} - x_v\|$
            Eventually update current shortest distance
            Eventually re-query facet candidate pool
        else
          Eventually update current shortest distance
          Eventually re-query facet candidate pool
  Threaded report value pairs $(d_{ip,n}, \boldsymbol{s_{ip,n}})$ $\forall ip$

References

[1] Kocks U F, Argon A S and Ashby M F 1975 *Thermodynamics and kinetics of slip* 1st ed (Pergamon Press, Oxford) ISBN 9780080179643

[2] Hosford W F 2010 *Solid Mechanics* 1st ed (Cambridge University Press, New York) ISBN 978-0-521-19229-3

[3] Humphreys J, Rohrer G S and Rollett A 2017 3rd ed (Elsevier) ISBN 978-0-08-098235-9 recrystallization and Related Annealing Phenomena

[4] Eshelby J D, Frank F C and Nabarro F R N 1951 *Philosophical Magazine* 35–1364

[5] Hall E O 1951 *Proceedings of the Physical Society B* **64** 495–502

[6] Mughrabi H 1983 *Acta Metallurgica* **31** 1367–1379

[7] Hirth J P and Lothe J 1992 *Theory of dislocations* 2nd ed (Krieger Publishing Company, Malabar, Florida Dental Journal) ISBN 0-89464-617-6

[8] Bayerschen E, McBride A T, Reddy B D and Böhlke T 2016 *Journal of Materials Science* **51** 2243–2258

[9] Beaudoin A J, Mathur K K, Dawson P R and Johnson G C 1993 *International Journal of Plasticity* **9** 833–860

[10] Beaudoin A J, Mecking H and Kocks U F 1996 *Philosophical Magazine A* **73** 1503–1517

[11] Raabe D, Zhao Z, Park S J and Roters F 2002 *Acta Materialia* **50** 421–440

[12] Quey R, Dawson P R and Driver J H 2012 *Journal of the Mechanics and Physics of Solids* **60**(3) 509–524

[13] Dillamore I L, Katoh H and Haslam K 1974 *Texture* **1** 151–156

[14] Sakai T and Jonas J J 1984 *Acta Metallurgica* **32** 189–209

[15] Wusatowska-Sarnek A M, Miura H and Sakai T 2002 *Materials Science and Engineering A* **323** 177–186

[16] Miura H, Sakai T, Hamaji H and Jonas J J 2004 *Scripta Materialia* **50** 65–69

[17] Miura H, Sakai T, Mogawa R and Gottstein G 2004 *Scripta Materialia*

[18] Miller M P and Dawson P R 2014 *Current Opinion in Solid State and Materials Science* **18** 286–299

[19] Sun S, Adams B L and King W E 2000 *Philosophical Magazine A* **80** 9–25

[20] Mishra S K, Pant P, Narasimhan K, Rollett A D and Samajdar I 2009 *Scripta Materialia* **61** 273–276

[21] Calcagnotto M, Ponge D, Demir E and Raabe D 2010 *Materials Science and Engineering A* **527** 2738–2746

[22] Di Gioacchino F and Da Quinta J 2013 *Experimental Mechanics* **53** 743–754

[23] Wright S I, Nowell M M, de Kloe R and Chan L 2014 *Microscopy & Microanalysis* **20** 852–863

[24] Subedi S, Pokharel R and Rollett A D 2015 *Materials Science and Engineering A* **638** 348–356

[25] Pokharel R, Lind J, Kanjarla A K, Lebensohn R A, Li S F, Kenesei P, Suter R M and Rollett A D 2014 *Annual Review of Condensed Matter Physics* **5** 317–346

[26] Pokharel R, Lind J, Li S F, Kenesei P, Lebensohn R A, Suter R M and Rollett A D 2015 *International Journal of Plasticity* **67** 217–234

[27] Hill R 1963 **11**(5) 357–372

[28] Drugan W J and Willis J R 1996 **44** 497–524

[29] Kanit T, Forest S, Galliet I, Mounoury V and Jeulin D 2003 *International Journal of Solids and Structures* **40** 3647–3679

[30] Kords C 2013 *On the role of dislocation transport in the constitutive description of crystal plasticity* (epubli GmbH)

[31] Reuber C, Eisenlohr P, Roters F and Raabe D 2014 *Acta Materialia* **71** 333–348

[32] Wong S L, Obstalecki M, Miller M P and Dawson P R 2015 *Journal of the Mechanics and Physics of Solids* **79** 157–185

[33] Ozturk T, Stein C, Pokharel R, Hefferan C, Tucker H, Jha S, John R, Lebensohn R A, Kenesei P,

Suter R M and Rollett A D 2016 *Modelling and Simulation in Materials Science and Engineering* **24**

[34] Tari V, Lebensohn R A, Pokharel R, Turneer T J, Shade P A, Bernier J V and Rollett A D 2018 *Acta Materialia* **154** 273–283

[35] Zhao P, Low T S E, Wang Y and Niezgoda S R 2016 *International Journal of Plasticity* **80** 38–55

[36] Maire L, Scholtes B, Moussa C, Bozzolo N, Muñoz D P, Settefrati A and Bernacki M 2017 *Materials & Design* **133** 498–519

[37] Tutcuoglu A D, Vidyasagar A, Bhattacharya K and Kochmann D M *Journal of the Mechanics and Physics of Solids* **122**

[38] Roters F, Diehl M, Shanthraj P, Eisenlohr P, Reuber C, Wong S L, Maiti T, Ebrahimi A, Hochrainer T, Fabritius H O, Nikolov S, Friák M, Fujita N, Grilli N, Janssens K G F, Jia N, Kok P J J, Ma D, Meier F, Werner E, Stricker M, Weygand D and Raabe D 2019 *Computational Materials Science* **158** 420–478

[39] Moulinec H and Suquet P 1994 *Comptes Rendus De L Academie des Sciences Serie II* **318** 1417–1423

[40] Moulinec H and Suquet P 1998 *Computer Methods in Applied Mechanics and Engineering* **157** 69–94

[41] Lebensohn R A 2001 *Acta Materialia* **49** 2723–2737

[42] Lebensohn R A 2012 *International Journal of Crystal Plasticity* **32-33** 59–69

[43] Eisenlohr P, Diehl M, Lebensohn R A and Roters F 2013 *International Journal of Plasticity* **46** 37–53

[44] Diehl M 2016 *High-Resolution Crystal Plasticity Simulations* (Apprimus Verlag, Aachen)

[45] Diehl M, Eisenlohr P, Zhang C, Nastola J, Shanthraj P and Roters F 2017 *Integrating Materials and Manufacturing Innovation* **6** 83–91

[46] Schwartz A J, Kumar M, Adams B L and Field D P (eds) 2009 *Electron Backscatter Diffraction in Materials Science* 2nd ed (Springer)

[47] Dillard S E, Bingert J F, Thoma D and Hamann B 2007 *IEEE Transactions on Visualization and Computer Graphics* **13** 1528–1535

[48] Rowenhorst D J, Lewis D C and Spanos G 2010 *Acta Materialia* **58** 5511–5519

[49] Konijnenberg P J, Zaefferer S, Lee S B, Rollett A D, Rohrer G S and Raabe D 2012 *Materials Science Forum* **702-703** 475–478

[50] McKenna I M, Poulsen S O, Lauridsen E M, Ludwig W and Voorhees P W 2014 *Acta Materialia* **78** 125–134

[51] Ullah A, Liu G, Luan J, Li W, ur Rahman M and Ali M 2014 *Materials Characterization* **91** 65–75

[52] Konijnenberg P J, Khorashadizadeh A, Zaefferer S and Raabe D 2013 *Microscopy and Microanalysis* **19** 846–847

[53] Groeber M A and Jackson M A 2014 *Integrating Materials and Manufacturing Innovation* **3**

[54] Bachmann F, Hielscher R and Schaeben H 2011 *Ultramicroscopy* **111** 1720–1733

[55] Roters F, Eisenlohr P, Kords C, Tjahjanto D D, Diehl M and Raabe D 2012 DAMASK: the Düsseldorf Advanced MAterial Simulation Kit for studying crystal plasticity using an FE based or a spectral numerical solver *IUTAM SYMPOSIUM ON LINKING SCALES IN COMPUTATIONS: FROM MICROSTRUCTURE TO MACRO-SCALE PROPERTIES* vol 3 pp 3–10

[56] McMahon C, Soe B, Loeb A, Vemulkar A, Ferry M and Bassman L 2013 *Ultramicroscopy* **133** 16–25

[57] Loeb A, Ferry M and Bassman L 2016 *Ultramicroscopy* **161** 83–89

[58] Newman M E J and Girvan M 2004 *Physical Review* **69**

[59] Dancette S, Browet A, Martin G, Willemet M and Delannay L 2016 *Modelling and Simulation in Materials Science and Engineering* **24**

[60] Blondel V D, Guillaume J L, Lambiotte R and Lefebvre E 2008 *Journal of Statistical Mechanics:*

Theory and Experiment

[61] Blondel V, Guillaume J L, Lambiotte R and Lefebvre E 2015 Louvain method for community detection in large graphs URL https://sourceforge.net/projects/louvain/files/

[62] Ester M, Kriegel H P, Sander J and Xu X 1996 A density-based algorithm for discovering clusters in large spatial databases with noise *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*

[63] Mießen C, Liesenjohann M, Barrales-Mora L A, Shvindlerman L S and Gottstein G 2015 *Acta Materialia* **99** 39–48

[64] Mießen C, Velinov N, Gottstein G and Barrales-Mora L A 2017 *Modelling and Simulation in Materials Science and Engineering* **25**

[65] Mießen C 2017 *A massive parallel simulation approach to 2D and 3D grain growth* Ph.D. thesis RWTH Aachen University, Aachen, Germany

[66] Osher S and Sethian J A 1988 *Journal of Computational Physics* **79** 12–49

[67] Elsey M, Esedoğlu S and Smereka P 2013 *Acta Materialia* **61** 2033–2043

[68] Scholtes B, Shakoor M, Bozzolo N, Bouchard P O, Settefrati A and Bernacki M 2015 *Key Engineering Materials* **651-653** 617–623

[69] Zhao H 2004 *Mathematics of Computation* **74** 603–627

[70] Rycroft C 2009 *Chaos* **19**

[71] Peterka T, Morozov D and Phillips C 2014 *INTERNATIONAL CONFERENCE FOR HIGH PERFORMANCE COMPUTING, NETWORKING, STORAGE AND ANALYSIS* 997–1007

[72] Okabe A, Boots B, Sugihara K and Chiu S N 2000 *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams* 2nd ed (John Wiley & Sons, Chichester) ISBN 978-0-471-98635-5 spatial Tessellations: Concepts and Applications of Voronoi Diagrams

[73] Bentley J L 1975 *Communications of the ACM* **18**(9) 509–517

[74] Brinkhoff T, Kriegel H P and Seeger B 1996 Parallel processing of spatial joins using r-trees *Proceedings of the 12^{t}h International Conference on Data Engineering, February 26th - March, 1st, 1996, New Orleans, LA* pp 258–265

[75] Balasubramanian L and Sugumaran M 2012 *International Journal of Computer Applications* **42** 35–41

[76] Hedges L 2017 AABB.cc — Dynamic AABB trees in C++ with support for periodic systems URL https://github.com/lohedges/aabbcc

[77] Hielscher R and Schaeben H 2008 *Journal of Applied Crystallography* **41** 1024–1037

[78] Mainprice D, Bachmann F, Hielscher R and Schaeben H 2014 *Rock Deformation from Field, Experiments and Theory: A Volume in Honour of Ernie Rutter* chap Descriptive tools for the analysis of texture projects with large datasets using MTEX: strength, symmetry and components

[79] Bachmann F, Hielscher R, Jupp P E, Pantleon W, Schaeben H and Wegert E 2010 *Journal of Applied Crystallography* **43** 1338–1355

[80] Gropp W, Lusk E and Skjellum A 1999 *Using MPI: Portable Parallel Programming with the Message Passing Interface* 2nd ed (MIT Press) ISBN 978-0262571326 using MPI: Portable Parallel Programming with the Message Passing Interface

[81] Gropp W, Lusk E and Skjellum A 1999 *Using MPI-2 Advanced Features of the Message Passing Interface* 1st ed (MIT Press) ISBN 978-0262571333 using MPI-2 Advanced Features of the Message Passing Interface

[82] Chapman B, Yost G and van der Paas R 2007 *Using OpenMP: portable shared memory parallel programming* (MIT Press)

[83] 2019 The OpenMP(TM) API Specification for Parallel Programming URL https://www.openmp.org

[84] Hennessy J L and Patterson D A 2012 *Computer Architectures: A Quantitative Approach* 5th ed (Morgan Kaufmann, Amsterdam) ISBN 978-0123838728

[85] Rauber T and Rünger G 2013 *Parallel Programming for Multicore and Cluster Systems* 2nd ed

(Springer, Heidelberg)
[86] Reinders J and Jeffers J 2014 *High Performance Parallelism Pearls Volume One: Multicore and Many-core Programming Approaches* 1st ed (Morgan Kaufmann)
[87] Jeffers J and Reinders J 2015 *High Performance Parallelism Pearls Volume Two: Multicore and Many-core Programming Approaches* 1st ed (Morgan Kaufmann)
[88] Patwary M M A, Byna S, Satish N R, Sundaram N, Lukic Z, Roytershteyn V, Anderson M J, Yao Y and Dubey P 2015 Bd-cats: Big data clustering at trillion particle scale *PROCEEDINGS OF SC15: THE INTERNATIONAL CONFERENCE FOR HIGH PERFORMANCE COMPUTING, NETWORKING, STORAGE AND ANALYSIS*
[89] Patwary M M A, Satish N R, Sundaram N, Liu J, Sadowski P, Racah E, Byna S, Tull C, Bhimji W, Prabhat and Dubey P 2016 Panda: Extreme scale parallel k-nearest neighbor on distributed architectures *2016 IEEE International Parallel and Distributed Processing Symposium*
[90] Hutchinson J W 1976 *Proceedings of the Royal Society A* **348** 101–127
[91] Ma A, Roters F and Raabe D 2006 *Acta Materialia* **54** 2169–2179
[92] IBM Raid 6 URL https://www.ibm.com/support/knowledgecenter/TI0002C/p8ebk/raidsix.htm
[93] Schling B 2011 *The Boost C++ Libraries* (XML Press)
[94] Gottlieb D and Shu C W *SIAM Review* **39** 644–668
[95] Gelb A and Gottlieb S 2007 *The Resolution of the Gibbs Phenomenon for Fourier Spectral Methods* ISBN ISBN 0-471-96733010-4-9
[96] Ashby M F 1970 *Philosophical Magazine* **21** 399–424
[97] Habiby F and Humphreys F J 1993 A microstructure investigation of the deformation and recrystallization of particle-containing alumnium silicon alloys *Symposium on Microscale Textures of Materials, At Materials Week 91, Cincinnati, OH, US, October 21-24, 1991* vol 20 pp 125–140
[98] Bellier S P and Doherty R D 1977 *Acta Metallurgica* **25** 10.1016/0001–6160(77)90192–4
[99] Miura H, Sakai T, Andiarwanto S and Jonas J J 2005 *Philosophical Magazine* **85** 2653–2669
[100] Arsenlis A, Becker R, Parks D M and Bulatov V V 2004 *AIP Conference Proceedings* **712**
[101] van der Giessen E and Needleman A 1994 On the solution of two-dimensional plasticity problems with discrete dislocations *Computational Material Modeling* pp 53–70 ASME AD-42
[102] Cleveringa H, van der Giessen E and Needleman A 1998 *Journal Physique IV France* **8** 83–92
[103] Chavarría-Miranda D, Halappanavar M and Kalyanaraman A 2014 Scaling graph community detection on the tilera many-core architecture *Proceedings of the 21st International Conference on High Performance Computing (HiPC)*
[104] Lu H, Halappanavar M and Kalyanaraman A 2015 *Joural of Parallel Computing* **47** 19–37
[105] Naim M, Manne F, Halappanavar M and Tumeo A 2017 Community detection on the gpu *Proceedings of the IEEE International Parallel and Distributed Processing Symposium (IPDPS), 2017, Orlando, Florida* pp 625–634
[106] Ghosh S, Halappanavar M, Tumeo A, Kalyanaraman A, Lu H, Khan A and Gebremedhin A 2018 Distributed louvain algorithm for graph community detection *Proceedings of the 32nd IEEE International Parallel & Distributed Processing Symposium. May 21 – May 25, 2018, Vancouver, Canada*
[107] Amdahl G M 1967 Validity of the single processor approach to achieving large-scale computer capabilities vol 30 pp 483–485
[108] Kühbach M and Roters F 2019 Damaskpdt: Additional post-processing functionality for DAMASK (the düsseldorf advanced material simulation kit) focusing on characterizing the spatial distribution of state variable values as a function of their distance to interfaces URL https://github.com/mkuehbach/damaskpdt
[109] Kühbach M and Roters F 2019 Quantification of 3D spatial correlations between state variables and distance to the grain boundary network in full-field crystal plasticity spectral method simulations URL www.zenodo.org/1249282

[110] Browet A, Absil P A and P van Dooren 2011 Community detection for hierarchical image segmentation *Proceedings of the 14th International Workshop on Combinatorial Image Analysis (IWCIA), Madrid, Spain, May 23-25, 2011*